

Laboratory for Atmospheric and Space Physics

Engineering Division
University of Colorado
Boulder, Colorado

Aeronomy of Ice in the Mesosphere (AIM)

CIPS Command & Telemetry Handbook

Document No. CIPS-T-80007

Revision: C

Date: February 19, 2007



Approvals

Written by:	David Gathright, CIPS Flight Software Lead
Approved by:	Susan Batiste, CIPS Systems Engineer
Approved by:	Deb McCabe, AIM Operations I&T Lead
Approved by:	Mark Lankton, Deputy Program Manager
LASP CM:	

Revisions

Rev.	Description of Change	By	Date
A	First formal release.	D. Gathright	June 27, 2005
B	Second formal release. Updates from CIPS I&T.	D. Gathright	April 26, 2006
C	Final release. Updates from Observatory I&T.	D. Gathright	February 19, 2007

Contents

1	Introduction	25
1.1	Purpose	25
1.2	Scope	25
1.3	Document Status and Schedule	25
1.4	Command and Telemetry Overview	25
1.5	Document Format	27
1.6	Numeric Conventions	27
1.6.1	Endianness	27
1.6.2	Constant Formats	27
1.7	Typographical Conventions	28
2	Reference Documents	29
3	Command Interface	31
3.1	Command List Fields	31
3.2	Command Creation	32
3.3	Command Names	32
3.3.1	Uniqueness	32
3.3.2	Subfields	32
3.4	Command Formats	33
3.5	Packet Formats	33
3.6	FSW Range and Limit Checking	33
3.7	Command Execution Verification	33
3.8	Command Sources	33
3.9	Command Sequences	34
3.9.1	Format	34
3.9.2	Sequence Execution	34
3.9.3	Timing Command Execution	34
3.9.3.1	Timetag Formats	34
3.9.3.2	Timetag Accuracy	35
3.10	Sequence Loads	35
3.11	Clock Synchronization	35
3.12	Hazardous Commands	35
4	Telemetry	37
4.1	Telemetry Measurement List Fields	37
4.2	Telemetry Packet List Fields	37
4.3	Telemetry Names	38
4.4	Packet Formats	38
4.5	Packet Corruption	38

4.6	Telemetry Rates and Bandwidth	38
4.7	Memory Dumps	39
4.8	Diagnostic Packets	39
4.8.1	Event Message Log	39
4.8.2	Post-Mortem Log	40
4.8.3	Exception Data	40
A	Command Verb Summary	41
B	Command List	43
B.1	arm cips	43
B.2	close cips shutter	44
B.3	copy cips memory	45
B.4	disable cips bos	46
B.5	disable cips cam_bin	47
B.6	disable cips cde_uart	48
B.7	disable cips comp	49
B.8	disable cips ee_write	50
B.9	disable cips em	51
B.10	disable cips lens_htr	52
B.11	disable cips sdp	53
B.12	disable cips shutter	54
B.13	disable cips sw_bin	55
B.14	disable cips tlm_out	56
B.15	dump cips memory	57
B.16	enable cips bos	59
B.17	enable cips cam_bin	60
B.18	enable cips cde_uart	61
B.19	enable cips comp	61
B.20	enable cips ee_write	62
B.21	enable cips em	63
B.22	enable cips exc_pm	64
B.23	enable cips lens_htr	65
B.24	enable cips sdp	66
B.25	enable cips sdram_ud	67
B.26	enable cips shutter	68
B.27	enable cips sw_bin	69
B.28	enable cips tlm_out	70
B.29	inform cips acs_state	71
B.30	inform cips pwrdown	73
B.31	inform cips sc_clock	73
B.32	init cips crc	74
B.33	init cips imaging	75
B.34	init cips sequence	77
B.35	init cips sram_code	78
B.36	issue cips cmd	79
B.37	issue cips evt_log	79
B.38	issue cips exception	80
B.39	issue cips hw_hk	81
B.40	issue cips pm	82
B.41	issue cips reserved_1	82
B.42	issue cips reserved_2	83

B.43	issue cips reserved_3	84
B.44	issue cips sw_stat	84
B.45	noop cips	85
B.46	open cips dust_cover	86
B.47	open cips shutter	87
B.48	pass cips cde	88
B.49	reset cips arm	89
B.50	reset cips cpa	90
B.51	reset cips sw_stat	91
B.52	safe cips	91
B.53	select cips cde_chnl	92
B.54	select cips gse_ccd	93
B.55	select cips img_mode	94
B.56	select cips tdi_dir	95
B.57	set cips cde_itf_rt	96
B.58	set cips clear_ct	96
B.59	set cips cmn_a_offset	97
B.60	set cips cmn_a_size	98
B.61	set cips em_mask	99
B.62	set cips frame_sync	100
B.63	set cips hw_hk_rt	101
B.64	set cips img_delay	102
B.65	set cips img_offset	103
B.66	set cips img_size	104
B.67	set cips integration	105
B.68	set cips mcp_volts	105
B.69	set cips obs_id	106
B.70	set cips sw_stat_rt	107
B.71	set cips tdi_row_ct	108
B.72	set cips tdi_tm	109
B.73	set cips to_mask	110
B.74	stuff cips cde_memory	111
B.75	stuff cips memory	112
B.76	stuff cips memory_wd	113
B.77	suspend cips abs_tm	115
B.78	suspend cips rel_tm	116
B.79	term cips sequence	117
B.80	turn off cips camera	117
B.81	turn off cips ips	118
B.82	turn off cips mcp	119
B.83	turn off cips phosphr	120
B.84	turn off cips sdram	121
B.85	turn on cips camera	122
B.86	turn on cips ips	122
B.87	turn on cips mcp	123
B.88	turn on cips phosphr	124
B.89	turn on cips sdram	125
B.90	wake cips	126
B.91	inform inst acs_state	127
B.92	inform inst sc_clock	128

C Command Opcode Summary

131

D Command Packet Summary 135

E Telemetry Measurement List 137

E.1	cips a_seq_addr	137
E.2	cips a_seq_cmd_f_code	137
E.3	cips a_seq_cmd_f_ct	138
E.4	cips a_seq_cmd_fail	139
E.5	cips a_seq_cmd_s_ct	139
E.6	cips a_seq_cmd_succ	140
E.7	cips a_seq_cur_pos	140
E.8	cips a_seq_length	140
E.9	cips a_seq_page	141
E.10	cips a_seq_state	142
E.11	cips a_seq_stop_code	143
E.12	cips a_seq_use_ct	144
E.13	cips a_seq_wait_time	144
E.14	cips a_seq_wait_type	145
E.15	cips acs_sun_uv_x	145
E.16	cips acs_sun_uv_y	146
E.17	cips acs_sun_uv_z	146
E.18	cips active_csci	147
E.19	cips adc_lup	147
E.20	cips armed	148
E.21	cips bits_per_pixel	148
E.22	cips bits_per_pixel_c	149
E.23	cips boot_ct_thresh	149
E.24	cips boot_error	150
E.25	cips bos_trip_ct	150
E.26	cips cam_bin_ena	151
E.27	cips cam_bin_enable	151
E.28	cips cam_fpga_ver	152
E.29	cips cami_fpga_ver	152
E.30	cips cami_pcb_temp	152
E.31	cips ccd_col_bin	153
E.32	cips ccd_row_bin	154
E.33	cips cde_cal_to	154
E.34	cips cde_cmd_rbuf_wds	155
E.35	cips cde_flash_to	155
E.36	cips cde_hb_flutter_ct	155
E.37	cips cde_hist_log_to	156
E.38	cips cde_init_flag	156
E.39	cips cde_itf_rt	157
E.40	cips cde_itf_to	157
E.41	cips cde_mem_dump_to	158
E.42	cips cde_sci_to	158
E.43	cips cde_sys_data_to	159
E.44	cips cde_tlm_rbuf_wds	159
E.45	cips cde_uart_f_err	160
E.46	cips cde_uart_rx_af	160
E.47	cips cde_uart_rx_e	160
E.48	cips cde_uart_rx_f	161
E.49	cips cde_uart_rx_or	161

E.50	cips cde_uart_rx_r	162
E.51	cips cde_uart_rx_r_mk	162
E.52	cips cde_uart_rxor_mk	163
E.53	cips cde_uart_tx_ae	163
E.54	cips cde_uart_tx_af	164
E.55	cips cde_uart_tx_b	164
E.56	cips cde_uart_tx_e	164
E.57	cips cde_uart_tx_e_mk	165
E.58	cips cde_uart_tx_f	165
E.59	cips ce_apid	166
E.60	cips ce_chksum	166
E.61	cips ce_cmd_data	166
E.62	cips ce_func_code	167
E.63	cips ce_opcode	167
E.64	cips ce_pkt_len	167
E.65	cips ce_sec_hdr	168
E.66	cips ce_seq_ct	168
E.67	cips ce_seq_flags	168
E.68	cips ce_type	169
E.69	cips ce_version	169
E.70	cips check_ccd_cmd_p	170
E.71	cips chksum_error	170
E.72	cips clock_tick_bits	171
E.73	cips clock_tick_ssec	171
E.74	cips cmd_echo_to	171
E.75	cips cmd_fail	172
E.76	cips cmd_fail_code	173
E.77	cips cmd_fail_ct	173
E.78	cips cmd_loss_time	174
E.79	cips cmd_rbuf_wds	175
E.80	cips cmd_succ	175
E.81	cips cmd_succ_ct	175
E.82	cips columns_per_map	176
E.83	cips comm_sram_a16	176
E.84	cips comp_scheme	177
E.85	cips cpa_p3_3v_mon	177
E.86	cips cpu_util	178
E.87	cips cpu_util_max	178
E.88	cips crc	179
E.89	cips crc_1	180
E.90	cips crc_1_addr	180
E.91	cips crc_1_length	180
E.92	cips crc_1_type	181
E.93	cips crc_1_valid	181
E.94	cips crc_2	182
E.95	cips crc_2_addr	182
E.96	cips crc_2_length	183
E.97	cips crc_2_type	183
E.98	cips crc_2_valid	184
E.99	cips crc_3	184
E.100	cips crc_3_addr	185
E.101	cips crc_3_length	185

E.102	cips crc_3_type	186
E.103	cips crc_3_valid	186
E.104	cips crc_4	187
E.105	cips crc_4_addr	187
E.106	cips crc_4_length	188
E.107	cips crc_4_type	188
E.108	cips crc_4_valid	189
E.109	cips crc_5	189
E.110	cips crc_5_addr	190
E.111	cips crc_5_length	190
E.112	cips crc_5_type	191
E.113	cips crc_5_valid	191
E.114	cips crc_6	192
E.115	cips crc_6_addr	192
E.116	cips crc_6_length	193
E.117	cips crc_6_type	193
E.118	cips crc_6_valid	194
E.119	cips crc_7	194
E.120	cips crc_7_addr	195
E.121	cips crc_7_length	195
E.122	cips crc_7_type	196
E.123	cips crc_7_valid	196
E.124	cips crc_addr	197
E.125	cips crc_length	197
E.126	cips crc_type	198
E.127	cips crc_valid	198
E.128	cips crc_wpc	199
E.129	cips csci	199
E.130	cips csci_version_num	200
E.131	cips csci_version_pl	200
E.132	cips ctm_rbuf_wds	201
E.133	cips cur_img_ct	201
E.134	cips current_cam	201
E.135	cips current_img	202
E.136	cips dc_activated	202
E.137	cips dc_conv_1_temp	203
E.138	cips dc_conv_2_temp	204
E.139	cips dc_timeout	205
E.140	cips dump_addr	205
E.141	cips dump_data	205
E.142	cips dump_length	206
E.143	cips dump_type	206
E.144	cips e_cami_fpga_ver	207
E.145	cips e_cde_uart_f_err	207
E.146	cips e_cde_uart_rx_af	207
E.147	cips e_cde_uart_rx_e	208
E.148	cips e_cde_uart_rx_f	208
E.149	cips e_cde_uart_rx_or	209
E.150	cips e_cde_uart_rx_r	209
E.151	cips e_cde_uart_tx_ae	210
E.152	cips e_cde_uart_tx_af	210
E.153	cips e_cde_uart_tx_b	210

E.154	cips e_cde_uart_tx_e	211
E.155	cips e_cde_uart_tx_f	211
E.156	cips e_hk_fpga_ver	212
E.157	cips e_proc_fpga_ver	212
E.158	cips e_srt_brx_mc_ill	212
E.159	cips e_srt_brx_sa_ill	213
E.160	cips e_srt_btx_mc_ill	213
E.161	cips e_srt_btx_sa_ill	213
E.162	cips e_srt_cr_bcen	214
E.163	cips e_srt_cr_bufr	214
E.164	cips e_srt_cr_chaen	214
E.165	cips e_srt_cr_chben	215
E.166	cips e_srt_cr_dynbc	215
E.167	cips e_srt_cr_etce	216
E.168	cips e_srt_cr_inten	216
E.169	cips e_srt_cr_ppen	217
E.170	cips e_srt_cr_stex	217
E.171	cips e_srt_desc_ptr	218
E.172	cips e_srt_ill_ptr	218
E.173	cips e_srt_rx_mc_ill	218
E.174	cips e_srt_rx_sa_ill	219
E.175	cips e_srt_tx_mc_ill	219
E.176	cips e_srt_tx_sa_ill	219
E.177	cips ee_write_enable	220
E.178	cips ee_write_ipd	220
E.179	cips eeprom_reset	220
E.180	cips em_acs_stat_rx	221
E.181	cips em_cde_cmd_tx	222
E.182	cips em_cde_tlm_rx	222
E.183	cips em_cmd_err	223
E.184	cips em_cmd_rx	223
E.185	cips em_debug	224
E.186	cips em_err	225
E.187	cips em_img_int	225
E.188	cips em_img_trigger	226
E.189	cips em_info	226
E.190	cips em_mode_chng	227
E.191	cips em_pwr	227
E.192	cips em_sdp_start	228
E.193	cips em_sdp_stop	228
E.194	cips em_seq_cmd_rx	229
E.195	cips em_seq_err	229
E.196	cips em_seq_evt	230
E.197	cips em_seq_mode_chng	230
E.198	cips em_time_rx	231
E.199	cips em_tlm_err	232
E.200	cips em_tlm_tx	232
E.201	cips em_u_test	233
E.202	cips em_warn_msg	233
E.203	cips err_to	234
E.204	cips event_ct	234
E.205	cips event_info	235

E.206	cips event_key	235
E.207	cips event_p0	236
E.208	cips event_p1	236
E.209	cips event_p2	237
E.210	cips event_p3	237
E.211	cips event_pc	237
E.212	cips event_time_s	238
E.213	cips event_time_ss	238
E.214	cips evt_log_to	238
E.215	cips exc_enable	239
E.216	cips exception_to	239
E.217	cips fast_clears	240
E.218	cips frame_sync_time	240
E.219	cips gndv_mon	241
E.220	cips hk_counter	242
E.221	cips hk_fpga_ver	242
E.222	cips hk_irq_1s	242
E.223	cips hk_irq_1s_mk	243
E.224	cips hk_irq_lpevt	243
E.225	cips hk_irq_lpevt_mk	244
E.226	cips hk_irq_lppwr	244
E.227	cips hk_irq_lppwr_mk	244
E.228	cips hk_irq_mx_bos	245
E.229	cips hk_irq_mx_bos_mk	245
E.230	cips hk_irq_my_bos	246
E.231	cips hk_irq_my_bos_mk	246
E.232	cips hk_irq_px_bos	247
E.233	cips hk_irq_px_bos_mk	247
E.234	cips hk_irq_py_bos	247
E.235	cips hk_irq_py_bos_mk	248
E.236	cips hk_irq_sdrami	248
E.237	cips hk_irq_sdrami_mk	249
E.238	cips hk_lpbit_ena	249
E.239	cips hk_mux_1_addr	250
E.240	cips hk_mux_2_addr	250
E.241	cips hk_mux_ena	250
E.242	cips hk_mux_test_ena	251
E.243	cips hk_pcb_temp	251
E.244	cips hw_hk_rt	252
E.245	cips hw_hk_to	253
E.246	cips i_cs_thresh	253
E.247	cips i_l_htr_lthresh	253
E.248	cips i_l_htr_uthresh	254
E.249	cips i_mx_bos_level	255
E.250	cips i_my_bos_level	256
E.251	cips i_px_bos_level	257
E.252	cips i_py_bos_level	257
E.253	cips img_cols	258
E.254	cips img_cols_ccd	258
E.255	cips img_cols_sdram	259
E.256	cips img_cols_tlm	259
E.257	cips img_delay_tm	259

E.258	cips img_mode	260
E.259	cips img_offset_col	260
E.260	cips img_offset_row	261
E.261	cips img_remaining	261
E.262	cips img_rows	261
E.263	cips img_rows_ccd	262
E.264	cips img_rows_sdram	262
E.265	cips img_rows_tlm	263
E.266	cips img_taken	263
E.267	cips img_timestamp_s	263
E.268	cips img_timestamp_ss	264
E.269	cips img_to_be_taken	265
E.270	cips img_with_mx_cam	265
E.271	cips img_with_my_cam	265
E.272	cips img_with_px_cam	266
E.273	cips img_with_py_cam	266
E.274	cips lens_htr_lthresh	267
E.275	cips lens_htr_uthresh	268
E.276	cips m15i_mon	269
E.277	cips m15v_mon	270
E.278	cips mem_dump_remain	270
E.279	cips mem_dump_to	271
E.280	cips mode	271
E.281	cips ms1553.ac.a.s.ct	272
E.282	cips ms1553.ac.b.s.ct	272
E.283	cips ms1553.b.a.ce.ct	273
E.284	cips ms1553.b.a.me.ct	273
E.285	cips ms1553.b.a.oe.ct	274
E.286	cips ms1553.b.a.pe.ct	274
E.287	cips ms1553.b.a.te.ct	275
E.288	cips ms1553.b.b.ce.ct	275
E.289	cips ms1553.b.b.me.ct	276
E.290	cips ms1553.b.b.oe.ct	276
E.291	cips ms1553.b.b.pe.ct	277
E.292	cips ms1553.b.b.te.ct	277
E.293	cips ms1553.bit.f.ct	278
E.294	cips ms1553.bit.f.wd	278
E.295	cips ms1553.bit.s.ct	279
E.296	cips ms1553.c1.a.s.ct	279
E.297	cips ms1553.c1.b.s.ct	280
E.298	cips ms1553.c2.a.s.ct	280
E.299	cips ms1553.c2.b.s.ct	280
E.300	cips ms1553.c.b.o.ct	281
E.301	cips ms1553.c.c.p.ct	281
E.302	cips ms1553.ct.a.s.ct	282
E.303	cips ms1553.ct.b.o.ct	282
E.304	cips ms1553.ct.b.s.ct	283
E.305	cips ms1553.ct.c.p.ct	283
E.306	cips ms1553.ct.wd.ct	284
E.307	cips ms1553.db.e.ct	284
E.308	cips ms1553.ei.ct	285
E.309	cips ms1553.hw.r.ct	285

E.310	cips ms1553_iiw_e_ct	286
E.311	cips ms1553_iiw_e_wd	286
E.312	cips ms1553_nt_a_s_ct	286
E.313	cips ms1553_nt_b_o_ct	287
E.314	cips ms1553_nt_b_s_ct	287
E.315	cips ms1553_nt_c_p_ct	288
E.316	cips ms1553_nt_wd_ct	288
E.317	cips ms1553_to_thresh	289
E.318	cips ms1553_ts_a_s_ct	289
E.319	cips ms1553_ts_b_s_ct	289
E.320	cips ms1553_tv_a_s_ct	290
E.321	cips ms1553_tv_b_s_ct	290
E.322	cips ms_delay	291
E.323	cips mx_bos_ena	291
E.324	cips mx_bos_level	291
E.325	cips mx_bos_thresh	292
E.326	cips mx_cam_clear_ct	293
E.327	cips mx_cam_f_sync	293
E.328	cips mx_cam_mcp_pwr	293
E.329	cips mx_cam_mcpv_mon	294
E.330	cips mx_cam_mode	295
E.331	cips mx_cam_p_pwr	295
E.332	cips mx_cam_pv_mon	296
E.333	cips mx_cam_pwr	296
E.334	cips mx_cam_ss_time	297
E.335	cips mx_cam_tdi_rows	297
E.336	cips mx_cam_tdi_time	298
E.337	cips mx_ccd_temp	298
E.338	cips mx_dc_status_pri	299
E.339	cips mx_dc_status_sec	300
E.340	cips mx_ib_of	300
E.341	cips mx_ib_of_mk	301
E.342	cips mx_img_end	301
E.343	cips mx_img_end_mk	302
E.344	cips mx_int_temp	302
E.345	cips mx_ips_pwr	303
E.346	cips mx_lens_a_temp	303
E.347	cips mx_lens_b_temp	304
E.348	cips mx_lens_htr_ena	305
E.349	cips mx_my_ips_temp	306
E.350	cips mx_opcode_err	306
E.351	cips mx_opcode_err_mk	307
E.352	cips mx_parity_err	307
E.353	cips mx_parity_err_mk	308
E.354	cips mx_rx	308
E.355	cips mx_rx_mk	308
E.356	cips mx_seq_err	309
E.357	cips mx_seq_err_mk	309
E.358	cips mx_shutr_disable	310
E.359	cips mx_shutr_state	310
E.360	cips my_bos_ena	311
E.361	cips my_bos_level	311

E.362	cips my_bos_thresh	312
E.363	cips my_cam_clear_ct	313
E.364	cips my_cam_f_sync	313
E.365	cips my_cam_mcp_pwr	313
E.366	cips my_cam_mcpv_mon	314
E.367	cips my_cam_mode	315
E.368	cips my_cam_p_pwr	315
E.369	cips my_cam_pv_mon	316
E.370	cips my_cam_pwr	316
E.371	cips my_cam_ss_time	317
E.372	cips my_cam_tdi_rows	317
E.373	cips my_cam_tdi_time	317
E.374	cips my_ccd_temp	318
E.375	cips my_dc_status_pri	319
E.376	cips my_dc_status_sec	319
E.377	cips my_ib_of	320
E.378	cips my_ib_of_mk	320
E.379	cips my_img_end	321
E.380	cips my_img_end_mk	321
E.381	cips my_int_temp	322
E.382	cips my_ips_pwr	322
E.383	cips my_lens_a_temp	323
E.384	cips my_lens_b_temp	324
E.385	cips my_lens_htr_ena	324
E.386	cips my_opcode_err	325
E.387	cips my_opcode_err_mk	325
E.388	cips my_parity_err	326
E.389	cips my_parity_err_mk	326
E.390	cips my_rx	327
E.391	cips my_rx_mk	327
E.392	cips my_seq_err	328
E.393	cips my_seq_err_mk	328
E.394	cips my_shutr_disable	328
E.395	cips my_shutr_state	329
E.396	cips o_seq_addr	329
E.397	cips o_seq_cmd_f_code	330
E.398	cips o_seq_cmd_f_ct	331
E.399	cips o_seq_cmd_fail	331
E.400	cips o_seq_cmd_s_ct	332
E.401	cips o_seq_cmd_succ	332
E.402	cips o_seq_cur_pos	332
E.403	cips o_seq_length	333
E.404	cips o_seq_page	333
E.405	cips o_seq_state	334
E.406	cips o_seq_stop_code	335
E.407	cips o_seq_use_ct	336
E.408	cips o_seq_wait_time	336
E.409	cips o_seq_wait_type	337
E.410	cips obs_id	337
E.411	cips opcode_err_ct	338
E.412	cips p15i_mon	338
E.413	cips p15v_mon	339

E.414	cips p20v_mon	340
E.415	cips p2_5v_mon	340
E.416	cips p352flg	341
E.417	cips p352hws	341
E.418	cips p352hwss	342
E.419	cips p352lws	342
E.420	cips p352lwss	343
E.421	cips p352pid	343
E.422	cips p352pl	343
E.423	cips p352sct	344
E.424	cips p353flg	344
E.425	cips p353hws	344
E.426	cips p353hwss	345
E.427	cips p353lws	345
E.428	cips p353lwss	345
E.429	cips p353pid	346
E.430	cips p353pl	346
E.431	cips p353sct	346
E.432	cips p354flg	347
E.433	cips p354hws	347
E.434	cips p354hwss	347
E.435	cips p354lws	348
E.436	cips p354lwss	348
E.437	cips p354pid	348
E.438	cips p354pl	349
E.439	cips p354sct	349
E.440	cips p355flg	349
E.441	cips p355hws	350
E.442	cips p355hwss	350
E.443	cips p355lws	350
E.444	cips p355lwss	351
E.445	cips p355pid	351
E.446	cips p355pl	351
E.447	cips p355sct	352
E.448	cips p356flg	352
E.449	cips p356hws	352
E.450	cips p356hwss	353
E.451	cips p356lws	353
E.452	cips p356lwss	353
E.453	cips p356pid	354
E.454	cips p356pl	354
E.455	cips p356sct	354
E.456	cips p357flg	355
E.457	cips p357hws	355
E.458	cips p357hwss	355
E.459	cips p357lws	356
E.460	cips p357lwss	356
E.461	cips p357pid	356
E.462	cips p357pl	357
E.463	cips p357sct	357
E.464	cips p358flg	357
E.465	cips p358hws	358

E.466	cips p358hwss	358
E.467	cips p358lws	358
E.468	cips p358lwss	359
E.469	cips p358pid	359
E.470	cips p358pl	359
E.471	cips p358sct	360
E.472	cips p359flg	360
E.473	cips p359hws	360
E.474	cips p359hwss	361
E.475	cips p359lws	361
E.476	cips p359lwss	361
E.477	cips p359pid	362
E.478	cips p359pl	362
E.479	cips p359sct	362
E.480	cips p360flg	363
E.481	cips p360hws	363
E.482	cips p360hwss	363
E.483	cips p360lws	364
E.484	cips p360lwss	364
E.485	cips p360pid	364
E.486	cips p360pl	365
E.487	cips p360sct	365
E.488	cips p361flg	365
E.489	cips p361hws	366
E.490	cips p361hwss	366
E.491	cips p361lws	366
E.492	cips p361lwss	367
E.493	cips p361pid	367
E.494	cips p361pl	367
E.495	cips p361sct	368
E.496	cips p362flg	368
E.497	cips p362hws	368
E.498	cips p362hwss	369
E.499	cips p362lws	369
E.500	cips p362lwss	369
E.501	cips p362pid	370
E.502	cips p362pl	370
E.503	cips p362sct	370
E.504	cips p363flg	371
E.505	cips p363hws	371
E.506	cips p363hwss	371
E.507	cips p363lws	372
E.508	cips p363lwss	372
E.509	cips p363pid	372
E.510	cips p363pl	373
E.511	cips p363sct	373
E.512	cips p5i_mon	373
E.513	cips p5v_mon	374
E.514	cips parity_err_ct	375
E.515	cips parm_tbl_crc	375
E.516	cips parm_tbl_flag	376
E.517	cips pixel_count_err	376

E.518	cips pixel_stream	377
E.519	cips pixels_in_pkt	377
E.520	cips pixels_per_map	377
E.521	cips pixels_per_pkt	378
E.522	cips pkt_crc	378
E.523	cips pkt_filler	378
E.524	cips pkt_version	379
E.525	cips pkt_version_el	379
E.526	cips pm_enable	379
E.527	cips pm_event_ct	380
E.528	cips pm_event_info	380
E.529	cips pm_key	381
E.530	cips pm_p0	381
E.531	cips pm_p1	381
E.532	cips pm_p2	382
E.533	cips pm_p3	382
E.534	cips pm_pc	382
E.535	cips pm_time_s	383
E.536	cips pm_time_ss	383
E.537	cips pm_to	383
E.538	cips proc_fpga_ver	384
E.539	cips proc_me_bterr	384
E.540	cips proc_me_mprot	385
E.541	cips proc_mode	385
E.542	cips proc_mode_rset	385
E.543	cips proc_mx_cam_rs	386
E.544	cips proc_my_cam_rs	386
E.545	cips proc_pcb_temp	387
E.546	cips proc_px_cam_rs	387
E.547	cips proc_py_cam_rs	388
E.548	cips proc_self_rset	388
E.549	cips proc_wd_rset	388
E.550	cips px_bos_ena	389
E.551	cips px_bos_level	389
E.552	cips px_bos_thresh	390
E.553	cips px_cam_clear_ct	391
E.554	cips px_cam_f_sync	391
E.555	cips px_cam_mcp_pwr	392
E.556	cips px_cam_mcpv_mon	392
E.557	cips px_cam_mode	393
E.558	cips px_cam_p_pwr	393
E.559	cips px_cam_pv_mon	394
E.560	cips px_cam_pwr	394
E.561	cips px_cam_ss_time	395
E.562	cips px_cam_tdi_rows	395
E.563	cips px_cam_tdi_time	396
E.564	cips px_ccd_temp	396
E.565	cips px_dc_status_pri	397
E.566	cips px_dc_status_sec	398
E.567	cips px_ib_of	398
E.568	cips px_ib_of_mk	399
E.569	cips px_img_end	399

E.570	cips_px_img_end_mk	400
E.571	cips_px_int_temp	400
E.572	cips_px_ips_pwr	401
E.573	cips_px_lens_a_temp	401
E.574	cips_px_lens_b_temp	402
E.575	cips_px_lens_htr_ena	403
E.576	cips_px_opcode_err	404
E.577	cips_px_opcode_err_mk	404
E.578	cips_px_parity_err	404
E.579	cips_px_parity_err_mk	405
E.580	cips_px_py_ips_temp	405
E.581	cips_px_rx	406
E.582	cips_px_rx_mk	407
E.583	cips_px_seq_err	407
E.584	cips_px_seq_err_mk	407
E.585	cips_px_shutr_disable	408
E.586	cips_px_shutr_state	408
E.587	cips_py_bos_ena	409
E.588	cips_py_bos_level	409
E.589	cips_py_bos_thresh	410
E.590	cips_py_cam_clear_ct	411
E.591	cips_py_cam_f_sync	411
E.592	cips_py_cam_mcp_pwr	411
E.593	cips_py_cam_mcpv_mon	412
E.594	cips_py_cam_mode	413
E.595	cips_py_cam_p_pwr	413
E.596	cips_py_cam_pv_mon	414
E.597	cips_py_cam_pwr	414
E.598	cips_py_cam_ss_time	415
E.599	cips_py_cam_tdi_rows	415
E.600	cips_py_cam_tdi_time	416
E.601	cips_py_ccd_temp	416
E.602	cips_py_dc_status_pri	417
E.603	cips_py_dc_status_sec	418
E.604	cips_py_ib_of	418
E.605	cips_py_ib_of_mk	419
E.606	cips_py_img_end	419
E.607	cips_py_img_end_mk	420
E.608	cips_py_int_temp	420
E.609	cips_py_ips_pwr	421
E.610	cips_py_lens_a_temp	421
E.611	cips_py_lens_b_temp	422
E.612	cips_py_lens_htr_ena	423
E.613	cips_py_opcode_err	424
E.614	cips_py_opcode_err_mk	424
E.615	cips_py_parity_err	424
E.616	cips_py_parity_err_mk	425
E.617	cips_py_rx	425
E.618	cips_py_rx_mk	426
E.619	cips_py_seq_err	426
E.620	cips_py_seq_err_mk	427
E.621	cips_py_shutr_disable	427

E.622	cips py_shutr_state	428
E.623	cips quick_boot_flag	428
E.624	cips reserved_1_to	428
E.625	cips reserved_2_to	429
E.626	cips reserved_3_to	429
E.627	cips reserved_4_to	430
E.628	cips reserved_bit	431
E.629	cips row_seq_err_ct	431
E.630	cips safe_type	431
E.631	cips sci_mx_cam_to	432
E.632	cips sci_my_cam_to	432
E.633	cips sci_px_cam_to	433
E.634	cips sci_py_cam_to	434
E.635	cips sdp_enable	434
E.636	cips sdp_in_progress	435
E.637	cips sdram_a_pwr	435
E.638	cips sdram_b_pwr	436
E.639	cips sdram_chip_remap	436
E.640	cips sdram_i_mon	437
E.641	cips sdram_i_thresh	438
E.642	cips sdram_pixel_map	438
E.643	cips sdram_trip_ct	439
E.644	cips sdram_update_ena	439
E.645	cips shutter_cl_time	440
E.646	cips shutter_disable	440
E.647	cips sram_img_1_addr	441
E.648	cips sram_img_1_crc	441
E.649	cips sram_img_1_len	442
E.650	cips sram_img_1_pg	442
E.651	cips sram_img_2_addr	443
E.652	cips sram_img_2_crc	443
E.653	cips sram_img_2_len	443
E.654	cips sram_img_2_pg	444
E.655	cips sram_img_to_ld	445
E.656	cips srt_bdrvc	445
E.657	cips srt_bdrvc_mk	445
E.658	cips srt_bitf	446
E.659	cips srt_bitf_mk	446
E.660	cips srt_brx_mc_ill	447
E.661	cips srt_brx_sa_ill	447
E.662	cips srt_btx_mc_ill	448
E.663	cips srt_btx_sa_ill	448
E.664	cips srt_cr_bcen	449
E.665	cips srt_cr_bufc	449
E.666	cips srt_cr_chaen	449
E.667	cips srt_cr_chben	450
E.668	cips srt_cr_dynbc	450
E.669	cips srt_cr_etce	451
E.670	cips srt_cr_inten	451
E.671	cips srt_cr_ppen	452
E.672	cips srt_cr_stex	452
E.673	cips srt_desc_ptr	452

E.674	cips srt_dmaf	453
E.675	cips srt_dmaf_mk	453
E.676	cips srt_ill_ptr	454
E.677	cips srt_illcmd	454
E.678	cips srt_illcmd_mk	455
E.679	cips srt_ixeq0	455
E.680	cips srt_ixeq0_mk	456
E.681	cips srt_merr	456
E.682	cips srt_merr_mk	456
E.683	cips srt_rx_mc_ill	457
E.684	cips srt_rx_sa_ill	457
E.685	cips srt_subad	458
E.686	cips srt_subad_mk	458
E.687	cips srt_tapf	459
E.688	cips srt_tapf_mk	459
E.689	cips srt_tx_mc_ill	460
E.690	cips srt_tx_sa_ill	460
E.691	cips srt_wrapf	461
E.692	cips srt_wrapf_mk	461
E.693	cips stack_util_max	461
E.694	cips start_pixel	462
E.695	cips sw_bin_ena	462
E.696	cips sw_bin_enable	463
E.697	cips sw_comp_enable	463
E.698	cips sw_stat_rt	464
E.699	cips sw_stat_to	464
E.700	cips tdi_dir	465
E.701	cips tdi_dwell	465
E.702	cips tdi_rows	466
E.703	cips test_cam_select	466
E.704	cips tlm_pkts_per_blk	467
E.705	cips tlm_rbuf_wds	467
E.706	cips tlm_words_sent	468
E.707	cips uart_ab_sel	468
E.708	cips uart_fifo_level	468
E.709	cips ut69r_acc	469
E.710	cips ut69r_ft	469
E.711	cips ut69r_ics	469
E.712	cips ut69r_mk	470
E.713	cips ut69r_pi	470
E.714	cips ut69r_r10	470
E.715	cips ut69r_r11	471
E.716	cips ut69r_r12	471
E.717	cips ut69r_r13	471
E.718	cips ut69r_r14	472
E.719	cips ut69r_r15	472
E.720	cips ut69r_r16	472
E.721	cips ut69r_r17	473
E.722	cips ut69r_r2	473
E.723	cips ut69r_r3	473
E.724	cips ut69r_r4	474
E.725	cips ut69r_r5	474

E.726	cips ut69r_r6	475
E.727	cips ut69r_r7	475
E.728	cips ut69r_r8	475
E.729	cips ut69r_r9	476
E.730	cips ut69r_rcvr	476
E.731	cips ut69r_status	476
E.732	cips ut69r_sw	477
E.733	cips ut69r_ta	477
E.734	cips ut69r_tb	477
E.735	cips wind_sync_err_ct	478
F	Telemetry Packet List	479
F.1	cips cmd_echo	480
F.2	cips err	481
F.3	cips evt_log	482
F.4	cips exception	484
F.5	cips hw_hk	492
F.6	cips mem_dump	499
F.7	cips post_mortem	500
F.8	cips sci_mx_cam	502
F.9	cips sci_my_cam	505
F.10	cips sci_px_cam	508
F.11	cips sci_py_cam	511
F.12	cips sw_status	514
G	Telemetry Packet Summary	523
H	Glossary	525
	References	527

List of Tables

4.1 CIPS Telemetry Bandwidth Calculation Assumptions 39

List of Figures

1.1 Flight Software Data Flow	26
---	----

1

Introduction

1.1 Purpose

The purpose of this document is to capture detailed information on the command and telemetry interfaces for the Cloud Imaging and Particle Size (CIPS) instrument on the Aeronomy of Ice in the Mesosphere (AIM) spacecraft. It satisfies the requirements set forth for command and telemetry documentation as documented in the *AIM Mission Software Development Plan*[1].

1.2 Scope

This document is written for the CIPS Integration and Test and AIM Mission Operations teams. It is also meant to be used by the CIPS Engineering Team to ensure that CIPS FSW is producing adequate telemetry for diagnostic purposes and to ensure that CIPS FSW can command every necessary feature of the CIPS hardware.

1.3 Document Status and Schedule

This is the initial draft of this document, which was written with the expectation that some details may change during the development process. Updates to this document will be performed according to the CIPS FSW standard configuration management process as defined in the *CIPS FSW Development Plan*[2].

The most current unofficial version of this document and all other CIPS FSW documentation can be obtained from the AIM FSW website at <http://lasp.colorado.edu/aim-eng/fsw>. Official releases should be obtained through AIM Product Assurance.

1.4 Command and Telemetry Overview

Figure 1.1 on the following page shows a high level view of data flow on the AIM mission. Almost all data exchanged between the S/C bus and CIPS is in the form of CCSDS packets¹.

¹The only spacecraft interfaces that are not CCSDS packets are the clock SYNC, transmit status word, transmitter shutdown, and override transmitter shutdown MIL-STD-1553B mode codes.

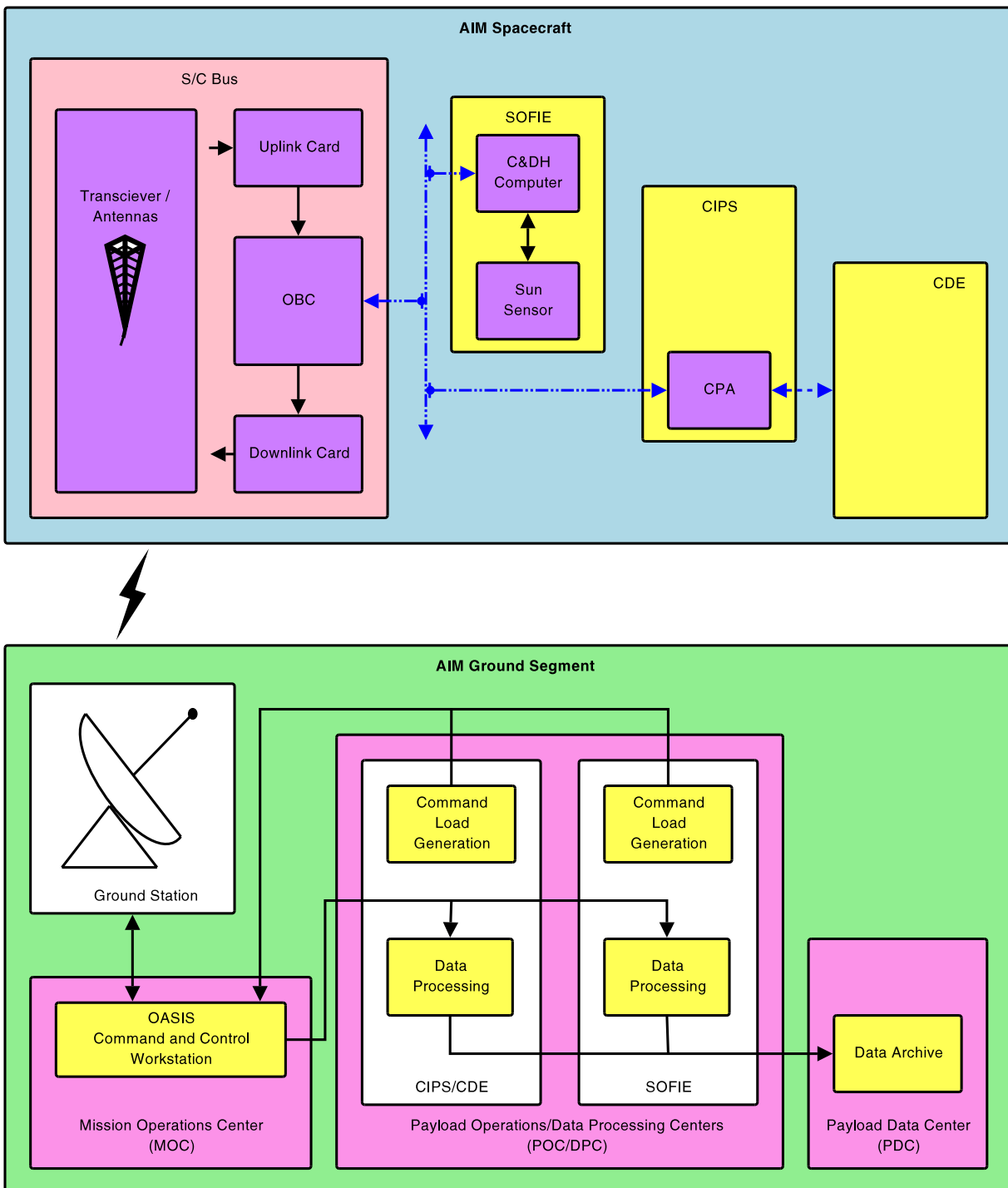


Figure 1.1: Flight Software Data Flow

All commands are in the form of CCSDS telecommand packets with only one command per packet. Commands are read from the spacecraft's MIL-STD-1553 data bus via the 1553 device driver and are buffered in the CIPS command buffer. At the appropriate time in the CIPS slice processing schedule², commands are read from the CIPS command buffer and are immediately processed. If the CCSDS packet contains a command for CDE, as specified by the packet's APID, CIPS routes this command to the CDE command buffer and immediately initiates transmission of this command data to CDE.

Telemetry packets are produced by CIPS only when needed; i.e. CIPS produces no 'fill' telemetry packets. Packets are written to one of the CIPS' two telemetry buffers by the telemetry service upon packet creation or receipt from CDE. Most telemetry is sent through the 'nominal' telemetry channel, but every other software status packet and all error packets are sent on the 'critical' telemetry channel. At the appropriate time in the 1553 bus schedule, the CIPS 1553 device driver removes up to 1024 bytes of integral telemetry packets from one of the CIPS telemetry buffers and writes them to the spacecraft data bus.

1.5 Document Format

The appendices contain the bulk of the information in this document and specify the bit-level command and telemetry formats. These appendices are actually reports from the AIM master command and telemetry database, which contains the single point definitions for all CIPS commands and telemetry.

A glossary of abbreviations and terms used in this document starts on page 524, while a bibliography of referenced documents can be found on page 527.

This document is available as an electronic portable document file (PDF), or in hard copy. Electronic file is the preferred format, as this document contains embedded hyperlinks to other sections and to external web pages. The latest electronic version of this document may be obtained from <http://lasp.colorado.edu/aim-eng/fsw>.

1.6 Numeric Conventions

1.6.1 Endianness

Unless otherwise specified, all bits and bytes are *big endian*, per CCSDS standards. In big endian storage, the most significant bit (MSb) is stored at the lowest (or starting) address, while the least significant bit (LSb) is stored at the highest (or ending) address. The same holds true for bytes: the most significant byte (MSB) is stored at the lowest (or starting) address, while the least significant byte (LSB) is stored at the highest (or ending) address.

1.6.2 Constant Formats

Unless otherwise specified, all constants are expressed as decimal values. Hexadecimal constants are expressed with a leading *0x*, and binary constants are expressed with a leading *0b*. Examples:

1234	decimal constant
0x89AB	hexadecimal constant
0xC000.0400	hexadecimal constant with 16 bit field separator
0b11011	binary constant

²The CIPS FSW Design Description[3] contains detailed information on slice processing.

1.7 Typographical Conventions

When viewed as a portable document format (PDF) electronic document, hyperlinks appear as **colored text**. There is a substantial number of hyperlinks contained in this document that are intended to show the relations between commands and telemetry.

Bibliographical citations appear as italicized text, followed by the citation index, for example, the *AIM Mission Software Development Plan*[\[1\]](#).

Specific command names, telemetry names, or packet names appear as bold text, followed by the item definition page number, as a hyperlink, for example the **instrument science**(90) telemetry packet would be defined on page 90.

2

Reference Documents

This document is written in accordance and in compliance with the guidelines and requirements outlined in the *AIM Mission Software Development Plan*[1].

External documents impacting instrument FSW command and telemetry interfaces include those listed below.

- *AIM Spacecraft - Instruments Communications ICD*[4]
- *New Horizons to SDC ICD*[5]
- *CIPS Level 3 Specification*[6]
- *CIPS FSW Requirements Specification*[7]

The *CIPS FSW Design Description*[3] contains additional information about command processing and telemetry production. All other documents referenced by this document are listed in the References section on page 527.

2. REFERENCE DOCUMENTS

3

Command Interface

3.1 Command List Fields

Every CIPS FSW command is detailed in Appendix B on page 43. Each command listing contains the following fields:

- the APID associated with that command;
- a brief description of the command's purpose;
- an optional discussion that elaborates on the command's execution;
- an optional description of the constraints on command execution;
- the command target (i.e. final processor of the command);
- the complete bit-wise definition of the command format;
- a listing of all command subfields, including the subfield's
 - summary description,
 - detailed discussion (if any),
 - units (if any),
 - state conversions (if any),
 - minimum and maximum permissible values, and
 - default value;
- whether or not the command is safe or hazardous;
- what telemetry item(s) can be used to verify command execution;
- a listing of any related commands;
- a listing of any related telemetry items; and
- a listing of any related telemetry packets.

3.2 Command Creation

A command has been created for every low level CIPS hardware or software function. Thus, the command interface will provide complete access to all instrument functionality and will support any future changes in the way instruments are operated. For generic access to hardware functionality or features, the *stuff cips memory_wd* command has been implemented with access to every FPGA register and the *dump cips memory* command has the capability of dumping FPGA register contents.

High level commands are more efficient in that combinations of low level functionality may be in a single command. High level commands may be implemented in FSW, in sequences or both. The initial approach to command creation is to create all required instrument low level commands. High level commands will be created as warranted by uplink time demands. This is possible because CIPS FSW has been designed such that it is relatively easy to extend or cut back on the commands it can process without necessitating a completely new code load.

3.3 Command Names

Command names follow the convention specified in the *OASIS-CC System Manager's Guide* [8]. All commands have the form

verb_phrase element [sub_element]

The *verb_phrase* describes the action, the *element* identifies the receiving element that performs the action, and the optional *sub_element* may further define the target of the action.

The convention of this document is to specify command names in lower case letters, though OASIS-CC is case independent.

A complete listing of the verb names used in CIPS commands appears in Appendix A on page 41.

The only element associated with CIPS commands is *cips*. There are four separate telecommand APIDs reserved for CIPS commands covering the range from 0x160 to 0x163. There is no difference in CIPS command processing for the same command sent in a packet with any of these APIDs—they are only there for possible future use or in case there is some issue with one of the APIDs.

3.3.1 Uniqueness

Command names will be globally unique. That is, there will not be multiple instances of a given command name. Portions of a command name taken by themselves may not be globally unique. Note that commands for CIPS may apply to one or both of the CIPS FSW CSCIs: SuROM and SRAM Code. The command listing in the appendix clearly designates which of the CSCIs accept each command.

3.3.2 Subfields

The inclusion of command subfields makes CIPS FSW command processing logic more complex. Thus, where possible and logical, commands have been defined with a minimum of command subfields. Subfield names were chosen to aid in the command readability, which also helps facilitate familiarity with the CIPS command set.

3.4 Command Formats

Command names distinguish different commands in the OASIS ground software system, whereas 8-bit opcodes serve the same purpose in CIPS FSW. Thus a unique 8-bit command opcode has been assigned to every command.

The AIM spacecraft's 1553 data bus is constrained to only handle even byte sized data transfers. Also, the CIPS microprocessor on which CIPS FSW executes is a 16-bit device. For these reasons, each command has been defined such that the command as a whole is an integer multiple of 16 bits in length. Command packets on AIM are constrained to be no more than 256 bytes in length, including all header fields.

3.5 Packet Formats

The AIM command packet format is defined in the *AIM Spacecraft to Instruments Communications ICD*[4]. All command packets are CCSDS telecommand packets with a 16-bit secondary header containing 8 unused (by CIPS FSW) bits followed by a checksum in the lower 8 bits of the secondary header. This checksum is a simple XOR of all bytes in the packet.

CIPS will declare a command packet as 'corrupt' if any of the following conditions are met:

- the version field in the CCSDS primary header is non-zero,
- the type flag in the CCSDS primary header is not one, or
- the sequence flags in the CCSDS primary header are not both set.

Packets are further checked for corruption by CIPS FSW through command checksum validation and validation of the command length, as each command has a fixed length.

3.6 FSW Range and Limit Checking

Prior to execution, all command subfields are verified by CIPS FSW to contain values within specified ranges, as specified in the command listing in Appendix B on page 43. CIPS will reject and not execute any commands that fail these tests.

3.7 Command Execution Verification

Successful execution of every CIPS command should be easily verified in telemetry. In most cases, the verification is very simple and thus, no explanation is included in the command listing in Appendix B on page 43. Commands that relate directly to telemetry packet production may not contain an entry in the Command Execution Verification field in the command listing.

All properly formatted command packets received by CIPS should result in either the *cips cmd_succ_ct* or *cips cmd_fail_ct* telemetry item incrementing.

3.8 Command Sources

CIPS commands may be classified in two groups based on where they originate. 'External' commands are those commands that originate external to CIPS. This includes commands routed in real-time from ground controllers through the

spacecraft and commands residing in the spacecraft's ATS or RTSes. All CIPS commands may be external commands with the exception of the *suspend* commands used to control command sequence execution timing.

Commands coming from CIPS command sequence engines are 'internal' commands. All CIPS commands may be internal commands with the exception of the *arm cips* and *wake cips* commands, which will only be executed if they originate from an external source (are external commands).

3.9 Command Sequences

CIPS has its own on-board command sequence capability, with the ability to execute up to two command sequences simultaneously. However, only one of these command sequences is completely under operator control. The other command sequence engine is used for internal, autonomous command sequencing associated with CIPS mode switches. This sequence engine may be commanded to stop command processing but cannot be directly restarted from ground command.

The main sequence engine, also called the operator-initiated sequence engine, is under complete control of the operators. This is the engine that will process the daily command loads.

All sequence functionality is identical with the command sequencing done on the SORCE instrument module micro-processor, with which AIM Mission Operators are very familiar. It is intended that the daily command loads be done in a ping-pong scheme in the same manner for CIPS as on SORCE, with two unique sequence storage areas allocated for successive days' command loads and each day's command load containing as the last command in the sequence an *init cips sequence* command that switches sequence command processing to the other buffer.

3.9.1 Format

Command sequence formats are discussed in detail in the CIPS FSW Design Description[3] in the 'Command Sequence Engine' section. Command sequences consist of a 18-bit sequence header followed by a contiguous stream of commands (opcode followed by command subfields). The sequence header contains the sequence length, a 16-bit CRC of the command data contained in the sequence, and a 16-bit use count, which allows a given command sequence to be executed either a finite (up to 65534) number of times or infinitely many times.

3.9.2 Sequence Execution

Commands are executed from each command sequence engine at most once per second, with the only delay in execution coming from *suspend* commands, as described in Section 3.9.3.

3.9.3 Timing Command Execution

Commands in a sequence may be executed at a specific time using *suspend* commands within the sequence. These commands specify how long CIPS should wait before processing the next command in the sequence. Using *suspend* commands means that memory is not wasted by requiring every command in a sequence to have a timetag for execution.

3.9.3.1 Timetag Formats

There are two types of *suspend* commands specifying either a relative (*suspend cips rel_tm*) or an absolute (*suspend cips abs_tm*) time wait. Absolute times are based on the internal CIPS clock, which is synchronized to the spacecraft

clock. Both commands have resolutions of 1 second.

3.9.3.2 Timetag Accuracy

The command immediately following a *suspend* command is guaranteed to be processed within 1 second of the time specified. Relative waits are independent of any variances in the CIPS local clock; however, any sudden jumps in the spacecraft time that propagate down into the CIPS clock may affect the timing of absolute time waits. For this reason, CIPS FSW will immediately terminate any sequence if the commanded wait time in an absolute wait is more than 1 second in the past. This verification of the wait time is also done for relative time waits out of sheer paranoia, with the same result if the wait took more than 1 second longer than it should have.

3.10 Sequence Loads

CIPS command sequences are loaded into CIPS memory via the *stuff cips memory* command. There are two buffers set aside in CIPS memory for the storage of daily command loads. The locations and sizes of these buffers are documented in the 'Data Entities' section of the *CIPS FSW Design Description*[\[3\]](#). There is no limit to the number of command sequences that may be stored in either one of these buffers (beyond the obvious limitation of the buffer size).

Autonomous sequences are initially stored in CIPS EEPROM. These sequences are copied at boot time to operand SRAM. Thus, it is possible to load new autonomous sequences to SRAM for initial testing before committing them to EEPROM. Sequence storage addresses and sizes are documented in the 'Data Entities' section of the *CIPS FSW Design Description*[\[3\]](#).

3.11 Clock Synchronization

Upon receipt of an *inform cips sc_clock* command, instrument software will save the spacecraft time indicated in the command. Then, upon receipt of a SYNC mode code from the Mil-Std-1553 data bus, the current local clock will be updated to the saved spacecraft time from the preceding *inform cips sc_clock* command. This happens once every second. CIPS uses its own internal clock for sub-second timing.

3.12 Hazardous Commands

There are only a handful of commands marked as 'hazardous' in the CIPS command set. These commands are marked as such to force operators to make a decision as to whether or not the command should really be sent. In addition, all hazardous CIPS commands must be immediately preceded by an *arm cips* command, or else they will be rejected by CIPS FSW. Since *arm* commands are not permitted within command sequences, neither are hazardous commands.

There are no features of the fully assembled CIPS instrument that are hazardous to people, and there is only one condition in which the CIPS instrument or other AIM hardware could be damaged due to command receipt: the powering on of the CIPS high voltages while the instrument is under partial vacuum. This condition should only arise for brief periods during thermal-vacuum testing. During this time, the commands to enable CIPS high voltage will be excluded from execution by temporarily removing them from the OASIS-CC command database.

4

Telemetry

4.1 Telemetry Measurement List Fields

Every CIPS FSW telemetry measurement is detailed in Appendix E on page 137. Each listing contains the following fields:

- a brief description of the telemetry measurement,
- the measurement data format (type and size),
- the measurement's root source,
- any state conversions for the measurement's values,
- any analog conversions (linear, exponential, or polynomial expansions),
- any high, low, or delta values that cause cautionary alarms,
- an optional detailed discussion of the telemetry item,
- a listing of the packets which contain this measurement,
- a listing of any related commands,
- a listing of any related telemetry measurements, and
- a listing of any related telemetry packets.

4.2 Telemetry Packet List Fields

Every CIPS FSW telemetry packet is detailed in Appendix F on page 479. Each listing contains the following fields:

- the packet's application identifier (APID),
- the OASIS stream name for the packet,
- a brief description of the telemetry packet,
- an optional detailed discussion of the packet,
- the packet's format (size and decommutation map),
- the format of any sub-packets the packet contains,

- a listing of any related commands,
- a listing of any related telemetry measurements, and
- a listing of any related telemetry packets.

4.3 Telemetry Names

Telemetry names follow the convention specified in the *OASIS-CC System Manager's Guide* [8]. All Telemetry names have the form

element telemetry_item

The convention of this document is to specify telemetry names in lower case letters, though OASIS-CC is case independent.

For all CIPS telemetry, the element is *cips*. Telemetry item names have been chosen to represent as clearly as possible (within OASIS name length restrictions) the data represented by the telemetry value.

4.4 Packet Formats

The AIM telemetry packet format is defined in the AIM Spacecraft to Instruments Communications ICD[4]. All telemetry packets are CCSDS telemetry packets with a 64-bit secondary header containing a timestamp denoting when the packet was created. This timestamp consists of 32 bits of seconds followed by 32 bits of subseconds, identical to the CIPS and spacecraft clocks. Telemetry packets can be up to 1024 bytes in length, including all header fields.

4.5 Packet Corruption

All packets produced by CIPS contain a 16-bit CRC as the last word in the packet. Corrupt CIPS packets can be detected by comparing a CRC of the entire telemetry packet with zero¹. If the CRC is not zero, the packet data was corrupted sometime after packetization by the CIPS instrument.

Further error checking can be done against fields in the CCSDS primary header, including the version number, which should be zero; the packet type, which should also be zero; the packet sequence flags, which should both be set; or the packet length.

4.6 Telemetry Rates and Bandwidth

The rate of production of some CIPS telemetry packets is controllable via the various *set cips *_rt* commands. Most telemetry packets, however, are produced asynchronously whenever needed. This includes all of the science packets, which are produced on a fairly regular basis after science data has been gathered. CIPS exception, post-mortem, and memory dump packets are only produced upon command, and CIPS error packets are only produced in rare occasions when FSW encounters data it can't properly read. CIPS command echo packets are produced every time an external command is received. CIPS event log packets are produced after 45 events have been logged. Note, however, that the types of events that actually make it into the event log is under operator control using the *enable cips em* and *disable cips em* commands.

¹A CRC computed over any length of data appended with the CRC of that data will always result in zero.

All of these packets—even those with rate control commands—are subjected to filtering before being transferred to the spacecraft. This is done using the telemetry output mask within the CIPS FSW parameter table, as described in the 'Telemetry Service' section of the CIPS FSW Design Description[3].

Detailed analysis of the expected CIPS telemetry bandwidth has been done to show that the requirement that CIPS produce at most 1.1 Gbit of telemetry per day will be met. This analysis assumes a particular rate of production of CIPS housekeeping packets, as shown in Table 4.1. The only tricky item in the list is the CIPS Event Log production rate—this will have to be 'tuned' based on the amount of each type of event logged in a nominal day.

Packet	Rate
<i>cips sw_status</i>	60 sec
<i>cips hw_hk</i>	60 sec
<i>cips event_log</i>	15 sec
<i>cips error</i>	none produced
<i>cips cmd_echo</i>	none produced
<i>cips exception</i>	none produced
<i>cips post_mortem</i>	none produced
<i>cips memory_dump</i>	none produced

Table 4.1: CIPS Telemetry Bandwidth Calculation Assumptions

4.7 Memory Dumps

Memory dumps occur only on command, and only contiguous memory addresses may be dumped with a single command. Maximum-sized memory dump packets will be produced once a second until all requested memory has been dumped; the last packet will be only as large as required to complete the requested memory dump. While memory dumps are being produced, any memory dump command processed will be accepted, effectively abandoning the former memory dump. In this way, long memory dumps may be 'canceled' by requesting a new, small memory dump.

4.8 Diagnostic Packets

Aside from the software status packet, CIPS produces three other important diagnostic packets: the event log, post-mortem log, and exception data telemetry packets.

4.8.1 Event Message Log

The event log is produced at regular intervals while CIPS is executing nominally. Events that are logged are described in the 'Logging Service' and 'Data Entities' sections of the CIPS FSW Design Description[3]. As previously mentioned, events may not make it into the log based on their type, with the *enable cips em* and *disable cips em* commands controlling the types of events that do make it into the log.

The information contained in this log is essential for FSW engineers when diagnosing problems with the CIPS instrument.

4.8.2 Post-Mortem Log

The information in the post-mortem log is very similar to the data recorded in the event log. However, this log is written to much more frequently than the event log and thus is only transmitted when requested. The 'Logging Service' section in the CIPS FSW Design Description[3] describes in more detail what data is entered into this log and when. In the event of a CIPS reset, this log is write-protected until operators re-enable it, ensuring that operators get a chance to dump the log before it is overwritten. This has the (intended) side effect of only capturing the data from the first reset should multiple resets occur before operators get a chance to dump the log and re-enable data collection.

The information contained in this log is essential for diagnosing CIPS instrument resets.

Post-mortem logging is enabled/disabled by a memory write command to the 'Post-Mortem Write Enable Flag' data item. The 'Data Entities' section of the CIPS FSW Design Description[3] contains the location of this parameter and the values used to enable/disable logging.

4.8.3 Exception Data

In the event of a CIPS reset, data is read from certain key hardware registers. This data is preserved in a write-protected buffer until operators re-enable writing to the buffer. This ensures that operators get a chance to dump the log before it is overwritten. This has the (intended) side effect of only capturing the data from the first reset should multiple resets occur before operators get a chance to dump the buffer contents and re-enable data collection.

The information contained in this log is essential for diagnosing CIPS instrument resets.

Exception data logging is enabled/disabled by a memory write command to the 'Exception Write Enable Flag' data item. The 'Data Entities' section of the CIPS FSW Design Description[3] contains the location of this parameter and the values used to enable/disable logging.

A

Command Verb Summary

<i>verb phrase</i>	<i>discrete</i>	<i>comment</i>
arm	yes	A prerequisite command for a potentially hazardous command.
close	yes	Mechanically close a device controlled by the command element. <i>close</i> works in conjunction with the <i>open</i> verb.
copy	no	Copy a specific data buffer from one region of memory to another.
disable	yes	Deactivate a capability or function. <i>disable</i> works in conjunction with the <i>enable</i> verb.
dump	no	Copy a specific data buffer of the command element to telemetry.
enable	yes	Activate a capability or function. <i>enable</i> works in conjunction with the <i>disable</i> verb.
inform	no	Inform an object of some current data.
inhibit	yes	Prevent the system from performing a specified action.
init	no	Begin a software function with an initial paramter. <i>init</i> works in conjunction with the <i>term</i> verb.
issue	yes	Send a telemetry packet.
move	no	Move a machanical object from one position to another.
noop	yes	Instruct the command element to perform no operation or 'do nothing'. <i>noop</i> commands are typically used to verify a command path without disturbing configuration.

A. COMMAND VERB SUMMARY

<i>verb phrase</i>	<i>discrete</i>	<i>comment</i>
observe	no	Science observation sequencing command.
open	yes	Mechanically open a device controlled by the command element. <i>open</i> works in conjunction with the <i>close</i> verb.
pass	yes	Pass subfield data unmodified to a secondary hardware or software command receiver.
reset	yes	Restore a given system to its initial configuration.
safe	yes	Configure a device to a safe state. <i>safe</i> works in conjunction with the <i>wake</i> verb.
select	yes	Select one of multiple discrete positions or modes.
set	no	Vary a continuous range parameter.
stuff	no	Load a given data set into memory at a given location.
suspend	yes	Pause command interpretation until an event occurs.
term	yes	End a software function. <i>term</i> works in conjunction with the <i>init</i> verb.
turn off	yes	Remove power from, or deactivate a device. <i>turn off</i> works in conjunction with the <i>turn on</i> verb.
turn on	yes	Apply power to, or activate a device. <i>turn on</i> works in conjunction with the <i>turn off</i> verb.
wake	yes	Wake the hardware from a safe state. <i>wake</i> works in conjunction with the <i>safe</i> verb.

B

Command List

OASIS-CC/FSW database version TBD, Mon Feb 19 11:16:30 2007.

B.1 arm cips

Packet Target Application Identifier

cips, 0x160, (352)

Description

Arm CIPS software for a subsequent hazardous command.

Discussion

There is no time limit for the arm condition. Any other external CIPS (i.e. not from a command sequence) command clears the arm condition; in this case, the command that clears the arm is actually executed, even if it is not a hazardous command. A 'reset cips arm' explicitly clears this arming.

Constraints

This command is not permitted when CIPS is in SAFE mode, nor is it allowed in stored command sequences.

Command Target

software

Format

bit	opcode		subfield(s)	
	0	7	8	15
	0x01		fill	

Command size = 16 bits = 2 bytes.

Subfield(s)

fill: Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

cips armed (148)

B.2 close cips shutter**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Close the electronic shutter on the given camera(s).

Constraints

The 'disable cips shutter' command must have been issued prior to this command in order for the command to work, because the 'cips px_shutr_disable', 'cips my_shutr_disable', 'cips mx_shutr_disable', and/or 'cips py_shutr_disable' must be 'TRUE' for this command to be accepted (only those cameras whose shutters are to be manually closed are checked to ensure that autonomous shutter control has been disabled).

This command is only valid when CIPS is executing SRAM Code.

Command Target

software

Format

opcode	subfield(s)
0x35	cameras

Command size = 16 bits = 2 bytes.

bit 0 7 8 15

Subfield(s)

cameras: Camera(s) whose shutter(s) will be closed.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	1
					PX	1
					PY	2
					MX	4
					X	5
					MY	8
					Y	10
					ALL	15

Safety Level

SAFE

Telemetry Verification

Related Commands

<i>command</i>	<i>description</i>
open cips shutter (87)	Open the electronic shutter on the given camera(s).
disable cips shutter (54)	Disable the autonomous opening and closing of the given shutter(s).
enable cips shutter (68)	Enable the autonomous opening and closing of the given shutter(s).

B.3 copy cips memory**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Copy data from one location to another.

Discussion

Copy data contained in memory or I/O space to another location in memory or I/O space.

Constraints

The length of the copy cannot exceed the amount of space available on the page, i.e. the address plus the length must not exceed 65536 for either the 'from' or 'to' page.

Command Target

software

Format

opcode		subfield(s)					
0x09		fill	from_page	to_page	from_addr	to_addr	length
bit	0 7	8 15	16 31	32 47	48 63	64 79	80 95

Command size = 96 bits = 12 bytes.

Subfield(s)**fill:** Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

from_page: Memory type/page specifier for buffer to copy from.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	2047	default	0

from_page discussion:

Pages in memory or I/O space are allocated as follows: Page 0 is stack space in operand SRAM, pages 1-7 are in nominal operand SRAM, pages 8-11 are in communications operand SRAM, pages 12-15 are in EEPROM, page 128 is I/O space, pages 256-272 are instruction SRAM, and pages 1024-2047 are SDRAM. Each page contains 32768 words addressed from 0x8000 to 0xFFFF and addresses 0x0000 to 0x7FFF are the same as page 0, with the exception of instruction memory and SDRAM, which have 65536 word pages occupying the full address space (0x0000 to 0xFFFF).

to_page: Memory type/page specifier for buffer to copy to.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	2047	default	6

to_page discussion:

Pages in memory or I/O space are allocated as follows: Page 0 is stack space in operand SRAM, pages 1-7 are in nominal operand SRAM, pages 8-11 are in communications operand SRAM, pages 12-15 are in EEPROM, page 128 is I/O space, pages 256-272 are instruction SRAM, and pages 1024-2047 are SDRAM. Each page contains 32768 words addressed from 0x8000 to 0xFFFF and addresses 0x0000 to 0x7FFF are the same as page 0, with the exception of instruction memory and SDRAM, which have 65536 word pages occupying the full address space (0x0000 to 0xFFFF).

from_addr: Starting address for buffer to copy from.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	65535	default	0

to_addr: Starting address for buffer to copy to.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	65535	default	32768

length: Length of data to copy.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	65535	default	1

Safety Level

SAFE

Telemetry Verification

cips cmd_succ_ct (175)

Related Commands

<i>command</i>	<i>description</i>
dump cips memory (57)	Dump CIPS memory.
init cips crc (74)	Checksum CIPS SRAM or EEPROM memory.
stuff cips memory (112)	Load memory.
stuff cips memory_wd (113)	Load a single 16-bit word of data to memory or I/O space.

B.4 disable cips bos

Packet Target Application Identifier

cips, 0x160, (352)

Description

Disable the CIPS bright object sensor.

Constraints

This command must be immediately preceeded by an 'arm' command and is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

bit	opcode		subfield(s)	
	0	7	8	15
	0x82		cameras	

Command size = 16 bits = 2 bytes.

Subfield(s)

cameras: Camera(s) to configure.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	1
					PX	1
					PY	2
					MX	4
					X	5
					MY	8
					Y	10
					ALL	15

Safety Level

HAZARDOUS

Telemetry Verification

cips py_bos_ena (409), cips px_bos_ena (389), cips my_bos_ena (311), cips mx_bos_ena (291)

B.5 disable cips cam_bin**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Disable the cameras' science data binning capability.

Discussion

The camera binning setting is identical across all CIPS cameras. With this binning enabled, image pixels are binned 4 by 2 before they are stored in SDRAM and processed by FSW.

Constraints

This command is only allowed when CIPS is executing SRAM Code and is not in the IMAGING state.

Command Target

software

Format

opcode	subfield(s)	
	0x33	fill
bit	0	7 8 15

Command size = 16 bits = 2 bytes.

Subfield(s)

fill: Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

cips cam_bin_enable (151)

Related Commands

<i>command</i>	<i>description</i>
disable cips comp (49)	Disable science data compression capability in CIPS FSW.
disable cips sw_bin (55)	Disable science data binning capability in CIPS FSW.
enable cips cam_bin (60)	Enable the cameras' science data binning capability.
enable cips comp (61)	Enable science data compression capability in CIPS FSW.
enable cips sw_bin (69)	Enable science data binning capability in CIPS FSW.

Related Telemetry Packets

<i>packet</i>	<i>description</i>
cips sci_px_cam (508)	CIPS +X camera science data packet.
cips sci_py_cam (511)	CIPS +Y camera science data packet.
cips sci_mx_cam (502)	CIPS -X camera science data packet.
cips sci_my_cam (505)	CIPS -Y camera science data packet.

B.6 disable cips cde_uart**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Disable communications between the CIPS and CDE instruments.

Constraints

This command must be immediately preceeded by an 'arm' command and is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

opcode		subfield(s)	
0x51		fill	
bit	0	7	15

Command size = 16 bits = 2 bytes.

Subfield(s)

fill: Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

HAZARDOUS

Telemetry Verification

cips_cde_uart_rxor_mk (163), cips_cde_uart_rx_r_mk (162), cips_cde_uart_tx_e_mk (165)

B.7 disable cips comp**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Disable science data compression capability in CIPS FSW.

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

opcode		subfield(s)	
0x2f		fill	
bit	0	7	15

Command size = 16 bits = 2 bytes.

Subfield(s)

fill: Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

cips_sw_comp_enable (463)

Related Commands

<i>command</i>	<i>description</i>
disable cips cam_bin (47)	Disable the cameras' science data binning capability.
disable cips sw_bin (55)	Disable science data binning capability in CIPS FSW.
enable cips cam_bin (60)	Enable the cameras' science data binning capability.
enable cips comp (61)	Enable science data compression capability in CIPS FSW.
enable cips sw_bin (69)	Enable science data binning capability in CIPS FSW.

Related Telemetry Packets

<i>packet</i>	<i>description</i>
cips sci_px_cam (508)	CIPS +X camera science data packet.
cips sci_py_cam (511)	CIPS +Y camera science data packet.
cips sci_mx_cam (502)	CIPS -X camera science data packet.
cips sci_my_cam (505)	CIPS -Y camera science data packet.

B.8 disable cips ee_write**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Disable writes to CIPS EEPROM.

Discussion

Disable writes to EEPROM. Any pending write protect timeout that auto-disables write protect is canceled by this command.

Command Target

software

Format

	opcode	subfield(s)	
	0x05	fill	Command size = 16 bits = 2 bytes.
bit	0	7 8 15	

Subfield(s)

fill: Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

cips ee_write_enable (220)

Related Commands

<i>command</i>	<i>description</i>
enable cips ee.write (62)	Enable writes to CIPS EEPROM.
stuff cips memory.wd (113)	Load a single 16-bit word of data to memory or I/O space.

B.9 disable cips em**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Disable given messages in the event log packet.

Command Target

software

Format

bit	opcode		subfield(s)			
	0	7	8	15	16	47
	0x76		fill	message		

Command size = 48 bits = 6 bytes.

Subfield(s)**fill:** Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

message: Event message to disable in the downlink stream.

bits	type	eng units	min (dn)	max (dn)	state	value
32	discrete state	any	0	4294967295	default	0
					INFO	1
					POWER	2
					CMD_RCVD	4
					TLM_SENT	8
					ACS_RCVD	16
					TIME_RCVD	32
					CDE_TLM_RCVD	64
					CDE_CMD_SENT	128
					SEQ_CMD_RCVD	256
					MODE_CHNG	65536
					SEQ_MODE_CHNG	131072
					WARNING	2097152
					SEQ_ERROR	33554432
					CMD_ERROR	67108864
					TLM_ERROR	134217728
					ERROR	268435456
					DEBUG	1073741824

Safety Level

SAFE

Telemetry Verification

cips em_warn_msg (233), cips em_seq_err (229), cips em_debug (224), cips em_tlm_err (232), cips em_info (226), cips em_pwr (227), cips em_u_test (233), cips em_cmd_err (223), cips em_mode_chng (227), cips em_tlm_tx (232), cips em_time_rx (231), cips em_acs_stat_rx (221), cips em_cde_cmd_tx (222), cips em_seq_mode_chng (230), cips em_cmd_rx (223), cips em_err (225), cips em_cde_tlm_rx (222)

Related Commands

<i>command</i>	<i>description</i>
disable cips em (51)	Disable given messages in the event log packet.
enable cips em (63)	Enable given messages in the event log packet.

Related Telemetry Packets

<i>packet</i>	<i>description</i>
cips evt_log (482)	cips event log packet.

B.10 disable cips lens_htr**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Enable the CIPS lens heater for the given camera(s).

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

opcode	subfield(s)	Command size = 16 bits = 2 bytes.
0x87	cameras	
bit 0	7 8 15	

Subfield(s)**cameras:** Camera(s) to configure.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	1
					PX	1
					PY	2
					MX	4
					X	5
					MY	8
					Y	10
					ALL	15

Safety Level

SAFE

Telemetry Verification

cips my_lens_htr_ena (324), cips px_lens_htr_ena (403), cips mx_lens_htr_ena (305), cips py_lens_htr_ena (423)

B.11 disable cips sdp**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Disable CIPS science data processing and downlink.

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

opcode		subfield(s)	
0x23		fill	
bit	0	7	15

Command size = 16 bits = 2 bytes.

Subfield(s)**fill:** Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

cips sdp_enable (434)

Related Commands

<i>command</i>	<i>description</i>
enable cips sdp (66)	Enable CIPS science data processing and downlink.

Related Telemetry Packets

<i>packet</i>	<i>description</i>
cips sci_px_cam (508)	CIPS +X camera science data packet.
cips sci_py_cam (511)	CIPS +Y camera science data packet.
cips sci_mx_cam (502)	CIPS -X camera science data packet.
cips sci_my_cam (505)	CIPS -Y camera science data packet.

B.12 disable cips shutter

Packet Target Application Identifier

cips, 0x160, (352)

Description

Disable the autonomous opening and closing of the given shutter(s).

Discussion

This command enables the manual opening and closing of the CIPS shutters via the 'open cips shutter' and 'close cips shutter' commands.

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

opcode	subfield(s)
0x39	cameras

Command size = 16 bits = 2 bytes.

bit 0 7 8 15

Subfield(s)

cameras: Camera(s) whose shutter action will be disabled.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	1
					PX	1
					PY	2
					MX	4
					X	5
					MY	8
					Y	10
					ALL	15

Safety Level

SAFE

Telemetry Verification

cips px_shutr_disable (408), cips my_shutr_disable (328), cips mx_shutr_disable (310), cips py_shutr_disable (427)

Related Commands

command	description
close cips shutter (44)	Close the electronic shutter on the given camera(s).
open cips shutter (87)	Open the electronic shutter on the given camera(s).
enable cips shutter (68)	Enable the autonomous opening and closing of the given shutter(s).

B.13 disable cips sw_bin

Packet Target Application Identifier

cips, 0x160, (352)

Description

Disable science data binning capability in CIPS FSW.

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

bit	opcode		subfield(s)	
	0x31		fill	
0	7	8	15	

Command size = 16 bits = 2 bytes.

Subfield(s)

fill: Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

cips sw_bin.enable (463)

Related Commands

<i>command</i>	<i>description</i>
disable cips cam_bin (47)	Disable the cameras' science data binning capability.
disable cips comp (49)	Disable science data compression capability in CIPS FSW.
enable cips cam_bin (60)	Enable the cameras' science data binning capability.
enable cips comp (61)	Enable science data compression capability in CIPS FSW.
enable cips sw_bin (69)	Enable science data binning capability in CIPS FSW.

Related Telemetry Packets

<i>packet</i>	<i>description</i>
cips sci_px_cam (508)	CIPS +X camera science data packet.
cips sci_py_cam (511)	CIPS +Y camera science data packet.
cips sci_mx_cam (502)	CIPS -X camera science data packet.
cips sci_my_cam (505)	CIPS -Y camera science data packet.

B.14 disable cips tlm_out

Packet Target Application Identifier

cips, 0x160, (352)

Description

Disable downlink of the given CIPS telemetry packet.

Discussion

Disabling telemetry output for a given packet is identical from the operator's perspective as setting the packet production rate to zero. However, not all CIPS and CDE packet rates can be controlled via 'set rate' commands. Also, FSW engineers may desire packets to be created but not downlinked, which is why this command exists.

Command Target

software

Format

	opcode		subfield(s)			
	0x79		fill	packet		
bit	0	7	8	15	16	47

Command size = 48 bits = 6 bytes.

Subfield(s)

fill: Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

packet: Packet to disable in the downlink stream.

bits	type	eng units	min (dn)	max (dn)	state	value
32	discrete state	any	0	4194304	default	0
					SW_STAT	1
					HW_HK	2
					ERR	4
					EVT_LOG	8
					MEM_DUMP	16
					POST_MORTEM	32
					EXCEPTION	64
					CMD_ECHO	128
					SCI_PX_CAM	256
					SCI_PY_CAM	512
					SCI_NX_CAM	1024
					SCI_NY_CAM	2048
					RESERVED_1	4096
					RESERVED_2	8192
					RESERVED_3	16384
					RESERVED_4	32768
					CDE_SCI	65536
					CDE_MEM_DUMP	131072
					CDE_SYS_DATA	262144
					CDE_HISTORY	524288
					CDE_CAL	1048576
					CDE_FLASH	2097152
					CDE_ITF	4194304

Safety Level

SAFE

Telemetry Verification

cips reserved_4_to (430), cips reserved_1_to (428), cips cde_mem_dump_to (158), cips cde_flash_to (155), cips cmd_echo_to (171), cips cde_hist_log_to (156), cips err_to (234), cips pm_to (383), cips reserved_3_to (429), cips mem_dump_to (271), cips cde_itf_to (157), cips evt_log_to (238), cips reserved_2_to (429), cips cde_sci_to (158), cips cde_sys_data_to (159), cips cde_cal_to (154), cips hw_hk_to (253), cips exception_to (239), cips sw_stat_to (464), cips sci_mx_cam_to (432), cips sci_my_cam_to (432), cips sci_py_cam_to (434), cips sci_px_cam_to (433)

Related Commands

<i>command</i>	<i>description</i>
disable cips tlm.out (56)	Disable downlink of the given CIPS telemetry packet.
set cips to_mask (110)	Set the CIPS telemetry output enable mask.
enable cips tlm.out (70)	Enable downlink of the given CIPS telemetry packet.

B.15 dump cips memory

Packet Target Application Identifier

cips, 0x160, (352)

Description

Dump CIPS memory.

Discussion

Dump data contained in memory or I/O space.

A single instance of this command may result in many memory dump telemetry packets over a period of time. The associated telemetry packet rate specifies the frequency to emit memory dump packets, and should be compatible with the available downlink telemetry bandwidth.

Constraints

The memory to be dumped must fit completely within a page boundary—the sum of the address and the length must not exceed 65536.

Command Target

software

Format

bit	opcode		subfield(s)									
	0x7c		fill	page	addr	length						
	0	7	8	15	16	31	32	47	48	63		

Command size = 64 bits = 8 bytes.

Subfield(s)

fill: Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

page: Memory dump type/page specifier.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	2047	default	0

page discussion:

Pages in memory or I/O space are allocated as follows: Page 0 is stack space in operand SRAM, pages 1-7 are in nominal operand SRAM, pages 8-11 are in communications operand SRAM, pages 12-15 are in EEPROM, page 128 is I/O space, pages 256-272 are instruction SRAM, and pages 1024-2047 are SDRAM. Each page contains 32768 words addressed from 0x8000 to 0xFFFF and addresses 0x0000 to 0x7FFF are the same as page 0, with the exception of instruction memory and SDRAM, which have 65536 word pages occupying the full address space (0x0000 to 0xFFFF).

addr: Memory dump starting address.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	65535	default	0

length: Memory dump length.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	65535	default	4

Safety Level

SAFE

Telemetry Verification

cips_dump_length (206), **cips_dump_addr** (205), **cips_dump_data** (205), **cips_mem_dump_remain** (270)

Related Commands

<i>command</i>	<i>description</i>
stuff cips memory (112)	Load memory.
copy cips memory (45)	Copy data from one location to another.
stuff cips memory_wd (113)	Load a single 16-bit word of data to memory or I/O space.

Related Telemetry Packets

<i>packet</i>	<i>description</i>
cips mem.dump (499)	cips memory dump packet.

B.16 enable cips bos**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Enable the CIPS bright object sensor.

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

opcode	subfield(s)
0x81	cameras
bit 0 7 8 15	Command size = 16 bits = 2 bytes.

Subfield(s)**cameras:** Camera(s) to configure.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	1
					PX	1
					PY	2
					MX	4
					X	5
					MY	8
					Y	10
					ALL	15

Safety Level

SAFE

Telemetry Verification**cips py_bos_ena** (409), **cips px_bos_ena** (389), **cips my_bos_ena** (311), **cips mx_bos_ena** (291)

B.17 enable cips cam_bin

Packet Target Application Identifier

cips, 0x160, (352)

Description

Enable the cameras' science data binning capability.

Discussion

The camera binning setting is identical across all CIPS cameras. With this binning enabled, image pixels are binned 4 by 2 before they are stored in SDRAM and processed by FSW.

Constraints

This command is only allowed when CIPS is executing SRAM Code and is not in the IMAGING state.

Command Target

software

Format

bit	opcode		subfield(s)	
	0	7	8	15
	0x32		fill	

Command size = 16 bits = 2 bytes.

Subfield(s)

fill: Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

cips cam_bin_enable (151)

Related Commands

command	description
disable cips cam_bin (47)	Disable the cameras' science data binning capability.
disable cips comp (49)	Disable science data compression capability in CIPS FSW.
disable cips sw_bin (55)	Disable science data binning capability in CIPS FSW.
enable cips comp (61)	Enable science data compression capability in CIPS FSW.
enable cips sw_bin (69)	Enable science data binning capability in CIPS FSW.

Related Telemetry Packets

packet	description
cips sci_px_cam (508)	CIPS +X camera science data packet.
cips sci_py_cam (511)	CIPS +Y camera science data packet.
cips sci_mx_cam (502)	CIPS -X camera science data packet.
cips sci_my_cam (505)	CIPS -Y camera science data packet.

B.18 enable cips cde_uart

Packet Target Application Identifier

cips, 0x160, (352)

Description

Enable communications between the CIPS and CDE instruments.

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

opcode		subfield(s)	
0x50		fill	
bit	0	7	15

Command size = 16 bits = 2 bytes.

Subfield(s)

fill: Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

cips cde_uart_rxor_mk (163), cips cde_uart_rx_r_mk (162), cips cde_uart_tx_e_mk (165)

B.19 enable cips comp

Packet Target Application Identifier

cips, 0x160, (352)

Description

Enable science data compression capability in CIPS FSW.

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

	opcode		subfield(s)	
	0x2e		fill	
bit	0	7	8	15

Command size = 16 bits = 2 bytes.

Subfield(s)

fill: Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

cips sw_comp_enable (463)

Related Commands

<i>command</i>	<i>description</i>
disable cips cam_bin (47)	Disable the cameras' science data binning capability.
disable cips comp (49)	Disable science data compression capability in CIPS FSW.
disable cips sw_bin (55)	Disable science data binning capability in CIPS FSW.
enable cips cam_bin (60)	Enable the cameras' science data binning capability.
enable cips sw_bin (69)	Enable science data binning capability in CIPS FSW.

Related Telemetry Packets

<i>packet</i>	<i>description</i>
cips sci_px_cam (508)	CIPS +X camera science data packet.
cips sci_py_cam (511)	CIPS +Y camera science data packet.
cips sci_mx_cam (502)	CIPS -X camera science data packet.
cips sci_my_cam (505)	CIPS -Y camera science data packet.

B.20 enable cips ee_write**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Enable writes to CIPS EEPROM.

Discussion

EEPROM contents are protected by this write enable, and cannot be altered unless this command is sent first. Writes can be disabled by the 'disable memory ee_write' command.

This command must be immediately preceded by an 'arm cips' command.

Constraints

This command will be rejected unless immediately preceded by an 'arm' command.

EEPROM contents should not be altered without prior authorization and an appropriate level of ground simulation and testing.

Command Target

software

Format

opcode		subfield(s)	
0x04		fill	
bit	0	7	15

Command size = 16 bits = 2 bytes.

Subfield(s)**fill:** Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

HAZARDOUS

Telemetry Verification

cips ee.write.enable (220)

Related Commands

<i>command</i>	<i>description</i>
disable cips ee.write (50)	Disable writes to CIPS EEPROM.
stuff cips memory_wd (113)	Load a single 16-bit word of data to memory or I/O space.

B.21 enable cips em**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Enable given messages in the event log packet.

Command Target

software

Format

opcode		subfield(s)	
0x75		fill	message
bit	0	7	47

Command size = 48 bits = 6 bytes.

Subfield(s)**fill:** Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

message: Event message to enable in the downlink stream.

bits	type	eng units	min (dn)	max (dn)	state	value
32	discrete state	any	0	4294967295	default	0
					INFO	1
					POWER	2
					CMD_RCVD	4
					TLM_SENT	8
					ACS_RCVD	16
					TIME_RCVD	32
					CDE_TLM_RCVD	64
					CDE_CMD_SENT	128
					SEQ_CMD_RCVD	256
					MODE_CHNG	65536
					SEQ_MODE_CHNG	131072
					WARNING	2097152
					SEQ_ERROR	33554432
					CMD_ERROR	67108864
					TLM_ERROR	134217728
					ERROR	268435456
					DEBUG	1073741824

Safety Level

SAFE

Telemetry Verification

cips em_warn_msg (233), cips em_seq_err (229), cips em_debug (224), cips em_tlm_err (232), cips em_info (226), cips em_pwr (227), cips em_u_test (233), cips em_cmd_err (223), cips em_mode_chng (227), cips em_tlm_tx (232), cips em_time_rx (231), cips em_acs_stat_rx (221), cips em_cde_cmd_tx (222), cips em_seq_mode_chng (230), cips em_cmd_rx (223), cips em_err (225), cips em_cde_tlm_rx (222)

Related Commands

<i>command</i>	<i>description</i>
disable cips em (51)	Disable given messages in the event log packet.

Related Telemetry Packets

<i>packet</i>	<i>description</i>
cips evt_log (482)	cips event log packet.

B.22 enable cips_exc_pm

Packet Target Application Identifier

cips, 0x160, (352)

Description

Re-enable CIPS exception and post-mortem data recording.

Discussion

This command is only needed to recover from a software exception event.

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

opcode		subfield(s)	
0x02		fill	
bit	0	7	15

Command size = 16 bits = 2 bytes.

Subfield(s)

fill: Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification**Related Telemetry**

telemetry *description*

B.23 enable cips lens_htr**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Enable the CIPS lens heater for the given camera(s).

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

opcode		subfield(s)	
0x86		cameras	
bit	0	7	15

Command size = 16 bits = 2 bytes.

Subfield(s)

cameras: Camera(s) to configure.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	1
					PX	1
					PY	2
					MX	4
					X	5
					MY	8
					Y	10
					ALL	15

Safety Level

SAFE

Telemetry Verification

cips my_lens_htr_ena (324), cips px_lens_htr_ena (403), cips mx_lens_htr_ena (305), cips py_lens_htr_ena (423)

B.24 enable cips sdp

Packet Target Application Identifier

cips, 0x160, (352)

Description

Enable CIPS science data processing and downlink.

Discussion

This command allows the CIPS instrument to process science data once it has been collected from the CIPS CCDs. Processing will commence upon the next receipt of a science data ready signal from the CIPS CCD hardware. As soon as all of that data has been processed, CIPS FSW will automatically disable science data processing.

Constraints

This command is only allowed when CIPS is executing SRAM Code and is not in the IMAGING state.

Command Target

software

Format

opcode	subfield(s)
0x22	fill
bit 0 7 8 15	

Command size = 16 bits = 2 bytes.

Subfield(s)

fill: Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

cips sdp_enable (434)

Related Commands

<i>command</i>	<i>description</i>
disable cips sdp (53)	Disable CIPS science data processing and downlink.

Related Telemetry Packets

<i>packet</i>	<i>description</i>
cips sci_px_cam (508)	CIPS +X camera science data packet.
cips sci_py_cam (511)	CIPS +Y camera science data packet.
cips sci_mx_cam (502)	CIPS -X camera science data packet.
cips sci_my_cam (505)	CIPS -Y camera science data packet.

B.25 enable cips sdram_ud**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Re-enable CIPS SDRAM current monitor telemetry.

Discussion

This command is only needed to recover from a SDRAM power off due to an SDRAM overcurrent condition.

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

hk fpga

Format

opcode	subfield(s)
0x2c	fill
bit 0 7 8 15	Command size = 16 bits = 2 bytes.

Subfield(s)**fill:** Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

Related Telemetry

<i>telemetry</i>	<i>description</i>
------------------	--------------------

B.26 enable cips shutter**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Enable the autonomous opening and closing of the given shutter(s).

Discussion

This disables the manual operation of the shutters (i.e. using the 'open cips shutter' and 'close cips shutter' commands).

Constraints

This command is only allowed when CIPS is executing SRAM Code and is NOT in SAFE mode.

Command Target

software

Format

opcode	subfield(s)
0x3a	cameras

Command size = 16 bits = 2 bytes.

bit 0 7 8 15

Subfield(s)**cameras:** Camera(s) whose shutter action will be enabled.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	1
					PX	1
					PY	2
					MX	4
					X	5
					MY	8
					Y	10
					ALL	15

Safety Level

SAFE

Telemetry Verification

cips px_shutr_disable (408), cips my_shutr_disable (328), cips mx_shutr_disable (310), cips py_shutr_disable (427)

Related Commands

<i>command</i>	<i>description</i>
close cips shutter (44)	Close the electronic shutter on the given camera(s).
open cips shutter (87)	Open the electronic shutter on the given camera(s).
disable cips shutter (54)	Disable the autonomous opening and closing of the given shutter(s).

B.27 enable cips sw_bin**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Enable science data binning capability in CIPS FSW.

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

	opcode	subfield(s)	
	0x30	fill	
bit	0	7	8 15

Command size = 16 bits = 2 bytes.

Subfield(s)**fill:** Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification**cips sw_bin_enable** (463)**Related Commands**

<i>command</i>	<i>description</i>
disable cips cam_bin (47)	Disable the cameras' science data binning capability.
disable cips comp (49)	Disable science data compression capability in CIPS FSW.
disable cips sw_bin (55)	Disable science data binning capability in CIPS FSW.
enable cips cam_bin (60)	Enable the cameras' science data binning capability.
enable cips comp (61)	Enable science data compression capability in CIPS FSW.

Related Telemetry Packets

<i>packet</i>	<i>description</i>
cips sci_px_cam (508)	CIPS +X camera science data packet.
cips sci_py_cam (511)	CIPS +Y camera science data packet.
cips sci_mx_cam (502)	CIPS -X camera science data packet.
cips sci_my_cam (505)	CIPS -Y camera science data packet.

B.28 enable cips tlm_out**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Enable downlink of the given CIPS telemetry packet.

Command Target

software

Format

bit	opcode		subfield(s)			
	0	7	8	15	16	47
	0x78		fill		packet	

Command size = 48 bits = 6 bytes.

Subfield(s)**fill:** Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

packet: Packet to enable in the downlink stream.

bits	type	eng units	min (dn)	max (dn)	state	value
32	discrete state	any	0	4194304	default	0
					SW_STAT	1
					HW_HK	2
					ERR	4
					EVT_LOG	8
					MEM_DUMP	16
					POST_MORTEM	32
					EXCEPTION	64
					CMD_ECHO	128
					SCI_PX_CAM	256
					SCI_PY_CAM	512
					SCI_NX_CAM	1024
					SCI_NY_CAM	2048
					RESERVED_1	4096
					RESERVED_2	8192
					RESERVED_3	16384
					RESERVED_4	32768
					CDE_SCI	65536
					CDE_MEM_DUMP	131072
					CDE_SYS_DATA	262144
					CDE_HISTORY	524288
					CDE_CAL	1048576
					CDE_FLASH	2097152
					CDE_ITF	4194304

Safety Level

SAFE

Telemetry Verification

cips reserved_4_to (430), cips reserved_1_to (428), cips cde_mem_dump_to (158), cips cde_flash_to (155), cips cmd_echo_to (171), cips cde_hist_log_to (156), cips err_to (234), cips pm_to (383), cips reserved_3_to (429), cips mem_dump_to (271), cips cde_itf_to (157), cips evt_log_to (238), cips reserved_2_to (429), cips cde_sci_to (158), cips cde_sys_data_to (159), cips cde_cal_to (154), cips hw_hk_to (253), cips exception_to (239), cips sw_stat_to (464), cips sci_mx_cam_to (432), cips sci_my_cam_to (432), cips sci_py_cam_to (434), cips sci_px_cam_to (433)

Related Commands

<i>command</i>	<i>description</i>
disable cips tlm_out (56)	Disable downlink of the given CIPS telemetry packet.
set cips to_mask (110)	Set the CIPS telemetry output enable mask.

B.29 inform cips acs_state

Packet Target Application Identifier

cips, 0x160, (352)

Description

Spacecraft ACS status update command.

Discussion

Inform the instruments of the current spacecraft attitude control subsystem (ACS) status.

Constraints

This command does not directly command or reconfigure science instruments. Though each instrument is programmable to respond to different ACS status values under multiple conditions.

Command Target

software

Format

opcode		subfield(s)									
0xfe		filler_1	filler_2	filler_3	seconds	subseconds	sc_to_sun_u_x	sc_to_sun_u_y	sc_to_sun_u_z	filler_4	
bit	0 7	8 15	16 47	48 79	80 111	112 143	144 207	208 271	272 335	336 447	

Command size = 448 bits = 56 bytes.

Subfield(s)

filler_1: Filler data to match OSC command structure.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	0	default	0

filler_2: Filler data to match OSC command structure.

bits	type	eng units	min (dn)	max (dn)	state	value
32	fill	any	0	0	default	0

filler_3: Filler data to match OSC command structure.

bits	type	eng units	min (dn)	max (dn)	state	value
32	fill	any	0	0	default	0

seconds: Seconds field of timestamp.

bits	type	eng units	min (dn)	max (dn)	state	value
32	unsigned integer	any	0	4294967295	default	0

subseconds: Fractional part of timestamp.

bits	type	eng units	min (dn)	max (dn)	state	value
32	unsigned integer	any	0	4294967295	default	0

sc_to_sun_u_x: Spacecraft to sun unit vector, spacecraft body frame, cartesian x coordinate.

bits	type	eng units	min (dn)	max (dn)	state	value
64	float	any	0	0	default	0

sc_to_sun_u_y: Spacecraft to sun unit vector, spacecraft body frame, cartesian y coordinate.

bits	type	eng units	min (dn)	max (dn)	state	value
64	float	any	0	1	default	0

sc_to_sun_u_z: Spacecraft to sun unit vector, spacecraft body frame, cartesian z coordinate.

bits	type	eng units	min (dn)	max (dn)	state	value
64	float	any	0	1	default	0

filler_4: Filler data to match OSC command structure.

bits	type	eng units	min (dn)	max (dn)	state	value
112	fill	any	0	0	default	0

Safety Level

SAFE

Telemetry Verification

cips acs_sun_uv.y (146), cips acs_sun_uv.z (146), cips acs_sun_uv.x (145)

B.30 inform cips pwrdown

Packet Target Application Identifier

cips, 0x160, (352)

Description

CIPS powerdown warning

Command Target

software

Format

bit	opcode		subfield(s)		Command size = 16 bits = 2 bytes.
	0	7	8	15	
	0xfd		fill		

Subfield(s)

fill: Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

B.31 inform cips sc_clock

Packet Target Application Identifier

cips, 0x160, (352)

Description

Notify FSW of spacecraft time at next time sync.

Discussion

Notify FSW of the spacecraft clock time at the next sync mode code upon the 1553 bus.

Constraints

This command only exists for FSW definition purposes, as this command is received by the CIPS FSW from the S/C bus once every second. Since the receipt of this command from the S/C bus will always occur after any command processing is done, even if this command were sent, the S/C command would override any previously commanded value before the time is applied to the clock.

Command Target

software

Format

opcode		subfield(s)						
0xfc		fill	seconds	subseconds				
bit	0	7	8	15	16	47	48	79

Command size = 80 bits = 10 bytes.

Subfield(s)

fill: Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

seconds: Integral seconds portion of spacecraft clock.

bits	type	eng units	min (dn)	max (dn)	state	value
32	unsigned integer	any	0	4294967295	default	0

subseconds: Integral subseconds portion of spacecraft clock.

bits	type	eng units	min (dn)	max (dn)	state	value
32	unsigned integer	any	0	4294967295	default	0

subseconds discussion:

Each bit represents 2 to the -32 seconds.

Safety Level

SAFE

Telemetry Verification

cips p352hws (341), cips p352lws (342)

B.32 init cips crc**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Checksum CIPS SRAM or EEPROM memory.

Discussion

This performs a CRC16 computation on the specified memory contents, returning the CRC value in telemetry.

Note that this capability cannot execute on SDRAM.

Constraints

The memory location to CRC must fit within page boundaries—the sum of the address and length must not exceed 65536.

Command Target

software

Format

bit	opcode		subfield(s)							
	0x07	fill	page	addr	length	Command size = 64 bits = 8 bytes.				
	0	7	8	15	16	31	32	47	48	63

Subfield(s)

fill: Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	259	default	0

page: Memory type/page to checksum.

bits	type	eng units	min (dn)	max (dn)	state	value
16	discrete state	any	0	271	default	0

addr: Memory checksum starting address.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	65535	default	0

length: Length of memory to checksum, in words.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	65535	default	4

Safety Level

SAFE

Telemetry Verification

cips crc_type (198), cips crc_addr (197), cips crc (179), cips crc_valid (198), cips crc_length (197)

Related Commands

command	description
copy cips memory (45)	Copy data from one location to another.

B.33 init cips imaging

Packet Target Application Identifier

cips, 0x160, (352)

Description

Immediately begin taking images with the indicated CIPS CCDs.

Constraints

This command is only allowed when CIPS is executing SRAM Code and is in the STANDBY state.

Since CIPS images are stored in SDRAM, the 'init imaging' command will also be rejected unless both banks of CIPS SDRAM are powered on. Likewise, this command will be rejected if any of the cameras with which images are to be collected are not powered on.

For images to be taken with one or more cameras in TDI mode AND one or more cameras in snapshot mode, the cameras must be configured with integration frames that are of identical durations. This integration frame time for cameras in TDI mode is calculated by multiplying the camera's TDI row count setting with its TDI dwell time setting then adding the camera's frame sync time setting. For cameras in snapshot mode, the integration frame time is computed by simply adding the camera's integration time and frame sync time settings. The 'init imaging' command will be rejected if the integration frame times are not identical for all cameras to be used in that observation.

Cameras used in the observation must also be configured with appropriate imaging areas—the programmed image window must fit within the programmed common area window as specified by the 'set cmn_a_size', 'set cmn_a_offset', 'set img_size', and 'set img_offset' commands. The 'init imaging' command will be rejected unless these conditions are met.

If camera binning is enabled at the time of the 'init imaging' command processing, the number of columns in the image must be divisible by four and the number of rows by two. Otherwise, the command will be rejected. This same logic does not apply to Software binning; however, it should be noted that if software binning is enabled and the image size after any camera binning has been performed is not evenly divisible (number of rows and columns must be even for the +/- Y cameras and the number of rows must be divisible by four for the +/- X cameras), the last row(s) or column(s) in the image will be cropped from the final image.

Since images are buffered in SDRAM and the mapping of how images are stored in SDRAM changes with image size, CIPS will reject any 'init imaging' command that would result in images that would overwrite unprocessed images currently residing in SDRAM (i.e. images that are bigger than those previously taken). Thus, if the 'img_taken' telemetry item is greater than zero and the size of the images being requested is larger than the size of the last image taken, the command will be rejected.

Finally, CIPS can only store in memory the metadata from 75 images. Thus, if CIPS will reject an 'init imaging' command if the number of images requested plus the number of images currently being processed (as shown by the 'cips img_taken' telemetry point) is greater than 75.

Command Target

software

Format

opcode		subfield(s)	
0x21		cameras	img_ct
bit	0 7 8 15 16 31	Command size = 32 bits = 4 bytes.	

Subfield(s)

cameras: Camera(s) with which data will be taken.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	1
					PX	1
					PY	2
					MX	4
					X	5
					MY	8
					Y	10
					ALL	15

img.ct: Number of images to take (per camera).

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	75	default	0

Safety Level

SAFE

Telemetry Verification

cips mode (271), cips img_to_be_taken (265), cips img_with_py_cam (266), cips img_with_mx_cam (265), cips img_with_px_cam (266), cips img_with_my_cam (265)

B.34 init cips sequence

Packet Target Application Identifier

cips, 0x160, (352)

Description

Start a CIPS command sequence.

Command Target

software

Format

opcode		subfield(s)						
0x63		fill	page	addr				
bit	0	7	8	15	16	31	32	47

Command size = 48 bits = 6 bytes.

Command size = 48 bits = 6 bytes.

Subfield(s)

fill: Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

page: Page in memory where sequence to start is located.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	2047	default	0

page discussion:

Pages in memory or I/O space are allocated as follows: Page 0 is stack space in operand SRAM, pages 1-7 are in nominal operand SRAM, pages 8-11 are in communications operand SRAM, pages 12-15 are in EEPROM, page 128 is I/O space, pages 256-272 are instruction SRAM, and pages 1024-2047 are SDRAM. Each page contains 32768 words addressed from 0x8000 to 0xFFFF and addresses 0x0000 to 0x7FFF are the same as page 0, with the exception of instruction memory and SDRAM, which have 65536 word pages occupying the full address space (0x0000 to 0xFFFF).

addr: Address of sequence to start.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	65535	default	0

Safety Level

SAFE

Telemetry Verification

cips o_seq_state (334), cips o_seq_addr (329), cips o_seq_page (333)

Related Commands

<i>command</i>	<i>description</i>
term cips sequence (117)	Stop the current command sequence in the given sequence engine.

B.35 init cips sram_code**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Command the CIPS SuROM to switch to the SRAM code.

Discussion

This command forces the CIPS SuROM to switch contexts into the SRAM code already loaded into instruction SRAM.

This command is invalid when CIPS FSW is running out of SRAM. If recieved in this mode, CIPS will reject the command as invalid.

Constraints

This command must be immediately preceeded by an 'arm' command and is only valid when CIPS is executing SuROM.

This command should not be sent if there were errors on boot (i.e. the 'boot_error' telemetry item is TRUE).

Command Target

SuROM software

Format

	opcode	subfield(s)	
	0x10	fill	Command size = 16 bits = 2 bytes.
bit	0	7 8 15	

Subfield(s)**fill:** Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

HAZARDOUS

Telemetry Verification

cips mode (271), cips active_csci (147)

Related Telemetry

telemetry	description
-----------	-------------

B.36 issue cips cmd**Packet Target Application Identifier**

cips, 0x160, (352)

Description

CIPS command

Command Target

software

Format

	opcode		subfield(s)	
	0x00		fill	
bit	0	7	8	15

Command size = 16 bits = 2 bytes.

Subfield(s)**fill:** Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification**B.37 issue cips evt log****Packet Target Application Identifier**

cips, 0x160, (352)

Description

Send an event log telemetry packet.

Discussion

This command forces CIPS to immediately issue the log contents. Nominally, log packets are issued only when sufficient data has been stored such that any new log entry would cause the log packet to be greater than the maximum possible length. In this case, the log packet is issued a new packet is started and the new log entry is added to the new packet.

Command Target

software

Format

opcode		subfield(s)	
0x74		fill	
bit	0	7	15

Command size = 16 bits = 2 bytes.

Subfield(s)

fill: Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification**Related Commands**

<i>command</i>	<i>description</i>
set cips em_mask (99)	Set the software event logging mask.

B.38 issue cips exception**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Send an exception telemetry packet.

Command Target

software

Format

opcode		subfield(s)	
0x7f		fill	
bit	0	7	15

Command size = 16 bits = 2 bytes.

Subfield(s)

fill: Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

Related Telemetry Packets

<i>packet</i>	<i>description</i>
cips exception (484)	cips exception packet.

B.39 issue cips hw_hk

Packet Target Application Identifier

cips, 0x160, (352)

Description

Command the CIPS FSW to issue a hardware housekeeping packet.

Command Target

software

Format

	opcode	subfield(s)	
	0x73	fill	
bit	0	7	8 15

Command size = 16 bits = 2 bytes.

Subfield(s)

fill: Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

Related Commands

<i>command</i>	<i>description</i>
set cips hw_hk_rt (101)	Set the cips housekeeping telemetry packet period.

Related Telemetry Packets

<i>packet</i>	<i>description</i>
cips hw_hk (492)	cips housekeeping packet.

B.40 issue cips pm

Packet Target Application Identifier

cips, 0x160, (352)

Description

Send a post-mortem telemetry packet.

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

opcode		subfield(s)	
0x7e		fill	
bit	0	7	8 15

Command size = 16 bits = 2 bytes.

Subfield(s)

fill: Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

Related Telemetry Packets

<i>packet</i>	<i>description</i>
cips post_mortem (500)	cips post-mortem packet.

B.41 issue cips reserved_1

Packet Target Application Identifier

cips, 0x161, (353)

Description

CIPS reserved command 1

Command Target

software

Format

	opcode	subfield(s)	Command size = 16 bits = 2 bytes.
	0x00	fill	
bit	0	7 8	15

Subfield(s)

fill: Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification**B.42 issue cips reserved_2****Packet Target Application Identifier**

cips, 0x162, (354)

Description

CIPS reserved command 2

Command Target

software

Format

	opcode	subfield(s)	Command size = 16 bits = 2 bytes.
	0x00	fill	
bit	0	7 8	15

Subfield(s)

fill: Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

B.43 issue cips reserved_3

Packet Target Application Identifier

cips, 0x163, (355)

Description

CIPS reserved command 3

Command Target

software

Format

opcode		subfield(s)	
0x00		fill	
bit	0	7	15

Command size = 16 bits = 2 bytes.

Subfield(s)

fill: Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

B.44 issue cips sw_stat

Packet Target Application Identifier

cips, 0x160, (352)

Description

Send a cips software status telemetry packet.

Command Target

software

Format

opcode		subfield(s)	
0x72		fill	
bit	0	7	15

Command size = 16 bits = 2 bytes.

Subfield(s)

fill: Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

Related Commands

<i>command</i>	<i>description</i>
set cips sw_stat_rt (107)	Set period of the cips software status telemetry packet.

Related Telemetry Packets

<i>packet</i>	<i>description</i>
cips sw_status (514)	cips software status.

B.45 noop cips

Packet Target Application Identifier

cips, 0x160, (352)

Description

No operation, or do nothing command.

Discussion

This command performs no useful function other than to verify correct operation of the cips command distributor.

The noop opcode is chosen to be 0, so that undesirable commands may be 'commented out' with the noop opcode.

There is no direct verification of this command.

Indirect verification may be accomplished via the telemetry verification list.

Command Target

software

Format

opcode	subfield(s)
0x00	fill
bit 0 7 8 15	Command size = 16 bits = 2 bytes.

Subfield(s)

fill: Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

cips cmd_succ (175), cips cmd_succ_ct (175), cips cmd_fail_ct (173), cips cmd_fail (172)

B.46 open cips dust_cover**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Deploy the dust cover protecting the given camera.

Constraints

This command must be immediately preceeded by an 'arm' command and is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

opcode		subfield(s)	
0x80	camera	channel	
bit 0	7	8 15	16 31

Command size = 32 bits = 4 bytes.

Subfield(s)

camera: Camera whose dsut cover will be deployed.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	8	default	1
					PX	1
					PY	2
					MX	4
					MY	8

channel: Power line to use for deployment.

bits	type	eng units	min (dn)	max (dn)	state	value
16	discrete state	any	0	1	default	0
					PRIMARY	0
					SECONDARY	1

Safety Level

HAZARDOUS

Telemetry Verification

cips px_dc_status_pri (397), cips my_dc_status_pri (319), cips my_dc_status_sec (319), cips py_dc_status_pri (417), cips mx_dc_status_pri (299), cips py_dc_status_sec (418), cips mx_dc_status_sec (300), cips px_dc_status_sec (398)

B.47 open cips shutter

Packet Target Application Identifier

cips, 0x160, (352)

Description

Open the electronic shutter on the given camera(s).

Constraints

Since the shutter is really just a high voltage photocathode, the HVPS for the camera(s) whose shutters are to be opened must be powered on. Otherwise, this command will be rejected.

The 'disable cips shutter' command must have been issued prior to this command in order for the command to work, because the 'cips px_shutr_disable', 'cips my_shutr_disable', 'cips mx_shutr_disable', and/or 'cips py_shutr_disable' must be 'TRUE' for this command to be accepted (only those cameras whose shutters are to be manually closed are checked to ensure that autonomous shutter control has been disabled).

This command is only valid when CIPS is executing SRAM Code.

Command Target

software

Format

opcode	subfield(s)	
	0x36	cameras
bit	0 7 8 15	Command size = 16 bits = 2 bytes.

Subfield(s)

cameras: Camera(s) whose shutter(s) will be opened.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	1
					PX	1
					PY	2
					MX	4
					X	5
					MY	8
					Y	10
					ALL	15

Safety Level

SAFE

Telemetry Verification

Related Commands

command	description
close cips shutter (44)	Close the electronic shutter on the given camera(s).
disable cips shutter (54)	Disable the autonomous opening and closing of the given shutter(s).
enable cips shutter (68)	Enable the autonomous opening and closing of the given shutter(s).

B.48 pass cips cde

Packet Target Application Identifier

cips, 0x160, (352)

Description

Pass a low-level command ITF to the CDE instrument.

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

opcode		subfield																		
0x52		fill	sync_pattern	msg_type	checksum	length	op_code	cmd_word1	cmd_word2	cmd_word3										
bit	0	7	8	15	16	39	40	47	48	55	56	71	72	87	88	95	96	103	104	111

Command size = 176 bits = 22 bytes.

Subfield(s)

fill: Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

sync_pattern: Sync pattern.

bits	type	eng units	min (dn)	max (dn)	state	value
24	unsigned integer	any	16710192	16710192	default	16710192

msg_type: Message type (command).

bits	type	eng units	min (dn)	max (dn)	state	value
8	unsigned integer	any	2	2	default	2

checksum: Checksum.

bits	type	eng units	min (dn)	max (dn)	state	value
8	unsigned integer	any	0	255	default	0

length: Message length.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	65535	default	0

op_code: Command opcode.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	65535	default	0

cmd_word1: Command word 1.

bits	type	eng units	min (dn)	max (dn)	state	value
8	unsigned integer	any	0	255	default	0

cmd_word2: Command word 2.

bits	type	eng units	min (dn)	max (dn)	state	value
8	unsigned integer	any	0	255	default	0

cmd_word3: Command word 3.

bits	type	eng units	min (dn)	max (dn)	state	value
8	unsigned integer	any	0	255	default	0

cmd_word4: Command word 4.

bits	type	eng units	min (dn)	max (dn)	state	value
8	unsigned integer	any	0	255	default	0

cmd_word5: Command word 5.

bits	type	eng units	min (dn)	max (dn)	state	value
8	unsigned integer	any	0	255	default	0

cmd_word6: Command word 6.

bits	type	eng units	min (dn)	max (dn)	state	value
8	unsigned integer	any	0	255	default	0

cmd_word7: Command word 7.

bits	type	eng units	min (dn)	max (dn)	state	value
8	unsigned integer	any	0	255	default	0

cmd_word8: Command word 8.

bits	type	eng units	min (dn)	max (dn)	state	value
8	unsigned integer	any	0	255	default	0

cmd_word9: Command word 9.

bits	type	eng units	min (dn)	max (dn)	state	value
8	unsigned integer	any	0	255	default	0

cmd_word10: Command word 10.

bits	type	eng units	min (dn)	max (dn)	state	value
8	unsigned integer	any	0	255	default	0

fill 2: Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

cips cmd_succ_ct (175)

B.49 reset cips arm

Packet Target Application Identifier

cips, 0x160, (352)

Description

Disarm CIPS software hazardous command acceptance.

Discussion

This command explicitly clears the CIPS software command arm condition. Although any command that immediately follows an 'arm cips' command would clear the arming condition, this does not apply to command sequences. The CIPS SAFING sequence needs to clear the software arm to prevent HVPS turn on while in SAFE mode.

Command Target

software

Format

opcode		subfield(s)	
0x0f		fill	
bit	0	7	15

Command size = 16 bits = 2 bytes.

Subfield(s)

fill: Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

cips armed (148)

B.50 reset cips cpa**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Reset the CIPS Processor Assembly.

Discussion

CAUTION: This command will reset the CIPS instrument software and hardware!!

Constraints

This command must be preceded by an 'arm' command.

Command Target

software

Format

opcode		subfield(s)	
0x0b		fill	
bit	0	7	15

Command size = 16 bits = 2 bytes.

Subfield(s)**fill:** Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

HAZARDOUS

Telemetry Verification

cips mode (271)

B.51 reset cips sw_stat**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Reset values of counters and error indicators in the software status packet.

Command Target

software

Format

opcode		subfield(s)	
0x08		fill	
bit	0	7	15

Command size = 16 bits = 2 bytes.

Subfield(s)**fill:** Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

cips cmd_succ.ct (175)

B.52 safe cips**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Place the cips instrument in a safe state.

Discussion

Any current cips instrument activity is halted, and the instrument is configured to a safe state via the default instrument shut down sequence. The cips instrument may be left in the safe configuration indefinitely.

Constraints

None. This command can be received at any time.

Command Target

software

Format

opcode		subfield(s)	
0xfd		fill	
bit	0	7	15

Command size = 16 bits = 2 bytes.

Subfield(s)

fill: Pad for even command length.

MUST be here to match OSC's powerdown message.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

cips mode (271)

Related Commands

<i>command</i>	<i>description</i>
wake cips (126)	Remove cips safing constraints, allowing transition out of safe hold.

B.53 select cips cde_chnl**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Select the channel CIPS uses to communicate with CDE.

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

opcode		subfield(s)	
0x53		channel	
bit	0	7	15

Command size = 16 bits = 2 bytes.

Subfield(s)

channel: Channel.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	0	1	default	0
					A	0
					B	1

Safety Level

SAFE

Telemetry Verification

cips_uart_ab_sel (468)

B.54 select cips_gse_ccd**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Selects the CCD whose data output is streamed to the GSE port.

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

opcode		subfield(s)	
0x37		cameras	
bit	0	7	15

Command size = 16 bits = 2 bytes.

Subfield(s)

cameras: Camera(s) to configure.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	1
					PX	1
					PY	2
					MX	4
					MY	8

Safety Level

SAFE

Telemetry Verification

cips test_cam_select (466)

B.55 select cips img_mode**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Select the type of imaging to be performed with this camera.

Constraints

This command is only allowed when CIPS is executing SRAM Code. Also, the cameras being commanded must be powered on. Otherwise, this command will be rejected.

Command Target

software

Format

bit	opcode		subfield(s)			
	0	7	8	15	16	31
	0x29		cameras		mode	

Command size = 32 bits = 4 bytes.

Subfield(s)**cameras:** Camera(s) to configure.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	1
					PX	1
					PY	2
					MX	4
					X	5
					MY	8
					Y	10
					ALL	15

mode: Imaging mode.

bits	type	eng units	min (dn)	max (dn)	state	value
16	discrete state	any	0	1	default	0
					TDI	0
					SNAPSHOT	1

Safety Level

SAFE

Telemetry Verification

cips py_cam_mode (413), cips mx_cam_mode (295), cips px_cam_mode (393), cips my_cam_mode (315)

B.56 select cips tdi_dir

Packet Target Application Identifier

cips, 0x160, (352)

Description

Select the direction in which Time Delayed Integration is done accross the CCD.

Constraints

This command is only allowed when CIPS is executing SRAM Code. Also, the cameras being commanded must be powered on. Otherwise, this command will be rejected.

Command Target

software

Format

opcode		subfield(s)	
0x2a	cameras	dir	
bit 0	7	8	15 16 31

Command size = 32 bits = 4 bytes.

Subfield(s)

cameras: Camera(s) to configure.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	1
					PX	1
					PY	2
					MX	4
					X	5
					MY	8
					Y	10
					ALL	15

dir: TDI direction (as seen from...).

bits	type	eng units	min (dn)	max (dn)	state	value
16	discrete state	any	0	3	default	0
					TO_A	0
					TO_B	1

Safety Level

SAFE

Telemetry Verification

cips cmd_succ.ct (175)

B.57 set cips cde_itf_rt

Packet Target Application Identifier

cips, 0x160, (352)

Description

Set the amount of time between successive CDE ITF data packets issued by CIPS.

Discussion

Despite the command name, this command actually sets the period between issuances of the CDE ITF packet, not the frequency of packet production.

In reality, this packet will be downlinked every N packets CIPS receives from CDE, where N is the value sent in this command. Since CIPS should be receiving data from CDE at 1 Hz, N translates directly to packet period.

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

opcode		subfield(s)	
0x7d	fill	to	
bit 0	7	8	15 16 31

Command size = 32 bits = 4 bytes.

Subfield(s)

fill: Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

to: Packet rate, LSB = 1 sec, 0 = off.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	sec	0	65535	default	0

Safety Level

SAFE

Telemetry Verification

B.58 set cips clear_ct

Packet Target Application Identifier

cips, 0x160, (352)

Description

Set the number of times the specified CCDs will be cleared before data is taken.

Constraints

This command is only allowed when CIPS is executing SRAM Code. Also, the cameras being commanded must be powered on. Otherwise, this command will be rejected.

Command Target

software

Format

bit	opcode		subfield(s)			
	0	7	8	15	16	31
	0x26		cameras	count		

Command size = 32 bits = 4 bytes.

Subfield(s)

cameras: Camera(s) to configure.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	1
					PX	1
					PY	2
					MX	4
					X	5
					MY	8
					Y	10
					ALL	15

count: Fast clear count.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	7	default	0

Safety Level

SAFE

Telemetry Verification

cips mx_cam_clear_ct (293), cips px_cam_clear_ct (391), cips py_cam_clear_ct (411), cips my_cam_clear_ct (313)

B.59 set cips cmn_a_offset**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Set the offsets for the CCD's 'Common Area' for all cameras.

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

camera fpgas

Format

bit	opcode		subfield(s)				
	0x3f	fill	row	column			
	0	7	8	15	16	31	32 47

Command size = 48 bits = 6 bytes.

Subfield(s)

fill: Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

row: Number of rows to offset the origin of the the CCDs' common area.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	2079	default	0

column: Number of rows to offset the origin of the the CCDs' common area.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	2071	default	0

Safety Level

SAFE

Telemetry Verification

cips cmd_succ_ct (175)

Related Commands

command	description
set cips cmn_a_size (98)	Set the size of the CCD's 'Common Area' for all cameras.

B.60 set cips cmn_a_size**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Set the size of the CCD's 'Common Area' for all cameras.

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

camera fpgas

Format

bit	opcode		subfield(s)				
	0x3e	fill	rows	columns			
	0	7	8	15	16	31	32 47

Command size = 48 bits = 6 bytes.

Subfield(s)

fill: Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

rows: Number of rows in the CCDs' common area.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	2080	default	0

columns: Number of columns in the CCDs' common area.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	2072	default	0

Safety Level

SAFE

Telemetry Verification

cips cmd_succ_ct (175)

Related Commands

command	description
set cips cmn_a_offset (97)	Set the offsets for the CCD's 'Common Area' for all cameras.

B.61 set cips em_mask

Packet Target Application Identifier

cips, 0x160, (352)

Description

Set the software event logging mask.

Discussion

This effectively controls the bandwidth taken up by output of the cips event log telemetry packet. It also affects how much detailed data FSW engineers and mission operations personnel have to use when troubleshooting.

Command Target

software

Format

opcode		subfield(s)			
0x77		fill	newmask	passmask	
bit	0 7	8 15	16 47	48 79	Command size = 80 bits = 10 bytes.

Subfield(s)

fill: Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

newmask: New event key mask.

bits	type	eng units	min (dn)	max (dn)	state	value
32	unsigned integer	any	0	4294967295	default	0

passmask: Mask for the bits specified in the newmask subfield.

bits	type	eng units	min (dn)	max (dn)	state	value
32	unsigned integer	any	0	4294967295	default	0

Safety Level

SAFE

Telemetry Verification

cips em_warn_msg (233), cips em_seq_err (229), cips em_debug (224), cips em_tlm_err (232), cips em_info (226), cips em_pwr (227), cips em_u_test (233), cips em_cmd_err (223), cips em_mode_chng (227), cips em_tlm_tx (232), cips em_time_rx (231), cips em_acs_stat_rx (221), cips em_cde_cmd_tx (222), cips em_seq_mode_chng (230), cips em_cmd_rx (223), cips em_err (225), cips em_cde_tlm_rx (222)

Related Commands

<i>command</i>	<i>description</i>
issue cips evt_log (79)	Send an event log telemetry packet.

Related Telemetry Packets

<i>packet</i>	<i>description</i>
cips evt_log (482)	cips event log packet.

B.62 set cips frame_sync

Packet Target Application Identifier

cips, 0x160, (352)

Description

Set the delay time between the 'GO' to take an image and CCD readout.

Discussion

This command is necessary because the integration timing for cameras in TDI mode is different than those in snapshot mode. It is preferable for all cameras to end their CCD readout time synchronously, so a delay is necessary for the cameras in snapshot mode as they have a shorter integration period.

Constraints

This command is only allowed when CIPS is executing SRAM Code. Also, the cameras being commanded must be powered on. Otherwise, this command will be rejected.

Command Target

software

Format

bit	opcode		subfield(s)				
	0	7	8	15	16	31	
	0x38		cameras	delay			

Command size = 32 bits = 4 bytes.

Subfield(s)**cameras:** Camera(s) to configure.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	1
					PX	1
					PY	2
					MX	4
					X	5
					MY	8
					Y	10
					ALL	15

delay:	bits	type	eng units	min (dn)	max (dn)	state	value
	16	unsigned integer	ms	0	65535	default	0

Safety Level

SAFE

Telemetry Verification

cips mx_cam_f_sync (293), cips my_cam_f_sync (313), cips py_cam_f_sync (411), cips px_cam_f_sync (391)

B.63 set cips hw_hk_rt**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Set the cips housekeeping telemetry packet period.

Discussion

Despite the command name, this command actually sets the period between issuances of the cips housekeeping telemetry packet, not the frequency of packet production.

Command Target

software

Format

opcode		subfield(s)	
0x71		fill	to
bit	0	7	8 15 16 31

Command size = 32 bits = 4 bytes.

Subfield(s)**fill:** Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

to: Packet rate, LSB = 1 sec, 0 = off.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	sec	0	65535	default	0

Safety Level

SAFE

Telemetry Verification

cips hw_hk_rt (252)

Related Commands

command	description
issue cips hw_hk (81)	Command the CIPS FSW to issue a hardware housekeeping packet.

Related Telemetry Packets

packet	description
cips hw_hk (492)	cips housekeeping packet.

B.64 set cips img_delay

Packet Target Application Identifier

cips, 0x160, (352)

Description

Set the time interval between CIPS images during the next observation.

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

bit	opcode		subfield(s)			
	0	7	8	15	16	31
	0x2d		fill		to	

Command size = 32 bits = 4 bytes.

Subfield(s)

fill: Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

to: Time between images.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	sec	1	65535	default	43

Safety Level

SAFE

Telemetry Verification

cips img_delay_tm (259)

B.65 set cips img_offset**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Set the raw CCD image origin offset from the top and side of the CCD.

Constraints

This command is only allowed when CIPS is executing SRAM Code. Also, the cameras being commanded must be powered on. Otherwise, this command will be rejected.

Command Target

software

Format

opcode		subfield(s)						
0x27		cameras		row		column		
bit	0	7	8	15	16	31	32	47

Command size = 48 bits = 6 bytes.

Subfield(s)**cameras:** Camera(s) to configure.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	1
					PX	1
					PY	2
					MX	4
					X	5
					MY	8
					Y	10
					ALL	15

row: Pixel offset.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	2047	default	0

column: Pixel offset.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	2047	default	0

Safety Level

SAFE

Telemetry Verification

cips cmd_succ_ct (175), cips img_offset_row (261), cips img_offset_col (260)

B.66 set cips_img_size

Packet Target Application Identifier

cips, 0x160, (352)

Description

Set the raw CCD image size for all CIPS cameras.

Discussion

Image sizes must be identical for all CIPS cameras, thus this command has no 'cameras' subfield.

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

bit	opcode		subfield(s)			
	0x28	fill	columns	rows		
0	7	8	15	16	31	32 47

Command size = 48 bits = 6 bytes.

Subfield(s)

fill: Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

columns: Image width, in pixels.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	1	2048	default	1539

rows: Image height, in pixels.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	1	2048	default	1539

Safety Level

SAFE

Telemetry Verification

cips_img_cols_ccd (258), cips_img_rows_ccd (262)

Related Telemetry Packets

packet	description
cips_sci_px_cam (508)	CIPS +X camera science data packet.
cips_sci_py_cam (511)	CIPS +Y camera science data packet.
cips_sci_mx_cam (502)	CIPS -X camera science data packet.
cips_sci_my_cam (505)	CIPS -Y camera science data packet.

B.67 set cips integration

Packet Target Application Identifier

cips, 0x160, (352)

Description

Set the integration time for the given CIPS camera(s).

Constraints

This command is only allowed when CIPS is executing SRAM Code. Also, the cameras being commanded must be powered on. Otherwise, this command will be rejected.

Command Target

software

Format

opcode		subfield(s)	
0x2b		cameras	time
bit	0 7 8 15 16 31	Command size = 32 bits = 4 bytes.	

Subfield(s)

cameras: Camera(s) to configure.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	1
					PX	1
					PY	2
					MX	4
					X	5
					MY	8
					Y	10
					ALL	15

time: Integration time.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	ms	0	65535	default	0

Safety Level

SAFE

Telemetry Verification

cips my_cam_ss.time (317), cips px_cam_ss.time (395), cips mx_cam_ss.time (297), cips py_cam_ss.time (415)

B.68 set cips mcp_volts

Packet Target Application Identifier

cips, 0x160, (352)

Description

Set the microchannel plate voltage level for the given CIPS camera(s).

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

hk fpga

Format

opcode		subfield(s)			
0x40		cameras	voltage		
bit	0	7	8	15	16 31

Command size = 32 bits = 4 bytes.

Subfield(s)

cameras: Camera(s) to configure.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	1
					PX	1
					PY	2
					MX	4
					X	5
					MY	8
					Y	10
					ALL	15

voltage: MCP voltage level.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	V	32	255	default	32

voltage conversion coefficients:

c0	c1	c2	c3
0	0.255	0	0

Safety Level

SAFE

Telemetry Verification

cips my_cam_mcpv_mon (314), cips px_cam_mcpv_mon (392), cips py_cam_mcpv_mon (412), cips mx_cam_mcpv_mon (294)

B.69 set cips obs_id**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Set a unique identifier for the ensuing science images.

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

opcode		subfield(s)				
0x3d		fill	id			
bit	0	7	8	15	16	31

Command size = 32 bits = 4 bytes.

Subfield(s)

fill: Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

id: Observation ID for the next images acquired.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	sec	0	65535	default	0

Safety Level

SAFE

Telemetry Verification

cips obs_id (337)

B.70 set cips sw_stat_rt**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Set period of the cips software status telemetry packet.

Discussion

Despite the command name, this command actually sets the period between issuances of the cips software status telemetry packet, not the frequency of packet production.

Setting the period to 0 turns off production of the cips software status telemetry packet.

Command Target

software

Format

opcode		subfield(s)				
0x70		fill		to		
bit	0	7	8	15	16	31

Command size = 32 bits = 4 bytes.

Subfield(s)

fill: Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

to: Packet rate, LSB = 1 SEC, 0 = off.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	sec	0	65535	default	0

Safety Level

SAFE

Telemetry Verification**Related Commands**

command	description
issue cips sw.stat (84)	Send a cips software status telemetry packet.

Related Telemetry Packets

packet	description
cips sw.status (514)	cips software status.

B.71 set cips tdi_row_ct**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Set the number of pixel rows over which to perform Time Delayed Integration.

Constraints

This command is only allowed when CIPS is executing SRAM Code. Also, the cameras being commanded must be powered on. Otherwise, this command will be rejected.

Command Target

software

Format

opcode		subfield(s)				Command size = 32 bits = 4 bytes.
0x25		cameras		count		
bit	0	7	8	15	16	31

Subfield(s)

cameras: Camera(s) to configure.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	1
					PX	1
					PY	2
					MX	4
					X	5
					MY	8
					Y	10
					ALL	15

count: Number of pixel rows.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

cips px_cam_tdi_rows (395), cips py_cam_tdi_rows (415), cips my_cam_tdi_rows (317), cips mx_cam_tdi_rows (297)

B.72 set cips tdi_tm

Packet Target Application Identifier

cips, 0x160, (352)

Description

Set the amount of time Time Delayed Integration will be performed per pixel row.

Constraints

This command is only allowed when CIPS is executing SRAM Code. Also, the cameras being commanded must be powered on. Otherwise, this command will be rejected.

Command Target

software

Format

opcode		subfield(s)				
0x24		cameras		time		
bit	0	7	8	15	16	31

Command size = 32 bits = 4 bytes.

Subfield(s)

cameras: Camera(s) to configure.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	1
					PX	1
					PY	2
					MX	4
					X	5
					MY	8
					Y	10
					ALL	15

time: TDI time (per pixel row).

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	ms	0	102	default	0

Safety Level

SAFE

Telemetry Verification

cips px_cam_tdi_time (396), **cips mx_cam_tdi_time** (298), **cips my_cam_tdi_time** (317), **cips py_cam_tdi_time** (416)

B.73 set cips to mask

Packet Target Application Identifier

cips, 0x160, (352)

Description

Set the CIPS telemetry output enable mask.

Command Target

software

Format

opcode		subfield(s)						
0x7a		fill	newmask		passmask			
bit	0	7	8	15	16	47	48	79

Command size = 80 bits = 10 bytes.

Subfield(s)

fill: Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

newmask: Telemetry output mask.

bits	type	eng units	min (dn)	max (dn)	state	value
32	unsigned integer	any	0	4294967295	default	0

passmask: Mask for the bits in the newmask subfield.

bits	type	eng units	min (dn)	max (dn)	state	value
32	unsigned integer	any	0	4294967295	default	0

Safety Level

SAFE

Telemetry Verification

cips reserved_4_to (430), cips reserved_1_to (428), cips cde_mem_dump_to (158), cips cde_flash_to (155), cips cmd_echo_to (171), cips cde_hist_log_to (156), cips err_to (234), cips pm_to (383), cips reserved_3_to (429), cips mem_dump_to (271), cips cde_itf_to (157), cips evt_log_to (238), cips reserved_2_to (429), cips cde_sci_to (158), cips cde_sys_data_to (159), cips cde_cal_to (154), cips hw_hk_to (253), cips exception_to (239), cips sw_stat_to (464), cips sci_mx_cam_to (432), cips sci_my_cam_to (432), cips sci_py_cam_to (434), cips sci_px_cam_to (433)

Related Commands

<i>command</i>	<i>description</i>
disable cips tlm_out (56)	Disable downlink of the given CIPS telemetry packet.
enable cips tlm_out (70)	Enable downlink of the given CIPS telemetry packet.

B.74 stuff cips cde_memory

Packet Target Application Identifier

cips, 0x160, (352)

Description

Load data to CDE memory.

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

cde

Format

		opcode		subfield(s)									
		0x54		type		addr		reserved_1		data		reserved_2	
bit		0	7	8	15	16	47	48	55	56	1095	1096	1103

Command size = 1104 bits = 138 bytes.

Subfield(s)

type: Type of memory to load.

bits	type	eng units	min (dn)	max (dn)	state	value
8	unsigned integer	any	0	255	default	0

addr: Memory load starting address.

bits	type	eng units	min (dn)	max (dn)	state	value
32	unsigned integer	any	0	4294967295	default	0

reserved_1: Pad for alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

data: Memory load data.

bits	type	eng units	min (dn)	max (dn)	state	value
1040	fill	any	0	0	default	0

reserved_2: Pad for alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

cips cmd_succ_ct (175)

B.75 stuff cips memory

Packet Target Application Identifier

cips, 0x160, (352)

Description

Load memory.

Discussion

Load data to memory or I/O space.

Constraints

The length of the memory to write must fit within the page boundary—the address to write to and the length of memory to write must not add up to a value greater than 65536.

Writes to EEPROM must be preceeded by the 'enable ee_write' command.

Mistakes in memory write commands can lead to loss of science data or processor reset. Be very sure of memory write command parameters before sending any memory write command.

Command Target

software

Format

opcode		subfield(s)										
0x06		fill	page		addr		length		data			
bit	0	7	8	15	16	31	32	47	48	63	64	1983

Command size = 1984 bits = 248 bytes.

Subfield(s)

fill: Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

page: Page in memory or I/O space to access

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	2047	default	6

page discussion:

Pages in memory or I/O space are allocated as follows: Page 0 is stack space in operand SRAM, pages 1-7 are in nominal operand SRAM, pages 8-11 are in communications operand SRAM, pages 12-15 are in EEPROM, page 128 is I/O space, pages 256-272 are instruction SRAM, and pages 1024-2047 are SDRAM. Each page contains 32768 words addressed from 0x8000 to 0xFFFF and addresses 0x0000 to 0x7FFF are the same as page 0, with the exception of instruction memory and SDRAM, which have 65536 word pages occupying the full address space (0x0000 to 0xFFFF).

addr: Memory load starting address.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	65535	default	32768

length: Memory load length.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	1	120	default	4

data: Memory load data.

bits	type	eng units	min (dn)	max (dn)	state	value
1920	fill	any	0	0	default	0

Safety Level

SAFE

Telemetry Verification

cips cmd_succ_ct (175)

Related Commands

<i>command</i>	<i>description</i>
disable cips ee_write (50)	Disable writes to CIPS EEPROM.
dump cips memory (57)	Dump CIPS memory.
enable cips ee_write (62)	Enable writes to CIPS EEPROM.
stuff cips memory_wd (113)	Load a single 16-bit word of data to memory or I/O space.
copy cips memory (45)	Copy data from one location to another.

B.76 stuff cips memory_wd

Packet Target Application Identifier

cips, 0x160, (352)

Description

Load a single 16-bit word of data to memory or I/O space.

Constraints

Writes to EEPROM must be preceeded by command to enable EEPROM writes.

Memory write errors can lead to loss of science data or processor reset. Be very sure of memory write command parameters.

Command Target

software

Format

bit	opcode		subfield(s)							
	0x03	fill	page	addr	word	Command size = 64 bits = 8 bytes.				
	0	7	8	15	16	31	32	47	48	63

Subfield(s)

fill: Unused field required for word alignment.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

page: Page in memory or I/O space to access

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	2047	default	0

page discussion:

Pages in memory or I/O space are allocated as follows: Page 0 is stack space in operand SRAM, pages 1-7 are in nominal operand SRAM, pages 8-11 are in communications operand SRAM, pages 12-15 are in EEPROM, page 128 is I/O space, pages 256-272 are instruction SRAM, and pages 1024-2047 are SDRAM. Each page contains 32768 words addressed from 0x8000 to 0xFFFF and addresses 0x0000 to 0x7FFF are the same as page 0, with the exception of instruction memory and SDRAM, which have 65536 word pages occupying the full address space (0x0000 to 0xFFFF).

addr: Memory load address.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	65535	default	0

word: Memory load data word.

bits	type	eng units	min (dn)	max (dn)	state	value
16	unsigned integer	any	0	65535	default	0

Safety Level

SAFE

Telemetry Verification

cips cmd_succ_ct (175)

Related Commands

<i>command</i>	<i>description</i>
disable cips ee.write (50)	Disable writes to CIPS EEPROM.
dump cips memory (57)	Dump CIPS memory.
enable cips ee.write (62)	Enable writes to CIPS EEPROM.
stuff cips memory (112)	Load memory.
copy cips memory (45)	Copy data from one location to another.

B.77 suspend cips abs_tm**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Wait until an absolute time.

Discussion

cips command interpretation pauses until the indicated absolute spacecraft time.

Current cips configuration and operations from previous commands continues.

Constraints

The spacecraft time at beginning of command interpretation must be less than the spacecraft time contained in the command, otherwise the command completes immediately.

This command will be rejected unless it comes from one of the internal CIPS command sequence engine.

Command Target

software

Format

opcode	subfield(s)	
	fill	to
0x61		

Command size = 48 bits = 6 bytes.

bit 0 7 8 15 16 47

Subfield(s)**fill:** Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

to: Packet rate,LSB = 1 s.

bits	type	eng units	min (dn)	max (dn)	state	value
32	unsigned integer	sec	0	4294967295	default	0

Safety Level

SAFE

Telemetry Verification

cips a_seq_wait_time (144), **cips a_seq_state** (142), **cips o_seq_state** (334), **cips o_seq_wait_time** (336), **cips a_seq_wait_type** (145), **cips o_seq_wait_type** (337)

Related Commands

<i>command</i>	<i>description</i>
suspend cips rel_tm (116)	Wait for a relative time delay.

B.78 suspend cips rel_tm

Packet Target Application Identifier

cips, 0x160, (352)

Description

Wait for a relative time delay.

Discussion

cips command interpretation pauses for a relative time delay. Current cips configuration and operations from previous commands continue.

Constraints

This command will be rejected unless it comes from one of the internal CIPS command sequence engines.

Command Target

software

Format

bit	opcode		subfield(s)			
	0	7	8	15	16	47
	0x60		fill		to	

Command size = 48 bits = 6 bytes.

Subfield(s)

fill: Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

to: Suspend time, LSB = 1 s.

bits	type	eng units	min (dn)	max (dn)	state	value
32	unsigned integer	sec	1	65535	default	1

Safety Level

SAFE

Telemetry Verification

cips a_seq_wait_time (144), **cips a_seq_state** (142), **cips o_seq_state** (334), **cips o_seq_wait_time** (336), **cips a_seq_wait_type** (145), **cips o_seq_wait_type** (337)

Related Commands

<i>command</i>	<i>description</i>
suspend cips abs_tm (115)	Wait until an absolute time.

B.79 term cips sequence

Packet Target Application Identifier

cips, 0x160, (352)

Description

Stop the current command sequence in the given sequence engine.

Constraints

This command will be rejected if the commanded sequence engine is in the IDLE state.

Command Target

software

Format

opcode		subfield(s)	
0x62		engine	
bit	0	7	8 15

Command size = 16 bits = 2 bytes.

Subfield(s)

engine: Sequence engine to command.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	0	1	default	0
					OP	0
					AUTO	1

Safety Level

SAFE

Telemetry Verification

cips o_seq_stop_code (335), cips a_seq_state (142), cips o_seq_state (334), cips a_seq_stop_code (143)

Related Commands

command	description
init cips sequence (77)	Start a CIPS command sequence.

B.80 turn off cips camera

Packet Target Application Identifier

cips, 0x160, (352)

Description

Disable power to the selected CIPS camera(s).

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

	opcode	subfield(s)	
	0x85	cameras	Command size = 16 bits = 2 bytes.
bit	0	7	8 15

Subfield(s)**cameras:** Camera(s) which will be turned off.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	1
					PX	1
					PY	2
					MX	4
					X	5
					MY	8
					Y	10
					ALL	15

Safety Level

SAFE

Telemetry Verification

cips py_cam_pwr (414), cips my_cam_pwr (316), cips px_cam_pwr (394), cips mx_cam_pwr (296)

B.81 turn off cips ips**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Turn off power to the given CIPS intensifier power supply.

Constraints

This command is only allowed when CIPS is executing SRAM Code. Further, the command will be rejected unless any of the manually controlled high voltage outputs (consisting of the MCP, phosphor, and photocathode [shutter]) are enabled (for the photocathode, enabled means that the shutter is open) for the given cameras.

Command Target

software

Format

	opcode	subfield(s)	
	0x45	cameras	Command size = 16 bits = 2 bytes.
bit	0	7	8 15

Subfield(s)

cameras: camera(s) whose intensifier power supply to turn off.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	15
					PX	1
					PY	2
					MX	4
					X	5
					MY	8
					Y	10
					ALL	15

Safety Level

SAFE

Telemetry Verification

cips my_ips_pwr (322), cips mx_ips_pwr (303), cips px_ips_pwr (401), cips py_ips_pwr (421)

B.82 turn off cips mcp

Packet Target Application Identifier

cips, 0x160, (352)

Description

Turns off the MCP high voltage power supply for the given CIPS camera(s).

Discussion

MCP = Microchannel Plate

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

opcode	subfield(s)	Command size = 16 bits = 2 bytes.
0x41	cameras	
bit 0 7 8 15		

Subfield(s)

cameras: Camera(s) to configure.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	1
					PX	1
					PY	2
					MX	4
					X	5
					MY	8
					Y	10
					ALL	15

Safety Level

SAFE

Telemetry Verification

cips my_cam_mcp_pwr (313), cips py_cam_mcp_pwr (411), cips mx_cam_mcp_pwr (293), cips px_cam_mcp_pwr (392)

B.83 turn off cips phosphr

Packet Target Application Identifier

cips, 0x160, (352)

Description

Turns off the phosphor high voltage power supply for the given CIPS camera(s).

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

opcode	subfield(s)
0x43	cameras

Command size = 16 bits = 2 bytes.

bit 0 7 8 15

Subfield(s)

cameras: Camera(s) to configure.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	1
					PX	1
					PY	2
					MX	4
					X	5
					MY	8
					Y	10
					ALL	15

Safety Level

SAFE

Telemetry Verification

cips px_cam_p_pwr (393), cips mx_cam_p_pwr (295), cips my_cam_p_pwr (315), cips py_cam_p_pwr (413)

B.84 turn off cips sdram**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Disable power to the specified bank(s) of SDRAM memory devices.

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

hk fpga

Format

opcode		subfield(s)	
0x3b		bank	
bit	0	7	15

Command size = 16 bits = 2 bytes.

Subfield(s)**bank:** Specifies which bank(s) of memory chips to turn off.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	0	2	default	0
					A	0
					B	1
					BOTH	2

Safety Level

SAFE

Telemetry Verification

cips sdram.a_pwr (435), cips sdram.b_pwr (436)

Related Commands

<i>command</i>	<i>description</i>
turn on cips sdram (125)	Enable power to the specified bank(s) of SDRAM memory devices.

B.85 turn on cips camera

Packet Target Application Identifier

cips, 0x160, (352)

Description

Enable power to the selected CIPS camera(s).

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

software

Format

	opcode	subfield(s)	
	0x84	camera	Command size = 16 bits = 2 bytes.
bit	0	7	8 15

Subfield(s)

camera: Camera(s) which will be turned on.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	8	default	1
					PX	1
					PY	2
					MX	4
					MY	8

Safety Level

SAFE

Telemetry Verification

cips py_cam_pwr (414), cips my_cam_pwr (316), cips px_cam_pwr (394), cips mx_cam_pwr (296)

B.86 turn on cips ips

Packet Target Application Identifier

cips, 0x160, (352)

Description

Turn on low voltage power to the given CIPS high voltage power supply.

Constraints

This command is only allowed when CIPS is executing SRAM Code. Further, the command will be rejected unless any of the manually controlled high voltage outputs (consisting of the MCP, phosphor, and photocathode [shutter]) are enabled (for the photocathode, enabled means that the shutter is open) for the given cameras.

Command Target

software

Format

opcode		subfield(s)	
0x46		cameras	
bit	0	7	8 15

Command size = 16 bits = 2 bytes.

Subfield(s)**cameras:** HVPS supply to turn on.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	15
					PX	1
					PY	2
					MX	4
					X	5
					MY	8
					Y	10
					ALL	15

Safety Level

SAFE

Telemetry Verification

cips my_ips_pwr (322), cips mx_ips_pwr (303), cips px_ips_pwr (401), cips py_ips_pwr (421)

B.87 turn on cips mcp**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Turns on the MCP high voltage power supply for the given CIPS camera(s).

Discussion

This command must be immediately preceeded by an 'arm cips' command.

MCP = Microchannel Plate

Constraints

This command must be immediately preceeded by an 'arm' command and is only allowed when CIPS is executing SRAM Code and is not in SAFE mode.

Also, if the BOS level for any of the cameras being commanded is currently at a level higher than that camera's BOS threshold, the command will be rejected.

Finally, the intensifier power supply for the cameras being commanded must be powered on. Otherwise, this command will be rejected.

Command Target

software

Format

	opcode	subfield(s)	
	0x42	cameras	Command size = 16 bits = 2 bytes.
bit	0	7	8
			15

Subfield(s)**cameras:** Camera(s) to configure.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	1
					PX	1
					PY	2
					MX	4
					X	5
					MY	8
					Y	10
					ALL	15

Safety Level

HAZARDOUS

Telemetry Verification

cips my_cam_mcp_pwr (313), cips py_cam_mcp_pwr (411), cips mx_cam_mcp_pwr (293), cips px_cam_mcp_pwr (392)

B.88 turn on cips phosphr**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Turns on the phosphor high voltage power supply for the given CIPS camera(s).

Discussion

This command must be immediately preceeded by an 'arm cips' command.

Constraints

This command must be immediately preceeded by an 'arm' command and is only allowed when CIPS is executing SRAM Code.

Also, since the shutter is really just a high voltage photocathode, the HVPS for the camera(s) whose shutters are to be opened must be powered on. Otherwise, this command will be rejected.

Command Target

software

Format

opcode		subfield(s)	Command size = 16 bits = 2 bytes.
0x44		cameras	
bit	0	7 8 15	

Subfield(s)

cameras: Camera(s) to configure.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	15	default	1
					PX	1
					PY	2
					MX	4
					X	5
					MY	8
					Y	10
					ALL	15

Safety Level

HAZARDOUS

Telemetry Verification

cips px_cam_p_pwr (393), cips mx_cam_p_pwr (295), cips my_cam_p_pwr (315), cips py_cam_p_pwr (413)

B.89 turn on cips sdram**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Enable power to the specified bank(s) of SDRAM memory devices.

Constraints

This command is only allowed when CIPS is executing SRAM Code.

Command Target

hk fpga

Format

opcode		subfield(s)	Command size = 16 bits = 2 bytes.
0x3c		bank	
bit	0	7 8 15	

Subfield(s)

bank: Specifies which bank(s) of memory chips to turn on.

bits	type	eng units	min (dn)	max (dn)	state	value
8	discrete state	any	1	2	default	1
					A	1
					B	2

Safety Level

SAFE

Telemetry Verification

cips sdram_a_pwr (435), cips sdram_b_pwr (436)

Related Commands

<i>command</i>	<i>description</i>
turn off cips sdram (121)	Disable power to the specified bank(s) of SDRAM memory devices.

B.90 wake cips**Packet Target Application Identifier**

cips, 0x160, (352)

Description

Remove cips safing constraints, allowing transition out of safe hold.

Discussion

This command forces CIPS out of SAFE mode.

Constraints

This command is only valid as a real-time command (cannot be executed in a sequence). It will also be rejected if CIPS is not in SAFE mode at the time of command receipt.

Command Target

software

Format

opcode		subfield(s)	
0xfb		fill	
bit	0	7	15

Command size = 16 bits = 2 bytes.

Subfield(s)**fill:** Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	fill	any	0	255	default	0

Safety Level

SAFE

Telemetry Verification

cips cmd_succ_ct (175)

Related Commands

<i>command</i>	<i>description</i>
safe cips (91)	Place the cips instrument in a safe state.

B.91 inform inst acs_state

Packet Target Application Identifier

inst, 0x15e, (350)

Description

Spacecraft ACS status update command.

Discussion

Inform the instruments of the current spacecraft attitude control subsystem (ACS) status.

Constraints

This command does not directly command or reconfigure science instruments. Though each instrument is programmable to respond to different ACS status values under multiple conditions.

Command Target

software

Format

opcode		subfield(s)									
0xfe		filler_1	filler_2	seconds	subseconds	sc_to_sun_u_x	sc_to_sun_u_y	sc_to_sun_u_z	filler_3		
bit	0 7	8 15	16 79	80 111	112 143	144 207	208 271	272 335	336 447		

Command size = 448 bits = 56 bytes.

Subfield(s)

filler_1: Filler data to match OSC command structure.

bits	type	eng units	min (dn)	max (dn)	state	value
8	unsigned integer	any	0	0	default	0

filler_2: Filler data to match OSC command structure.

bits	type	eng units	min (dn)	max (dn)	state	value
64	unsigned integer	any	0	0	default	0

seconds: Seconds field of timestamp.

bits	type	eng units	min (dn)	max (dn)	state	value
32	unsigned integer	any	0	4294967295	default	0

subseconds: Fractional part of timestamp.

bits	type	eng units	min (dn)	max (dn)	state	value
32	unsigned integer	any	0	4294967295	default	0

sc_to_sun_u_x: Spacecraft to sun unit vector, spacecraft body frame, cartesian x coordinate.

bits	type	eng units	min (dn)	max (dn)	state	value
64	float	any	0	0	default	0

sc_to_sun_u_y: Spacecraft to sun unit vector, spacecraft body frame, cartesian y coordinate.

bits	type	eng units	min (dn)	max (dn)	state	value
64	float	any	0	0	default	0

sc_to_sun_u_z: Spacecraft to sun unit vector, spacecraft body frame, cartesian z coordinate.

bits	type	eng units	min (dn)	max (dn)	state	value
64	float	any	0	0	default	0

filler_3: Filler data to match OSC command structure.

bits	type	eng units	min (dn)	max (dn)	state	value
112	unsigned integer	any	0	0	default	0

Safety Level

SAFE

Telemetry Verification

B.92 inform inst sc_clock

Packet Target Application Identifier

inst, 0x15d, (349)

Description

Notify FSW of spacecraft time at next time sync.

Discussion

Notify FSW of the spacecraft clock time at the next sync mode code upon the 1553 bus.

Constraints

By itself, this command will not set the instrument propagated spacecraft clock. To set the clock, this command must be followed by a MIL-STD-1553 SYNC mode code within TBD seconds.

An incorrect value of spacecraft clock can seriously degrade instrument command sequencing and the resulting collection of science data.

Command Target

software

Format

opcode		subfield(s)			
0xfc		reserved	seconds	subseconds	
bit	0 7	8 15	16 47	48	79

Command size = 80 bits = 10 bytes.

bit 0 7 8 15 16 47 48 79

Subfield(s)

reserved: Pad for even command length.

bits	type	eng units	min (dn)	max (dn)	state	value
8	unsigned integer	any	0	255	default	0

seconds: Integral seconds portion of spacecraft clock.

bits	type	eng units	min (dn)	max (dn)	state	value
32	unsigned integer	any	0	4294967295	default	0

subseconds: Integral subseconds portion of spacecraft clock.

bits	type	eng units	min (dn)	max (dn)	state	value
32	unsigned integer	any	0	4294967295	default	0

subseconds discussion:

Each bit represents 2 to the -32 seconds.

Safety Level

SAFE

Telemetry Verification

C

Command Opcode Summary

Command Opcode Summary

cmd ref.	apid ref.	apid (hex)	opcode (hex)	command
1	1	0x15d	0xfc	inform inst sc_clock (128)
2	1	0x15e	0xfe	inform inst acs_state (127)
3	1	0x160	0x00	noop cips (85)
4	2	0x160	0x00	issue cips cmd (79)
5	1	0x163	0x00	issue cips reserved_3 (84)
6	1	0x162	0x00	issue cips reserved_2 (83)
7	1	0x161	0x00	issue cips reserved_1 (82)
8	1	0x160	0x01	arm cips (43)
9	2	0x160	0x02	enable cips exc_pm (64)
10	3	0x160	0x03	stuff cips memory_wd (113)
11	4	0x160	0x04	enable cips ee_write (62)
12	5	0x160	0x05	disable cips ee_write (50)
13	6	0x160	0x06	stuff cips memory (112)
14	7	0x160	0x07	init cips crc (74)
15	8	0x160	0x08	reset cips sw_stat (91)
16	9	0x160	0x09	copy cips memory (45)
17	10	0x160	0x0b	reset cips cpa (90)
18	11	0x160	0x0f	reset cips arm (89)
19	12	0x160	0x10	init cips sram_code (78)
20	13	0x160	0x21	init cips imaging (75)
21	14	0x160	0x22	enable cips sdp (66)
22	15	0x160	0x23	disable cips sdp (53)
23	16	0x160	0x24	set cips tdi_tm (109)
24	17	0x160	0x25	set cips tdi_row_ct (108)
25	18	0x160	0x26	set cips clear_ct (96)
26	19	0x160	0x27	set cips img_offset (103)
27	20	0x160	0x28	set cips img_size (104)
28	21	0x160	0x29	select cips img_mode (94)
29	22	0x160	0x2a	select cips tdi_dir (95)

C. COMMAND OPCODE SUMMARY

Command Opcode Summary, cont'd

cmd ref.	apid ref.	apid (hex)	opcode (hex)	command
30	23	0x160	0x2b	set cips integration (105)
31	24	0x160	0x2c	enable cips sdram_ud (67)
32	25	0x160	0x2d	set cips img_delay (102)
33	26	0x160	0x2e	enable cips comp (61)
34	27	0x160	0x2f	disable cips comp (49)
35	28	0x160	0x30	enable cips sw_bin (69)
36	29	0x160	0x31	disable cips sw_bin (55)
37	30	0x160	0x32	enable cips cam_bin (60)
38	31	0x160	0x33	disable cips cam_bin (47)
39	32	0x160	0x35	close cips shutter (44)
40	33	0x160	0x36	open cips shutter (87)
41	34	0x160	0x37	select cips gse_ccd (93)
42	35	0x160	0x38	set cips frame_sync (100)
43	36	0x160	0x39	disable cips shutter (54)
44	37	0x160	0x3a	enable cips shutter (68)
45	38	0x160	0x3b	turn off cips sdram (121)
46	39	0x160	0x3c	turn on cips sdram (125)
47	40	0x160	0x3d	set cips obs_id (106)
48	41	0x160	0x3e	set cips cmn_a_size (98)
49	42	0x160	0x3f	set cips cmn_a_offset (97)
50	43	0x160	0x40	set cips mcp_volts (105)
51	44	0x160	0x41	turn off cips mcp (119)
52	45	0x160	0x42	turn on cips mcp (123)
53	46	0x160	0x43	turn off cips phosphr (120)
54	47	0x160	0x44	turn on cips phosphr (124)
55	48	0x160	0x45	turn off cips ips (118)
56	49	0x160	0x46	turn on cips ips (122)
57	50	0x160	0x50	enable cips cde_uart (61)
58	51	0x160	0x51	disable cips cde_uart (48)
59	52	0x160	0x52	pass cips cde (88)
60	53	0x160	0x53	select cips cde_chnl (92)
61	54	0x160	0x54	stuff cips cde_memory (111)
62	55	0x160	0x60	suspend cips rel_tm (116)
63	56	0x160	0x61	suspend cips abs_tm (115)
64	57	0x160	0x62	term cips sequence (117)
65	58	0x160	0x63	init cips sequence (77)
66	59	0x160	0x70	set cips sw_stat_rt (107)
67	60	0x160	0x71	set cips hw_hk_rt (101)
68	61	0x160	0x72	issue cips sw_stat (84)
69	62	0x160	0x73	issue cips hw_hk (81)
70	63	0x160	0x74	issue cips evt_log (79)
71	64	0x160	0x75	enable cips em (63)
72	65	0x160	0x76	disable cips em (51)
73	66	0x160	0x77	set cips em_mask (99)
74	67	0x160	0x78	enable cips tlm_out (70)
75	68	0x160	0x79	disable cips tlm_out (56)

Command Opcode Summary, cont'd

cmd ref.	apid ref.	apid (hex)	opcode (hex)	command
76	69	0x160	0x7a	set cips to_mask (110)
77	70	0x160	0x7c	dump cips memory (57)
78	71	0x160	0x7d	set cips cde_itf_rt (96)
79	72	0x160	0x7e	issue cips pm (82)
80	73	0x160	0x7f	issue cips exception (80)
81	74	0x160	0x80	open cips dust_cover (86)
82	75	0x160	0x81	enable cips bos (59)
83	76	0x160	0x82	disable cips bos (46)
84	77	0x160	0x84	turn on cips camera (122)
85	78	0x160	0x85	turn off cips camera (117)
86	79	0x160	0x86	enable cips lens_htr (65)
87	80	0x160	0x87	disable cips lens_htr (52)
88	81	0x160	0xfb	wake cips (126)
89	82	0x160	0xfc	inform cips sc_clock (73)
90	83	0x160	0xfd	inform cips pwrdown (73)
91	84	0x160	0xfd	safe cips (91)
92	85	0x160	0xfe	inform cips acs_state (71)

C. COMMAND OPCODE SUMMARY

D

Command Packet Summary

Command Packet Summary

pkt no.	apid (hex)	packet
1	0x15d	inst
2	0x15e	inst
3	0x160	cips
4	0x161	cips
5	0x162	cips
6	0x163	cips

D. COMMAND PACKET SUMMARY

E

Telemetry Measurement List

OASIS-CC/FSW database version TBD, Mon Feb 19 11:16:32 2007.

E.1 cips a_seq_addr

Description

Address in memory where the current autonomous sequence resides.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips a_seq_page (141)	Page in memory where the current autonomous sequence resides.
cips a_seq_cur_pos (140)	Address of the current command in the current autonomous sequence.
cips a_seq_length (140)	Length of the current autonomous sequence.

E.2 cips a_seq_cmd_f_code

Description

Reason for failure of the last command that failed execution.

Data Type

16 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
SUCCESS	0	GOOD
OPCODE_ERR	1	BAD
LEN_ERR	2	BAD
CHKSUM_ERR	3	BAD
FMT_ERR	4	BAD
RANGE_ERR	5	BAD
INVALID_STATE	6	BAD
INVALID_APID	7	BAD
METHOD_ERR	8	BAD

Telemetry Source Packets**cips sw_status (514)****Related Telemetry**

<i>telemetry</i>	<i>description</i>
cips a_seq_cmd_f_ct (138)	Number of commands that failed execution by the autonomous sequence engine.
cips a_seq_cmd_fail (139)	Opcode of the last command that failed execution by the autonomous seq engine.

E.3 cips a_seq_cmd_f_ct**Description**

Number of commands that failed execution by the autonomous sequence engine.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets**cips sw_status (514)**

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips a_seq_cmd_f.code (137)	Reason for failure of the last command that failed execution.
cips a_seq_cmd_fail (139)	Opcode of the last command that failed execution by the autonomous seq engine.
cips a_seq_cmd_s.ct (139)	Number of commands sucessfully executed by the autonomous sequence engine.

E.4 cips a_seq_cmd_fail**Description**

Opcode of the last command that failed execution by the autonomous seq engine.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips a_seq_cmd_f.code (137)	Reason for failure of the last command that failed execution.
cips a_seq_cmd_f.ct (138)	Number of commands that failed execution by the autonomous sequence engine.
cips a_seq_cmd_succ (140)	Opcode of the last command executed by the autonomous sequence engine.

E.5 cips a_seq_cmd_s_ct**Description**

Number of commands sucessfully executed by the autonomous sequence engine.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips a_seq_cmd_succ (140)	Opcode of the last command executed by the autonomous sequence engine.
cips a_seq_cmd_f.ct (138)	Number of commands that failed execution by the autonomous sequence engine.

E.6 cips a_seq_cmd_succ

Description

Opcode of the last command executed by the autonomous sequence engine.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips a_seq_cmd_s_ct (139)	Number of commands successfully executed by the autonomous sequence engine.
cips a_seq_cmd_fail (139)	Opcode of the last command that failed execution by the autonomous seq engine.

E.7 cips a_seq_cur_pos

Description

Address of the current command in the current autonomous sequence.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips a_seq_page (141)	Page in memory where the current autonomous sequence resides.
cips a_seq_addr (137)	Address in memory where the current autonomous sequence resides.
cips a_seq_length (140)	Length of the current autonomous sequence.
cips sram_img_1_pg (442)	Page in memory where SRAM Code image 1 starts.
cips sram_img_2_pg (444)	Page in memory where SRAM Code image 2 starts.

E.8 cips a_seq_length

Description

Length of the current autonomous sequence.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips a_seq_page (141)	Page in memory where the current autonomous sequence resides.
cips a_seq_addr (137)	Address in memory where the current autonomous sequence resides.
cips a_seq_length (140)	Length of the current autonomous sequence.
cips a_seq_cur_pos (140)	Address of the current command in the current autonomous sequence.
cips sram_img_1.pg (442)	Page in memory where SRAM Code image 1 starts.
cips sram_img_2.pg (444)	Page in memory where SRAM Code image 2 starts.

E.9 cips a_seq_page**Description**

Page in memory where the current autonomous sequence resides.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
O_SRAM_P0	0	BAD
O_SRAM_P1	1	BAD
O_SRAM_P2	2	GOOD
O_SRAM_P3	3	BAD
O_SRAM_P4	4	BAD
O_SRAM_P5	5	BAD
O_SRAM_P6	6	BAD
O_SRAM_P7	7	BAD
O_SRAM_P8	8	BAD
O_SRAM_P9	9	BAD
O_SRAM_P10	10	BAD
O_SRAM_P11	11	BAD
O_EEPROM_P12	12	BAD
O_EEPROM_P13	13	BAD
O_EEPROM_P14	14	BAD
O_EEPROM_P15	15	BAD
IO	128	BAD
I_SRAM	255	BAD

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips a_seq_page (141)	Page in memory where the current autonomous sequence resides.
cips a_seq_cur_pos (140)	Address of the current command in the current autonomous sequence.
cips a_seq_length (140)	Length of the current autonomous sequence.
cips a_seq_addr (137)	Address in memory where the current autonomous sequence resides.
cips sram_img_1_pg (442)	Page in memory where SRAM Code image 1 starts.
cips sram_img_2_pg (444)	Page in memory where SRAM Code image 2 starts.

E.10 cips a_seq_state**Description**

Current state of the autonomous sequence engine.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
IDLE	0	GOOD
INIT	1	GOOD
ACTIVE	2	GOOD
WAIT	3	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
suspend cips abs_tm (115)	Wait until an absolute time.
term cips sequence (117)	Stop the current command sequence in the given sequence engine.
suspend cips rel_tm (116)	Wait for a relative time delay.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips a_seq_wait_time (144)	Wait time for execution of the next command in the autonomous sequence engine.
cips a_seq_wait_type (145)	Type of wait currently being executed in the autonomous sequence engine.

E.11 cips a_seq_stop_code**Description**

Reason command processing in the autonomous sequence engine stopped.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
NONE	0	GOOD
END_OF_SEQUENCE	1	GOOD
COMMANDED	2	GOOD
STATE_ERROR	3	BAD
LENGTH_ERROR	4	BAD
USE_COUNT_ERROR	5	BAD
PAGE_ERROR	6	BAD
CHECKSUM_ERROR	7	BAD
PARSE_ERROR	8	BAD

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
term cips sequence (117)	Stop the current command sequence in the given sequence engine.

E.12 cips a_seq_use_ct**Description**

Number of uses remaining for the last autonomous sequence initiated.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.13 cips a_seq_wait_time**Description**

Wait time for execution of the next command in the autonomous sequence engine.

Data Type

32 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Discussion

This can be a relative time (i.e. number of seconds left in the wait), or an absolute time (i.e. 2005/300-12:15:59), based on the type of 'suspend' command executed.

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
suspend cips abs_tm (115)	Wait until an absolute time.
suspend cips rel_tm (116)	Wait for a relative time delay.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips a_seq_state (142)	Current state of the autonomous sequence engine.
cips a_seq_wait_type (145)	Type of wait currently being executed in the autonomous sequence engine.

E.14 cips a_seq_wait_type

Description

Type of wait currently being executed in the autonomous sequence engine.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
NONE	0	GOOD
REL	1	GOOD
ABS	2	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
suspend cips abs_tm (115)	Wait until an absolute time.
suspend cips rel_tm (116)	Wait for a relative time delay.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips a_seq_wait_time (144)	Wait time for execution of the next command in the autonomous sequence engine.
cips a_seq_state (142)	Current state of the autonomous sequence engine.

E.15 cips acs_sun_uv_x

Description

Spacecraft to sun unit vector, Cartesian x coordinate.

Data Type

64 bits, float, engineering units = any.

Measurement Source

spacecraft bus

Discussion

This telemetry item is an echo of the value passed to the instrument in the ACS status message.

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
inform cips.acs.state (71)	Spacecraft ACS status update command.

E.16 cips.acs.sun.uv.y**Description**

Spacecraft to sun unit vector, Cartesian y coordinate.

Data Type

64 bits, float, engineering units = any.

Measurement Source

spacecraft bus

Discussion

This telemetry item is an echo of the value passed to the instrument in the ACS status message.

Telemetry Source Packets

cips.sw.status (514)

Related Commands

<i>command</i>	<i>description</i>
inform cips.acs.state (71)	Spacecraft ACS status update command.

E.17 cips.acs.sun.uv.z**Description**

Spacecraft to sun unit vector, Cartesian z coordinate.

Data Type

64 bits, float, engineering units = any.

Measurement Source

spacecraft bus

Discussion

This telemetry item is an echo of the value passed to the instrument in the ACS status message.

Telemetry Source Packets

cips.sw.status (514)

Related Commands

<i>command</i>	<i>description</i>
inform cips.acs.state (71)	Spacecraft ACS status update command.

E.18 cips active_csci

Description

Software currently being executed on CIPS.

Data Type

16 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
SRAM.Other	21767	CAUTION
SRAM.Test	21886	CAUTION
SRAM.Debug	21982	CAUTION
SRAM.Flight	22001	GOOD
SuROM.Other	43527	CAUTION
SuROM.Test	43646	CAUTION
SuROM.Debug	43742	CAUTION
SuROM.Flight	43761	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Commands

command	description
init cips sram.code (78)	Command the CIPS SuROM to switch to the SRAM code.

Related Telemetry

telemetry	description
cips csci.version.pl (200)	Patch level of the software currently being executed on CIPS.
cips csci.version.num (200)	Version number of the software currently being executed on CIPS.
cips boot.error (150)	CIPS boot error status.

E.19 cips adc_lup

Description

CIPS camera ADC latchup status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

camera fpga

State Conversions

state	value	desirability
OK	0	GOOD
ERROR	1	BAD

Discussion

The estimated latchup rate is 6E-6 latchups/day in the AIM orbit. If a latchup does occur, the internal latchup protect circuit should clear the event automatically within 1 to 2 msec. Thus, worst case, we expect that one horizontal line in the image could be affected by this condition. The effect is that all data should be zero during the latchup condition.

Telemetry Source Packets

cips sci_mx_cam (502), cips sci_my_cam (505), cips sci_px_cam (508), cips sci_py_cam (511)

E.20 cips armed**Description**

Safety state of the cips command receiver.

Data Type

16 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
NOT_ARMED	0	GOOD
ARMED	1	CAUTION

Discussion

This telemetry item should be set only immediately prior to receipt of a 'hazardous' command, as it is set when the command interpreter sees an 'arm' command. As soon as the next command is received, be it 'hazardous' or 'safe', the armed status is cleared, and thus the telemetry point is reset to not_armed.

Telemetry Source Packets

cips sw_status (514)

Related Commands

command	description
reset cips arm (89)	Disarm CIPS software hazardous command acceptance.
arm cips (43)	Arm CIPS software for a subsequent hazardous command.

E.21 cips bits_per_pixel**Description**

Bits per pixel.

Data Type

8 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sci_mx_cam (502), cips sci_my_cam (505), cips sci_px_cam (508), cips sci_py_cam (511)

E.22 cips bits_per_pixel_c**Description**

Bits per pixel in the current image.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.23 cips boot_ct_thresh**Description**

CIPS warm boot count threshold.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Discussion

This threshold defines the point at which CIPS will cease trying autonomous quick boots. The value of this parameter is not important when CIPS is in LONG boot mode.

Telemetry Source Packets

cips sw_status (514)

E.24 cips boot_error

Description

CIPS boot error status.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
FALSE	0	GOOD
TRUE	1	BAD

Discussion

In SuROM code, this telemetry point is used to indicate whether or not CIPS booted without error. If no errors were detected on boot, it is safe for operators to use the 'init cips sram_code' command to transition CIPS to SRAM Code. If errors were detected on boot, the CIPS event log will contain the information on what errors were detected.

This telemetry point should always be 'FALSE' for SRAM Code.

Telemetry Source Packets

cips sw_status (514)

Related Commands

command	description
init cips sram_code (78)	Command the CIPS SuROM to switch to the SRAM code.

Related Telemetry

telemetry	description
cips active_csci (147)	Software currently being executed on CIPS.

E.25 cips bos_trip_ct

Description

Cumulative count of the number of times the CIPS BOS have tripped.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.26 cips cam_bin_ena**Description**

Describes if camera FPGA binning was enabled for this image.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
ENA	0	GOOD
DIS	1	GOOD

Telemetry Source Packets

cips sci_mx_cam (502), cips sci_my_cam (505), cips sci_px_cam (508), cips sci_py_cam (511)

E.27 cips cam_bin_enable**Description**

Last commanded CIPS camera binning enable setting.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

camera fpga

State Conversions

state	value	desirability
FALSE	0	GOOD
TRUE	1	GOOD

Telemetry Source Packets**cips sw_status** (514)**Related Commands**

<i>command</i>	<i>description</i>
disable cips cam_bin (47)	Disable the cameras' science data binning capability.
enable cips cam_bin (60)	Enable the cameras' science data binning capability.

E.28 cips cam_fpga_ver**Description**

CIPS camera FPGA version.

Data Type

8 bits, unsigned integer, engineering units = any.

Measurement Source

hk fpga

Telemetry Source Packets**cips sci_mx_cam** (502), **cips sci_my_cam** (505), **cips sci_px_cam** (508), **cips sci_py_cam** (511)**E.29 cips cami_fpga_ver****Description**

CIPS camera interface (CAMI) board FPGA version.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

processor fpga

Telemetry Source Packets**cips hw_hk** (492)**E.30 cips cami_pcb_temp****Description**

Camera Interface printed circuit board temperature.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Limits

alarm	limit
red high	59
yellow high	54
yellow low	-35
red low	-40
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.31 cips ccd_col_bin**Description**

CIPS CCD column binning status.

Data Type

3 bits, discrete state, engineering units = dn.

Measurement Source

camera fpga

State Conversions

state	value	desirability
OFF	1	GOOD
ON_x.2	2	GOOD
ON_x.3	3	CAUTION
ON_x.4	4	CAUTION

Telemetry Source Packets

cips sci_mx_cam (502), cips sci_my_cam (505), cips sci_px_cam (508), cips sci_py_cam (511)

E.32 cips ccd_row_bin

Description

CIPS CCD row binning status.

Data Type

3 bits, discrete state, engineering units = dn.

Measurement Source

camera fpga

State Conversions

state	value	desirability
OFF	1	GOOD
ON_x_2	2	GOOD
ON_x_3	3	CAUTION
ON_x_4	4	CAUTION
ON_x_5	5	CAUTION
ON_x_6	6	CAUTION

Telemetry Source Packets

cips sci_mx_cam (502), cips sci_my_cam (505), cips sci_px_cam (508), cips sci_py_cam (511)

E.33 cips cde_cal_to

Description

State of downlink enable for CDE calibration telemetry packets.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
DISABLED	0	CAUTION
ENABLED	1	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips to_mask (110)	Set the CIPS telemetry output enable mask.
disable cips tlm_out (56)	Disable downlink of the given CIPS telemetry packet.
enable cips tlm_out (70)	Enable downlink of the given CIPS telemetry packet.

E.34 cips cde_cmd_rbuf_wds

Description

Number of words of data currently in the CDE command ring buffer.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.35 cips cde_flash_to

Description

State of downlink enable for CDE flash summary telemetry packets.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
DISABLED	0	CAUTION
ENABLED	1	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips to_mask (110)	Set the CIPS telemetry output enable mask.
disable cips tlm_out (56)	Disable downlink of the given CIPS telemetry packet.
enable cips tlm_out (70)	Enable downlink of the given CIPS telemetry packet.

E.36 cips cde_hb_fluttr_ct

Description

Counter for receipt of successive CDE tlm ITFs with the same heartbeat value.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.37 cips cde_hist_log_to**Description**

State of downlink enable for CDE history log telemetry packets.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
DISABLED	0	CAUTION
ENABLED	1	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips to_mask (110)	Set the CIPS telemetry output enable mask.
disable cips tlm_out (56)	Disable downlink of the given CIPS telemetry packet.
enable cips tlm_out (70)	Enable downlink of the given CIPS telemetry packet.

E.38 cips cde_init_flag**Description**

Flag used to determine if communications with CDE should be initiated on startup

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
DISABLE	0	CAUTION
ENABLE	21845	GOOD

Telemetry Source Packets

cips sw_status (514)

E.39 cips cde_itf_rt**Description**

Packet production period for CIPS software status packets.

Data Type

16 bits, unsigned integer, engineering units = sec.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.40 cips cde_itf_to**Description**

State of downlink enable for CDE ITF data telemetry packets.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
DISABLED	0	CAUTION
ENABLED	1	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips to_mask (110)	Set the CIPS telemetry output enable mask.
disable cips tlm_out (56)	Disable downlink of the given CIPS telemetry packet.
enable cips tlm_out (70)	Enable downlink of the given CIPS telemetry packet.

E.41 cips_cde_mem_dump_to**Description**

State of downlink enable for CDE memory dump telemetry packets.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
DISABLED	0	CAUTION
ENABLED	1	GOOD

Telemetry Source Packets

cips_sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips to_mask (110)	Set the CIPS telemetry output enable mask.
disable cips tlm_out (56)	Disable downlink of the given CIPS telemetry packet.
enable cips tlm_out (70)	Enable downlink of the given CIPS telemetry packet.

E.42 cips_cde_sci_to**Description**

State of downlink enable for CDE science telemetry packets.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
DISABLED	0	CAUTION
ENABLED	1	GOOD

Telemetry Source Packets**cips sw_status** (514)**Related Commands**

<i>command</i>	<i>description</i>
set cips to_mask (110)	Set the CIPS telemetry output enable mask.
disable cips tlm_out (56)	Disable downlink of the given CIPS telemetry packet.
enable cips tlm_out (70)	Enable downlink of the given CIPS telemetry packet.

E.43 cips cde_sys_data_to**Description**

State of downlink enable for CDE system data telemetry packets.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
DISABLED	0	CAUTION
ENABLED	1	GOOD

Telemetry Source Packets**cips sw_status** (514)**Related Commands**

<i>command</i>	<i>description</i>
set cips to_mask (110)	Set the CIPS telemetry output enable mask.
disable cips tlm_out (56)	Disable downlink of the given CIPS telemetry packet.
enable cips tlm_out (70)	Enable downlink of the given CIPS telemetry packet.

E.44 cips cde_tlm_rbuf_wds**Description**

Number of words of data currently in the CDE telemetry ring buffer.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets**cips sw_status** (514)

E.45 cips cde_uart_f_err

Description

CDE UART receiver format error (did not get correct stop bit).

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
OK	0	GOOD
ERROR	1	BAD

Telemetry Source Packets

cips hw_hk (492)

E.46 cips cde_uart_rx_af

Description

CDE UART receive FIFO almost full (contains 31 or more bytes).

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
OK	0	GOOD
A_FULL	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

E.47 cips cde_uart_rx_e

Description

CDE UART receive FIFO empty status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
DATA_READY	0	GOOD
EMPTY	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

E.48 cips cde_uart_rx_f**Description**

CDE UART receive FIFO fill status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
OK	0	GOOD
FULL	1	BAD

Telemetry Source Packets

cips hw_hk (492)

E.49 cips cde_uart_rx_or**Description**

CDE UART receiver overrun error (receiver FIFO full and another byte received).

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
OK	0	GOOD
ERROR	1	BAD

Telemetry Source Packets

cips hw_hk (492)

E.50 cips cde_uart_rx_r**Description**

CDE UART receive FIFO ready status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
EMPTY	0	GOOD
DATA_READY	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

E.51 cips cde_uart_rx_r_mk**Description**

CDE UART receive FIFO ready interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

Related Commands

<i>command</i>	<i>description</i>
disable cips cde_uart (48)	Disable communications between the CIPS and CDE instruments.
enable cips cde_uart (61)	Enable communications between the CIPS and CDE instruments.

E.52 cips cde_uart_rxor_mk

Description

CDE UART receiver overrun error interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

Related Commands

<i>command</i>	<i>description</i>
disable cips cde_uart (48)	Disable communications between the CIPS and CDE instruments.
enable cips cde_uart (61)	Enable communications between the CIPS and CDE instruments.

E.53 cips cde_uart_tx_ae

Description

CDE UART transmit FIFO almost empty (contains one byte or less) status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
OK	0	GOOD
A_EMPTY	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

E.54 cips cde_uart_tx_af**Description**

CDE UART transmit FIFO almost full (contains 31 or more bytes) status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
OK	0	GOOD
A_FULL	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

E.55 cips cde_uart_tx_b**Description**

CDE UART transmitter state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
NOT_BUSY	0	GOOD
BUSY	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

E.56 cips cde_uart_tx_e**Description**

CDE UART transmit FIFO empty status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
OK	0	GOOD
EMPTY	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

E.57 cips cde_uart_tx_e_mk**Description**

CDE UART transmit FIFO empty interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

Related Commands

<i>command</i>	<i>description</i>
disable cips cde_uart (48)	Disable communications between the CIPS and CDE instruments.
enable cips cde_uart (61)	Enable communications between the CIPS and CDE instruments.

E.58 cips cde_uart_tx_f**Description**

CDE UART transmit FIFO full status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
OK	0	GOOD
FULL	1	BAD

Telemetry Source Packets

cips hw_hk (492)

E.59 cips ce_apid**Description**

Command echo CCSDS APID.

Data Type

11 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips cmd_echo (480)

E.60 cips ce_chksum**Description**

Command echo secondary header checksum.

Data Type

8 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips cmd_echo (480)

E.61 cips ce_cmd_data**Description**

Command echo command subfield data.

Data Type

1976 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips cmd_echo (480)

E.62 cips ce_func_code**Description**

Command echo secondary header function code.

Data Type

7 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips cmd_echo (480)

E.63 cips ce_opcode**Description**

Command echo command opcode.

Data Type

8 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips cmd_echo (480)

E.64 cips ce_pkt_len**Description**

Command echo CCSDS packet length.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets**cips cmd_echo (480)****E.65 cips ce_sec_hdr****Description**

Command echo CCSDS secondary header flag.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
NO	0	BAD
YES	1	GOOD

Telemetry Source Packets**cips cmd_echo (480)****E.66 cips ce_seq_ct****Description**

Command echo CCSDS sequence count.

Data Type

14 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets**cips cmd_echo (480)****E.67 cips ce_seq_flags****Description**

Command echo CCSDS sequence flags.

Data Type

2 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips cmd_echo (480)

E.68 cips ce_type**Description**

Command echo CCSDS packet type.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
TCSP	1	GOOD

Telemetry Source Packets

cips cmd_echo (480)

E.69 cips ce_version**Description**

Command echo CCSDS version number.

Data Type

3 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
V_0	0	GOOD
V_1	1	BAD

Telemetry Source Packets

cips cmd_echo (480)

E.70 cips check_ccd_cmd_p

Description

Processor FPGA enable flag for parity checking on received CCD commands.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
FALSE	0	CAUTION
TRUE	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

E.71 cips chksum_error

Description

CIPS image checksum status.

Data Type

2 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
UNKNOWN	0	GOOD
TRUE	1	BAD
FALSE	2	GOOD

Discussion

Checksums for the pixels in each image are calculated at the time those data are written to SDRAM by the CAMI FPGA. When those data are read out of the SDRAM by the microprocessor, the checksums are again calculated and compared to the stored values computed by the CAMI FPGA. This telemetry item shows the status of this check.

Since image data can come down in multiple CCSDS packets, the checksum will change from UNKNOWN to either ERROR or NO_ERROR in the last packet for that image.

Telemetry Source Packets

cips sci_mx_cam (502), cips sci_my_cam (505), cips sci_px_cam (508), cips sci_py_cam (511)

E.72 cips_clock_tick_bits

Description

Number of bits per CIPS clock 'tick'.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Discussion

Each bit represents 10 microseconds.

Telemetry Source Packets

cips_sw_status (514)

E.73 cips_clock_tick_ssec

Description

Number of subseconds per CIPS clock 'tick'.

Data Type

32 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Discussion

Each bit represents 2^{-32} seconds.

Telemetry Source Packets

cips_sw_status (514)

E.74 cips_cmd_echo_to

Description

State of downlink enable for CIPS command echo telemetry packets.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
DISABLED	0	GOOD
ENABLED	1	CAUTION

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips to_mask (110)	Set the CIPS telemetry output enable mask.
disable cips tlm_out (56)	Disable downlink of the given CIPS telemetry packet.
enable cips tlm_out (70)	Enable downlink of the given CIPS telemetry packet.

E.75 cips cmd_fail**Description**

Last cips failed command.

Data Type

8 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
noop cips (85)	No operation, or do nothing command.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips cmd_fail_code (173)	Last cips failed command reason code.
cips cmd_succ (175)	Last cips successful command.
cips cmd_fail_ct (173)	Cumulative count of cips failed commands.

E.76 cips_cmd_fail_code

Description

Last cips failed command reason code.

Data Type

16 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
SUCCESS	0	GOOD
OPCODE_ERR	1	BAD
LEN_ERR	2	BAD
CHKSUM_ERR	3	BAD
FMT_ERR	4	BAD
RANGE_ERR	5	BAD
INVALID_STATE	6	BAD
INVALID_APID	7	BAD
METHOD_ERR	8	BAD

Telemetry Source Packets

cips_sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips_cmd_fail_ct (173)	Cumulative count of cips failed commands.
cips_cmd_fail (172)	Last cips failed command.

E.77 cips_cmd_fail_ct

Description

Cumulative count of cips failed commands.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

Related Commands

command	description
noop cips (85)	No operation, or do nothing command.

Related Telemetry

telemetry	description
cips cmd_fail_code (173)	Last cips failed command reason code.
cips cmd_succ_ct (175)	Cumulative count of cips successful commands.
cips cmd_fail (172)	Last cips failed command.

E.78 cips cmd_loss_time**Description**

Time since CIPS received a command across the 1553 bus.

Data Type

16 bits, unsigned integer, engineering units = sec.

Measurement Source

software

Limits

alarm	limit
red high	3
yellow high	1
yellow low	-1
red low	-1
delta	none

Telemetry Source Packets

cips sw_status (514)

E.79 cips cmd_rbuf_wds

Description

Number of words of data currently in the command ring buffer.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.80 cips cmd_succ

Description

Last cips successful command.

Data Type

8 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
noop cips (85)	No operation, or do nothing command.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips cmd_fail (172)	Last cips failed command.

E.81 cips cmd_succ_ct

Description

Cumulative count of cips successful commands.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets**cips sw_status** (514)**Related Commands**

<i>command</i>	<i>description</i>
noop cips (85)	No operation, or do nothing command.
set cips cmn_a_size (98)	Set the size of the CCD's 'Common Area' for all cameras.
copy cips memory (45)	Copy data from one location to another.
reset cips sw_stat (91)	Reset values of counters and error indicators in the software status packet.
stuff cips cde_memory (111)	Load data to CDE memory.
set cips cmn_a_offset (97)	Set the offsets for the CCD's 'Common Area' for all cameras.
pass cips cde (88)	Pass a low-level command ITF to the CDE instrument.
stuff cips memory_wd (113)	Load a single 16-bit word of data to memory or I/O space.
set cips img_offset (103)	Set the raw CCD image origin offset from the top and side of the CCD.
select cips tdi_dir (95)	Select the direction in which Time Delayed Integration is done across the CCD.
wake cips (126)	Remove cips safing constraints, allowing transition out of safe hold.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips cmd_fail_ct (173)	Cumulative count of cips failed commands.

E.82 cips columns_per_map**Description**

Number of columns in a single image 'map' in SDRAM.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets**cips sw_status** (514)**E.83 cips comm_sram_a16****Description**

Value of bit 16 in communications (1553) SRAM addresses.

Data Type

1 bits, unsigned integer, engineering units = dn.

Measurement Source

proc fpga

Telemetry Source Packets

cips hw_hk (492)

E.84 cips comp_scheme**Description**

Type of compression performed on image data within this packet.

Data Type

3 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
NONE	0	GOOD
ROOT2N	1	GOOD

Telemetry Source Packets

cips sci_mx_cam (502), cips sci_my_cam (505), cips sci_px_cam (508), cips sci_py_cam (511)

E.85 cips cpa_p3_3v_mon**Description**

CIPS CPA +3.3 volt monitor.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-22.25
c1	0.002716064453125
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Limits

alarm	limit
red high	3.46000003814697
yellow high	3.40000009536743
yellow low	3.20000004768372
red low	3.14000010490417
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.86 cips cpu_util**Description**

Current CIPS CPU utilization.

Data Type

16 bits, unsigned integer, engineering units = pc.

Measurement Source

software

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	0
c0	0
c1	0.00305185094475746
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Telemetry Source Packets

cips sw_status (514)

E.87 cips cpu_util_max**Description**

Maximum CIPS CPU utilization measured since last reset/reboot.

Data Type

16 bits, unsigned integer, engineering units = pc.

Measurement Source

software

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	0
c0	0
c1	0.00305185094475746
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Telemetry Source Packets

cips sw_status (514)

E.88 cips crc**Description**

CRC value for the operator-initiated CRC calculation.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
init cips crc (74)	Checksum CIPS SRAM or EEPROM memory.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_type (198)	Page in CIPS memory to be CRC'd, operator-initiated CRC calculation.
cips crc_addr (197)	Buffer start address for operator-initiated CRC calculation.
cips crc_valid (198)	Validity flag for the operator-initiated CRC calculation.
cips crc_length (197)	Length of memory to be CRC'd, CRC Array slot 1.

E.89 cips crc_1

Description

CRC value for the indicated buffer space, CRC Array slot 1.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_1.valid (181)	Validity flag for the related CRC calculation, CRC Array slot 1.
cips crc_1.length (180)	Length of memory to be CRC'd, CRC Array slot 1.
cips crc_1.addr (180)	Buffer start address for CRC calculation, CRC Array slot 1.
cips crc_1.type (181)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 1.

E.90 cips crc_1_addr

Description

Buffer start address for CRC calculation, CRC Array slot 1.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_1.valid (181)	Validity flag for the related CRC calculation, CRC Array slot 1.
cips crc_1.length (180)	Length of memory to be CRC'd, CRC Array slot 1.
cips crc_1 (180)	CRC value for the indicated buffer space, CRC Array slot 1.
cips crc_1.type (181)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 1.

E.91 cips crc_1_length

Description

Length of memory to be CRC'd, CRC Array slot 1.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_1.valid (181)	Validity flag for the related CRC calculation, CRC Array slot 1.
cips crc_1.addr (180)	Buffer start address for CRC calculation, CRC Array slot 1.
cips crc_1 (180)	CRC value for the indicated buffer space, CRC Array slot 1.
cips crc_1.type (181)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 1.

E.92 cips crc_1_type**Description**

Type of/page in CIPS memory for to be CRC'd, CRC Array slot 1.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_1.length (180)	Length of memory to be CRC'd, CRC Array slot 1.
cips crc_1.addr (180)	Buffer start address for CRC calculation, CRC Array slot 1.
cips crc_1 (180)	CRC value for the indicated buffer space, CRC Array slot 1.
cips crc_1.valid (181)	Validity flag for the related CRC calculation, CRC Array slot 1.

E.93 cips crc_1_valid**Description**

Validity flag for the related CRC calculation, CRC Array slot 1.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
IN_PROGRESS	0	GOOD
VALID	21845	GOOD

Telemetry Source Packets**cips sw_status** (514)**Related Telemetry**

<i>telemetry</i>	<i>description</i>
cips crc_1.length (180)	Length of memory to be CRC'd, CRC Array slot 1.
cips crc_1.addr (180)	Buffer start address for CRC calculation, CRC Array slot 1.
cips crc_1 (180)	CRC value for the indicated buffer space, CRC Array slot 1.
cips crc_1.type (181)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 1.

E.94 cips crc_2**Description**

CRC value for the indicated buffer space, CRC Array slot 2.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets**cips sw_status** (514)**Related Telemetry**

<i>telemetry</i>	<i>description</i>
cips crc_2.valid (184)	Validity flag for the related CRC calculation, CRC Array slot 2.
cips crc_2.type (183)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 2.
cips crc_2.length (183)	Length of memory to be CRC'd, CRC Array slot 2.
cips crc_2.addr (182)	Buffer start address for CRC calculation, CRC Array slot 2.

E.95 cips crc_2_addr**Description**

Buffer start address for CRC calculation, CRC Array slot 2.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_2.valid (184)	Validity flag for the related CRC calculation, CRC Array slot 2.
cips crc_2.type (183)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 2.
cips crc_2.length (183)	Length of memory to be CRC'd, CRC Array slot 2.
cips crc_2 (182)	CRC value for the indicated buffer space, CRC Array slot 2.

E.96 cips crc_2_length**Description**

Length of memory to be CRC'd, CRC Array slot 2.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_2.valid (184)	Validity flag for the related CRC calculation, CRC Array slot 2.
cips crc_2.type (183)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 2.
cips crc_2 (182)	CRC value for the indicated buffer space, CRC Array slot 2.
cips crc_2.addr (182)	Buffer start address for CRC calculation, CRC Array slot 2.

E.97 cips crc_2_type**Description**

Type of/page in CIPS memory for to be CRC'd, CRC Array slot 2.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets**cips sw_status** (514)**Related Telemetry**

<i>telemetry</i>	<i>description</i>
cips crc_2_length (183)	Length of memory to be CRC'd, CRC Array slot 2.
cips crc_2 (182)	CRC value for the indicated buffer space, CRC Array slot 2.
cips crc_2_addr (182)	Buffer start address for CRC calculation, CRC Array slot 2.
cips crc_2_valid (184)	Validity flag for the related CRC calculation, CRC Array slot 2.

E.98 cips crc_2_valid**Description**

Validity flag for the related CRC calculation, CRC Array slot 2.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
IN_PROGRESS	0	GOOD
VALID	21845	GOOD

Telemetry Source Packets**cips sw_status** (514)**Related Telemetry**

<i>telemetry</i>	<i>description</i>
cips crc_2_type (183)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 2.
cips crc_2_length (183)	Length of memory to be CRC'd, CRC Array slot 2.
cips crc_2 (182)	CRC value for the indicated buffer space, CRC Array slot 2.
cips crc_2_addr (182)	Buffer start address for CRC calculation, CRC Array slot 2.

E.99 cips crc_3**Description**

CRC value for the indicated buffer space, CRC Array slot 3.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets**cips sw_status** (514)**Related Telemetry**

<i>telemetry</i>	<i>description</i>
cips crc_3.length (185)	Length of memory to be CRC'd, CRC Array slot 3.
cips crc_3.valid (186)	Validity flag for the related CRC calculation, CRC Array slot 3.
cips crc_3.type (186)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 3.
cips crc_3.addr (185)	Buffer start address for CRC calculation, CRC Array slot 3.

E.100 cips crc_3_addr**Description**

Buffer start address for CRC calculation, CRC Array slot 3.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets**cips sw_status** (514)**Related Telemetry**

<i>telemetry</i>	<i>description</i>
cips crc_3.length (185)	Length of memory to be CRC'd, CRC Array slot 3.
cips crc_3.valid (186)	Validity flag for the related CRC calculation, CRC Array slot 3.
cips crc_3.type (186)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 3.
cips crc_3 (184)	CRC value for the indicated buffer space, CRC Array slot 3.

E.101 cips crc_3_length**Description**

Length of memory to be CRC'd, CRC Array slot 3.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets**cips sw_status** (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_3.valid (186)	Validity flag for the related CRC calculation, CRC Array slot 3.
cips crc_3.type (186)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 3.
cips crc_3.addr (185)	Buffer start address for CRC calculation, CRC Array slot 3.
cips crc_3 (184)	CRC value for the indicated buffer space, CRC Array slot 3.

E.102 cips crc_3_type**Description**

Type of/page in CIPS memory for to be CRC'd, CRC Array slot 3.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_3.length (185)	Length of memory to be CRC'd, CRC Array slot 3.
cips crc_3.addr (185)	Buffer start address for CRC calculation, CRC Array slot 3.
cips crc_3 (184)	CRC value for the indicated buffer space, CRC Array slot 3.
cips crc_3.valid (186)	Validity flag for the related CRC calculation, CRC Array slot 3.

E.103 cips crc_3_valid**Description**

Validity flag for the related CRC calculation, CRC Array slot 3.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
IN_PROGRESS	0	GOOD
VALID	21845	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_3.length (185)	Length of memory to be CRC'd, CRC Array slot 3.
cips crc_3.type (186)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 3.
cips crc_3.addr (185)	Buffer start address for CRC calculation, CRC Array slot 3.
cips crc_3 (184)	CRC value for the indicated buffer space, CRC Array slot 3.

E.104 cips crc_4**Description**

CRC value for the indicated buffer space, CRC Array slot 4.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_4.valid (189)	Validity flag for the related CRC calculation, CRC Array slot 4.
cips crc_4.addr (187)	Buffer start address for CRC calculation, CRC Array slot 4.
cips crc_4.type (188)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 4.
cips crc_4.length (188)	Length of memory to be CRC'd, CRC Array slot 4.

E.105 cips crc_4_addr**Description**

Buffer start address for CRC calculation, CRC Array slot 4.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_4 (187)	CRC value for the indicated buffer space, CRC Array slot 4.
cips crc_4.valid (189)	Validity flag for the related CRC calculation, CRC Array slot 4.
cips crc_4.type (188)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 4.
cips crc_4.length (188)	Length of memory to be CRC'd, CRC Array slot 4.

E.106 cips crc_4_length**Description**

Length of memory to be CRC'd, CRC Array slot 4.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_4 (187)	CRC value for the indicated buffer space, CRC Array slot 4.
cips crc_4.valid (189)	Validity flag for the related CRC calculation, CRC Array slot 4.
cips crc_4.addr (187)	Buffer start address for CRC calculation, CRC Array slot 4.
cips crc_4.type (188)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 4.

E.107 cips crc_4_type**Description**

Type of/page in CIPS memory for to be CRC'd, CRC Array slot 4.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_4 (187)	CRC value for the indicated buffer space, CRC Array slot 4.
cips crc_4.addr (187)	Buffer start address for CRC calculation, CRC Array slot 4.
cips crc_4.length (188)	Length of memory to be CRC'd, CRC Array slot 4.
cips crc_4.valid (189)	Validity flag for the related CRC calculation, CRC Array slot 4.

E.108 cips crc_4_valid**Description**

Validity flag for the related CRC calculation, CRC Array slot 4.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
IN_PROGRESS	0	GOOD
VALID	21845	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_4 (187)	CRC value for the indicated buffer space, CRC Array slot 4.
cips crc_4.addr (187)	Buffer start address for CRC calculation, CRC Array slot 4.
cips crc_4.type (188)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 4.
cips crc_4.length (188)	Length of memory to be CRC'd, CRC Array slot 4.

E.109 cips crc_5**Description**

CRC value for the indicated buffer space, CRC Array slot 5.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_5.valid (191)	Validity flag for the related CRC calculation, CRC Array slot 5.
cips crc_5.length (190)	Length of memory to be CRC'd, CRC Array slot 5.
cips crc_5.type (191)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 5.
cips crc_5.addr (190)	Buffer start address for CRC calculation, CRC Array slot 5.

E.110 cips crc_5_addr**Description**

Buffer start address for CRC calculation, CRC Array slot 5.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_5.valid (191)	Validity flag for the related CRC calculation, CRC Array slot 5.
cips crc_5 (189)	CRC value for the indicated buffer space, CRC Array slot 5.
cips crc_5.length (190)	Length of memory to be CRC'd, CRC Array slot 5.
cips crc_5.type (191)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 5.

E.111 cips crc_5_length**Description**

Length of memory to be CRC'd, CRC Array slot 5.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_5.valid (191)	Validity flag for the related CRC calculation, CRC Array slot 5.
cips crc_5 (189)	CRC value for the indicated buffer space, CRC Array slot 5.
cips crc_5.type (191)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 5.
cips crc_5.addr (190)	Buffer start address for CRC calculation, CRC Array slot 5.

E.112 cips crc_5_type**Description**

Type of/page in CIPS memory for to be CRC'd, CRC Array slot 5.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_5 (189)	CRC value for the indicated buffer space, CRC Array slot 5.
cips crc_5.length (190)	Length of memory to be CRC'd, CRC Array slot 5.
cips crc_5.addr (190)	Buffer start address for CRC calculation, CRC Array slot 5.
cips crc_5.valid (191)	Validity flag for the related CRC calculation, CRC Array slot 5.

E.113 cips crc_5_valid**Description**

Validity flag for the related CRC calculation, CRC Array slot 5.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
IN_PROGRESS	0	GOOD
VALID	21845	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_5 (189)	CRC value for the indicated buffer space, CRC Array slot 5.
cips crc_5.length (190)	Length of memory to be CRC'd, CRC Array slot 5.
cips crc_5.type (191)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 5.
cips crc_5.addr (190)	Buffer start address for CRC calculation, CRC Array slot 5.

E.114 cips crc_6**Description**

CRC value for the indicated buffer space, CRC Array slot 6.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_6.type (193)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 6.
cips crc_6.addr (192)	Buffer start address for CRC calculation, CRC Array slot 6.
cips crc_6.length (193)	Length of memory to be CRC'd, CRC Array slot 6.
cips crc_6.valid (194)	Validity flag for the related CRC calculation, CRC Array slot 6.

E.115 cips crc_6_addr**Description**

Buffer start address for CRC calculation, CRC Array slot 6.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_6.type (193)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 6.
cips crc_6.length (193)	Length of memory to be CRC'd, CRC Array slot 6.
cips crc_6 (192)	CRC value for the indicated buffer space, CRC Array slot 6.
cips crc_6.valid (194)	Validity flag for the related CRC calculation, CRC Array slot 6.

E.116 cips crc_6_length**Description**

Length of memory to be CRC'd, CRC Array slot 6.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_6.type (193)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 6.
cips crc_6.addr (192)	Buffer start address for CRC calculation, CRC Array slot 6.
cips crc_6 (192)	CRC value for the indicated buffer space, CRC Array slot 6.
cips crc_6.valid (194)	Validity flag for the related CRC calculation, CRC Array slot 6.

E.117 cips crc_6_type**Description**

Type of/page in CIPS memory for to be CRC'd, CRC Array slot 6.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_6.addr (192)	Buffer start address for CRC calculation, CRC Array slot 6.
cips crc_6.length (193)	Length of memory to be CRC'd, CRC Array slot 6.
cips crc_6 (192)	CRC value for the indicated buffer space, CRC Array slot 6.
cips crc_6.valid (194)	Validity flag for the related CRC calculation, CRC Array slot 6.

E.118 cips crc_6_valid**Description**

Validity flag for the related CRC calculation, CRC Array slot 6.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
IN_PROGRESS	0	GOOD
VALID	21845	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_6.type (193)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 6.
cips crc_6.addr (192)	Buffer start address for CRC calculation, CRC Array slot 6.
cips crc_6.length (193)	Length of memory to be CRC'd, CRC Array slot 6.
cips crc_6 (192)	CRC value for the indicated buffer space, CRC Array slot 6.

E.119 cips crc_7**Description**

CRC value for the indicated buffer space, CRC Array slot 7.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_7_length (195)	Length of memory to be CRC'd, CRC Array slot 7.
cips crc_7_valid (196)	Validity flag for the related CRC calculation, CRC Array slot 7.
cips crc_7_type (196)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 7.
cips crc_7_addr (195)	Buffer start address for CRC calculation, CRC Array slot 7.

E.120 cips crc_7_addr**Description**

Buffer start address for CRC calculation, CRC Array slot 7.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_7 (194)	CRC value for the indicated buffer space, CRC Array slot 7.
cips crc_7_length (195)	Length of memory to be CRC'd, CRC Array slot 7.
cips crc_7_valid (196)	Validity flag for the related CRC calculation, CRC Array slot 7.
cips crc_7_type (196)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 7.

E.121 cips crc_7_length**Description**

Length of memory to be CRC'd, CRC Array slot 7.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_7 (194)	CRC value for the indicated buffer space, CRC Array slot 7.
cips crc_7.valid (196)	Validity flag for the related CRC calculation, CRC Array slot 7.
cips crc_7.type (196)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 7.
cips crc_7.addr (195)	Buffer start address for CRC calculation, CRC Array slot 7.

E.122 cips crc_7_type**Description**

Type of/page in CIPS memory for to be CRC'd, CRC Array slot 7.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_7 (194)	CRC value for the indicated buffer space, CRC Array slot 7.
cips crc_7.length (195)	Length of memory to be CRC'd, CRC Array slot 7.
cips crc_7.addr (195)	Buffer start address for CRC calculation, CRC Array slot 7.
cips crc_7.valid (196)	Validity flag for the related CRC calculation, CRC Array slot 7.

E.123 cips crc_7_valid**Description**

Validity flag for the related CRC calculation, CRC Array slot 7.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
IN_PROGRESS	0	GOOD
VALID	21845	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_7 (194)	CRC value for the indicated buffer space, CRC Array slot 7.
cips crc_7_length (195)	Length of memory to be CRC'd, CRC Array slot 7.
cips crc_7_type (196)	Type of/page in CIPS memory for to be CRC'd, CRC Array slot 7.
cips crc_7_addr (195)	Buffer start address for CRC calculation, CRC Array slot 7.

E.124 cips crc_addr**Description**

Buffer start address for operator-initiated CRC calculation.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
init cips crc (74)	Checksum CIPS SRAM or EEPROM memory.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_type (198)	Page in CIPS memory to be CRC'd, operator-initiated CRC calculation.
cips crc (179)	CRC value for the operator-initiated CRC calculation.
cips crc_valid (198)	Validity flag for the operator-initiated CRC calculation.
cips crc_length (197)	Length of memory to be CRC'd, CRC Array slot 1.

E.125 cips crc_length**Description**

Length of memory to be CRC'd, CRC Array slot 1.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
init cips crc (74)	Checksum CIPS SRAM or EEPROM memory.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_type (198)	Page in CIPS memory to be CRC'd, operator-initiated CRC calculation.
cips crc_addr (197)	Buffer start address for operator-initiated CRC calculation.
cips crc (179)	CRC value for the operator-initiated CRC calculation.
cips crc_valid (198)	Validity flag for the operator-initiated CRC calculation.

E.126 cips_crc_type**Description**

Page in CIPS memory to be CRC'd, operator-initiated CRC calculation.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
init cips crc (74)	Checksum CIPS SRAM or EEPROM memory.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_addr (197)	Buffer start address for operator-initiated CRC calculation.
cips crc (179)	CRC value for the operator-initiated CRC calculation.
cips crc_length (197)	Length of memory to be CRC'd, CRC Array slot 1.
cips crc_valid (198)	Validity flag for the operator-initiated CRC calculation.

E.127 cips_crc_valid**Description**

Validity flag for the operator-initiated CRC calculation.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
IN_PROGRESS	0	GOOD
VALID	21845	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
init cips crc (74)	Checksum CIPS SRAM or EEPROM memory.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips crc_type (198)	Page in CIPS memory to be CRC'd, operator-initiated CRC calculation.
cips crc_addr (197)	Buffer start address for operator-initiated CRC calculation.
cips crc (179)	CRC value for the operator-initiated CRC calculation.
cips crc_length (197)	Length of memory to be CRC'd, CRC Array slot 1.

E.128 cips crc_wpc**Description**

Words used in background CRC per cycle through the software's executive loop.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.129 cips csci**Description**

Software currently being executed on CIPS.

Data Type

8 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
SRAM	85	GOOD
SuROM	170	GOOD

Telemetry Source Packets

cips evt_log (482), cips post_mortem (500)

E.130 cips_csci_version_num**Description**

Version number of the software currently being executed on CIPS.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips_csci_version_pl (200)	Patch level of the software currently being executed on CIPS.
cips_active_csci (147)	Software currently being executed on CIPS.

E.131 cips_csci_version_pl**Description**

Patch level of the software currently being executed on CIPS.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips_active_csci (147)	Software currently being executed on CIPS.
cips_csci_version_num (200)	Version number of the software currently being executed on CIPS.

E.132 cips ctml_rbuf_wds**Description**

Number of words of data currently in the critical telemetry ring buffer.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.133 cips cur_img_ct**Description**

Image number for this camera in the current SDP cycle.

Data Type

8 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Discussion

The image sequence number is reset only when SDP has completed for all images previously taken.

Telemetry Source Packets

cips sci_mx_cam (502), cips sci_my_cam (505), cips sci_px_cam (508), cips sci_py_cam (511)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips current_img (202)	Current image being processed by on-board CIPS SDP.
cips img_taken (263)	Total number of images taken since last SDP reset.
cips sdp_in_progress (435)	Flag showing SDP activity.

E.134 cips current_cam**Description**

Current camera whose image data is being processed by on-board CIPS SDP.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
PX	0	GOOD
PY	1	GOOD
MX	2	GOOD
MY	3	GOOD

Telemetry Source Packets

cips sw_status (514)

E.135 cips current_img**Description**

Current image being processed by on-board CIPS SDP.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips cur_img_ct (201)	Image number for this camera in the current SDP cycle.

E.136 cips dc_activated**Description**

CIPS camera dust cover deployment activation.

Data Type

4 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
NONE	0	GOOD
PX_CAM_PRIMARY	1	CAUTION
PY_CAM_PRIMARY	2	CAUTION
MX_CAM_PRIMARY	3	CAUTION
MY_CAM_PRIMARY	4	CAUTION
PX_CAM_SECONDARY	5	CAUTION
PY_CAM_SECONDARY	6	CAUTION
MX_CAM_SECONDARY	7	CAUTION
MY_CAM_SECONDARY	8	CAUTION

Discussion

Indicates and which camera's dust cover is currently being opened and which circuit is being used.

Telemetry Source Packets

cips hw_hk (492)

E.137 cips dc_conv_1_temp**Description**

CIPS DC-DC converter (1) temperature.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Limits

alarm	limit
red high	61
yellow high	56
yellow low	-35
red low	-40
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.138 cips dc_conv_2_temp**Description**

CIPS DC-DC converter (2) temperature.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Limits

alarm	limit
red high	61
yellow high	56
yellow low	-35
red low	-40
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.139 cips dc_timeout

Description

Dust cover deployment timeout value.

Data Type

16 bits, unsigned integer, engineering units = ms.

Measurement Source

hk fpga

Discussion

The CIPS HK FPGA will power the motor that deploys the camera's dust cover for this long.

Telemetry Source Packets

cips hw_hk (492)

E.140 cips dump_addr

Description

Memory dump address.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips mem_dump (499)

Related Commands

<i>command</i>	<i>description</i>
dump cips memory (57)	Dump CIPS memory.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips mem_dump_remain (270)	Number of words remaining to be dumped from CIPS memory.

E.141 cips dump_data

Description

cips memory dump data.

Data Type

16 bits, fill, engineering units = any.

Measurement Source

software

Telemetry Source Packets**cips mem_dump** (499)**Related Commands**

<i>command</i>	<i>description</i>
dump cips memory (57)	Dump CIPS memory.

E.142 cips dump_length**Description**

Memory dump size in 16-bit words.

Data Type

32 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets**cips mem_dump** (499)**Related Commands**

<i>command</i>	<i>description</i>
dump cips memory (57)	Dump CIPS memory.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips mem_dump_remain (270)	Number of words remaining to be dumped from CIPS memory.

E.143 cips dump_type**Description**

Memory dump type.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets**cips mem_dump** (499)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips mem_dump_remain (270)	Number of words remaining to be dumped from CIPS memory.

E.144 cips e_cami_fpga_ver**Description**

CIPS camera interface (CAMI) board FPGA version.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

processor fpga

Telemetry Source Packets

cips exception (484)

E.145 cips e_cde_uart_f_err**Description**

CDE UART receiver format error (did not get correct stop bit).

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
OK	0	GOOD
ERROR	1	GOOD

Telemetry Source Packets

cips exception (484)

E.146 cips e_cde_uart_rx_af**Description**

CDE UART receive FIFO almost full (contains 31 or more bytes).

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
OK	0	GOOD
A_FULL	1	GOOD

Telemetry Source Packets

cips exception (484)

E.147 cips e_cde_uart_rx_e**Description**

CDE UART receive FIFO empty status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
DATA_READY	0	GOOD
EMPTY	1	GOOD

Telemetry Source Packets

cips exception (484)

E.148 cips e_cde_uart_rx_f**Description**

CDE UART receive FIFO fill status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
OK	0	GOOD
FULL	1	GOOD

Telemetry Source Packets

cips exception (484)

E.149 cips e_cde_uart_rx_or**Description**

CDE UART receiver overrun error (receiver FIFO full and another byte received).

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
OK	0	GOOD
ERROR	1	GOOD

Telemetry Source Packets

cips exception (484)

E.150 cips e_cde_uart_rx_r**Description**

CDE UART receive FIFO ready status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
EMPTY	0	GOOD
DATA_READY	1	GOOD

Telemetry Source Packets

cips exception (484)

E.151 cips e_cde_uart_tx_ae

Description

CDE UART transmit FIFO almost empty (contains one byte or less) status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
OK	0	GOOD
A_EMPTY	1	GOOD

Telemetry Source Packets

cips exception (484)

E.152 cips e_cde_uart_tx_af

Description

CDE UART transmit FIFO almost full (contains 31 or more bytes) status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
OK	0	GOOD
A_FULL	1	GOOD

Telemetry Source Packets

cips exception (484)

E.153 cips e_cde_uart_tx_b

Description

CDE UART transmitter state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
NOT_BUSY	0	GOOD
BUSY	1	GOOD

Telemetry Source Packets

cips exception (484)

E.154 cips e_cde_uart_tx_e**Description**

CDE UART transmit FIFO empty status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
OK	0	GOOD
EMPTY	1	GOOD

Telemetry Source Packets

cips exception (484)

E.155 cips e_cde_uart_tx_f**Description**

CDE UART transmit FIFO full status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
OK	0	GOOD
FULL	1	GOOD

Telemetry Source Packets

cips exception (484)

E.156 cips e_hk_fpga_ver**Description**

CIPS housekeeping FPGA version.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

hk fpga

Telemetry Source Packets

cips exception (484)

E.157 cips e_proc_fpga_ver**Description**

CIPS processor FPGA version.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

processor fpga

Telemetry Source Packets

cips exception (484)

E.158 cips e_srt_brx_mc_ill**Description**

SuMMIT chip broadcast receive mode code illegalization flags.

Data Type

32 bits, unsigned integer, engineering units = dn.

Measurement Source

summit chip

Telemetry Source Packets

cips exception (484)

E.159 cips e_srt_brx_sa_ill

Description

SuMMIT chip broadcast receive subaddress illegalization flags.

Data Type

32 bits, unsigned integer, engineering units = dn.

Measurement Source

summit chip

Telemetry Source Packets

cips exception (484)

E.160 cips e_srt_btx_mc_ill

Description

SuMMIT chip broadcast transmit mode code illegalization flags.

Data Type

32 bits, unsigned integer, engineering units = dn.

Measurement Source

summit chip

Telemetry Source Packets

cips exception (484)

E.161 cips e_srt_btx_sa_ill

Description

SuMMIT chip broadcast transmit subaddress illegalization flags.

Data Type

32 bits, unsigned integer, engineering units = dn.

Measurement Source

summit chip

Telemetry Source Packets

cips exception (484)

E.162 cips e_srt_cr_bcen**Description**

SuMMIT chip broadcast enable flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
DISABLED	0	GOOD
ENABLED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.163 cips e_srt_cr_bufc**Description**

SuMMIT chip buffer mode enable flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
DISABLED	0	GOOD
ENABLED	1	GOOD

Discussion

This buffer mode is NOT the circular buffer mode, but rather the 32-word DMA 'burst' buffer mode.

Telemetry Source Packets

cips exception (484)

E.164 cips e_srt_cr_chaen**Description**

SuMMIT chip channel A enable flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
FALSE	0	GOOD
TRUE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.165 cips e_srt_cr_chben**Description**

SuMMIT chip channel B enable flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
DISABLED	0	GOOD
ENABLED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.166 cips e_srt_cr_dynbc**Description**

SuMMIT chip dynamic bus control acceptance flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
DISABLED	0	GOOD
ENABLED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.167 cips e_srt_cr_etce**Description**

SuMMIT chip external timer clock enable flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
DISABLED	0	GOOD
ENABLED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.168 cips e_srt_cr_inten**Description**

SuMMIT chip interrupt logging enable flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
DISABLED	0	GOOD
ENABLED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.169 cips e_srt_cr_ppen

Description

SuMMIT chip ping-pong mode enable flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
DISABLED	0	GOOD
ENABLED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.170 cips e_srt_cr_stex

Description

SuMMIT chip execution enable flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
DISABLED	0	GOOD
ENABLED	1	GOOD

Discussion

This bit should never be set, because if it is, no telemetry will be sent from the CIPS instrument and thus, no telemetry can be received.

Telemetry Source Packets

cips exception (484)

E.171 cips e_srt_desc_ptr**Description**

SuMMIT chip descriptor pointer.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

summit chip

Telemetry Source Packets

cips exception (484)

E.172 cips e_srt_ill_ptr**Description**

SuMMIT chip interrupt log list pointer.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

summit chip

Telemetry Source Packets

cips exception (484)

E.173 cips e_srt_rx_mc_ill**Description**

SuMMIT chip receive mode code illegalization flags.

Data Type

32 bits, unsigned integer, engineering units = dn.

Measurement Source

summit chip

Telemetry Source Packets

cips exception (484)

E.174 cips e_srt_rx_sa_ill**Description**

SuMMIT chip receive subaddress illegalization flags.

Data Type

32 bits, unsigned integer, engineering units = dn.

Measurement Source

summit chip

Telemetry Source Packets

cips exception (484)

E.175 cips e_srt_tx_mc_ill**Description**

SuMMIT chip transmit mode code illegalization flags.

Data Type

32 bits, unsigned integer, engineering units = dn.

Measurement Source

summit chip

Telemetry Source Packets

cips exception (484)

E.176 cips e_srt_tx_sa_ill**Description**

SuMMIT chip transmit subaddress illegalization flags.

Data Type

32 bits, unsigned integer, engineering units = dn.

Measurement Source

summit chip

Telemetry Source Packets

cips exception (484)

E.177 cips ee_write_enable**Description**

EEPROM write enable.

Data Type

16 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
FALSE	0	GOOD
TRUE	1	CAUTION

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
disable cips ee_write (50)	Disable writes to CIPS EEPROM.
enable cips ee_write (62)	Enable writes to CIPS EEPROM.

E.178 cips ee_write_ipd**Description**

CIPS EEPROM write inter-page delay.

Data Type

16 bits, unsigned integer, engineering units = ms.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.179 cips eeprom_reset**Description**

EEPROM held in reset flag.

Data Type

3 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
NOT_IN_RESET	0	GOOD
RESET	7	CAUTION

Discussion

CIPS FSW enables this hold of EEPROM in a reset condition when in the SAFE mode. EEPROM should be held in reset during a power cycle to prevent possible EEPROM corruption. While in reset, EEPROM cannot be read or written to.

Telemetry Source Packets

cips hw_hk (492)

E.180 cips em_acs_stat_rx**Description**

ACS status command received log event mask status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
OFF	0	GOOD
ON	1	CAUTION

Discussion

Indicates whether or not acs status command received log messages are written to the log history packet.

Note that since this command is sent from the main spacecraft bus computer to CIPS at a rate of 1 Hz, if this event mask is active, the log message packet will fill and be passed to the ground much more rapidly than the nominal rate.

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
enable cips em (63)	Enable given messages in the event log packet.
disable cips em (51)	Disable given messages in the event log packet.
set cips em_mask (99)	Set the software event logging mask.

E.181 cips em_cde_cmd_tx

Description

CDE command sent log event mask status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
OFF	0	CAUTION
ON	1	GOOD

Discussion

Indicates whether or not CDE command sent messages are written to the log history packet.

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
enable cips em (63)	Enable given messages in the event log packet.
disable cips em (51)	Disable given messages in the event log packet.
set cips em_mask (99)	Set the software event logging mask.

E.182 cips em_cde_tlm_rx

Description

CDE telemetry received log event mask status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
OFF	0	CAUTION
ON	1	GOOD

Discussion

Indicates whether or not CDE telemetry received messages are written to the log history packet.

Since the CDE instrument puts out telemetry packets at 1 Hz, enabling these event messages will significantly increase CIPS telemetry data output.

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
enable cips em (63)	Enable given messages in the event log packet.
disable cips em (51)	Disable given messages in the event log packet.
set cips em_mask (99)	Set the software event logging mask.

E.183 cips em_cmd_err

Description

Log event mask command error status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
OFF	0	CAUTION
ON	1	GOOD

Discussion

Indicates whether or not software command error messages are written to the log history packet.

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
enable cips em (63)	Enable given messages in the event log packet.
disable cips em (51)	Disable given messages in the event log packet.
set cips em_mask (99)	Set the software event logging mask.

E.184 cips em_cmd_rx

Description

Command received log event mask status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
OFF	0	CAUTION
ON	1	GOOD

Discussion

Indicates whether or not command received messages are written to the log history packet.

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
enable cips em (63)	Enable given messages in the event log packet.
disable cips em (51)	Disable given messages in the event log packet.
set cips em_mask (99)	Set the software event logging mask.

E.185 cips em_debug**Description**

Debug message log event mask status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
OFF	0	GOOD
ON	1	CAUTION

Discussion

Indicates whether or not debugging messages are written to the log packet.

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
enable cips em (63)	Enable given messages in the event log packet.
disable cips em (51)	Disable given messages in the event log packet.
set cips em_mask (99)	Set the software event logging mask.

E.186 cips em_err

Description

CIPS generic error log event mask status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
OFF	0	CAUTION
ON	1	GOOD

Discussion

Indicates whether or not CIPS generic error messages are written to the log history packet.

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
enable cips em (63)	Enable given messages in the event log packet.
disable cips em (51)	Disable given messages in the event log packet.
set cips em_mask (99)	Set the software event logging mask.

E.187 cips em_img_int

Description

Event mark for receipt of image acquisition complete interrupts.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
OFF	0	CAUTION
ON	1	GOOD

Telemetry Source Packets

cips sw_status (514)

E.188 cips em_img_trigger

Description

Event mark for trigger of science data acquisition (camera 'go' command).

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
OFF	0	CAUTION
ON	1	GOOD

Telemetry Source Packets

cips sw_status (514)

E.189 cips em_info

Description

Noteworthy non-error information log event mask status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
OFF	0	CAUTION
ON	1	GOOD

Discussion

Indicates whether or not non-error information messages are written to the log history packet.

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
enable cips em (63)	Enable given messages in the event log packet.
disable cips em (51)	Disable given messages in the event log packet.
set cips em_mask (99)	Set the software event logging mask.

E.190 cips em_mode_chng

Description

CIPS operational mode change log event mask status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
OFF	0	CAUTION
ON	1	GOOD

Discussion

Indicates whether or not CIPS operational mode change messages are written to the log history packet.

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
enable cips em (63)	Enable given messages in the event log packet.
disable cips em (51)	Disable given messages in the event log packet.
set cips em_mask (99)	Set the software event logging mask.

E.191 cips em_pwr

Description

Instrument power change event log event mask status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
OFF	0	CAUTION
ON	1	GOOD

Discussion

Indicates whether or not instrument power change event log messages are written to the log history packet.

Telemetry Source Packets**cips sw_status** (514)**Related Commands**

<i>command</i>	<i>description</i>
enable cips em (63)	Enable given messages in the event log packet.
disable cips em (51)	Disable given messages in the event log packet.
set cips em_mask (99)	Set the software event logging mask.

E.192 cips em_sdp_start**Description**

Start science data processing event mark.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
OFF	0	CAUTION
ON	1	GOOD

Discussion

Indicates when CIPS FSW begins on-board processing of science data.

Telemetry Source Packets**cips sw_status** (514)**E.193 cips em_sdp_stop****Description**

Stop science data processing event mark.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
OFF	0	CAUTION
ON	1	GOOD

Discussion

Indicates when CIPS FSW completes on-board processing of science data.

Telemetry Source Packets

cips sw_status (514)

E.194 cips em_seq_cmd_rx**Description**

Command sequence command receipt log mask status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
OFF	0	CAUTION
ON	1	GOOD

Discussion

Indicates whether or not event messages generated when a command is processed from an on-board CIPS command sequence are written to the log history packet.

Telemetry Source Packets

cips sw_status (514)

E.195 cips em_seq_err**Description**

Command sequence error log mask status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
OFF	0	CAUTION
ON	1	GOOD

Discussion

Indicates whether or not command sequence error messages are written to the log history packet.

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
enable cips em (63)	Enable given messages in the event log packet.
disable cips em (51)	Disable given messages in the event log packet.
set cips em_mask (99)	Set the software event logging mask.

E.196 cips em_seq_evt**Description**

Command sequence event log mask status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
OFF	0	CAUTION
ON	1	GOOD

Discussion

Indicates whether or not command sequence event messages are written to the log history packet.

Telemetry Source Packets

cips sw_status (514)

E.197 cips em_seq_mode_chng**Description**

CIPS sequence mode change log event mask status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
OFF	0	CAUTION
ON	1	GOOD

Discussion

Indicates whether or not CIPS sequence mode change messages are written to the log history packet.

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
enable cips em (63)	Enable given messages in the event log packet.
disable cips em (51)	Disable given messages in the event log packet.
set cips em_mask (99)	Set the software event logging mask.

E.198 cips em_time_rx**Description**

Time value command received log event mask status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
OFF	0	GOOD
ON	1	CAUTION

Discussion

Indicates whether or not time value command received log messages are written to the log history packet.

Note that since this command is sent from the main spacecraft bus computer to CIPS at a rate of 1 Hz, if this event mask is active, the log message packet will fill and be passed to the ground much more rapidly than the nominal rate.

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
enable cips em (63)	Enable given messages in the event log packet.
disable cips em (51)	Disable given messages in the event log packet.
set cips em_mask (99)	Set the software event logging mask.

E.199 cips em_tlm_err

Description

Telemetry error log event mask status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
OFF	0	CAUTION
ON	1	GOOD

Discussion

Indicates whether or not software telemetry error messages are written to the log history packet.

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
enable cips em (63)	Enable given messages in the event log packet.
disable cips em (51)	Disable given messages in the event log packet.
set cips em_mask (99)	Set the software event logging mask.

E.200 cips em_tlm_tx

Description

Telemetry sent log event mask status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
OFF	0	GOOD
ON	1	CAUTION

Discussion

Indicates whether or not software telemetry sent messages are written to the log history packet.

Telemetry Source Packets**cips sw_status** (514)**Related Commands**

<i>command</i>	<i>description</i>
enable cips em (63)	Enable given messages in the event log packet.
disable cips em (51)	Disable given messages in the event log packet.
set cips em_mask (99)	Set the software event logging mask.

E.201 cips em_u_test**Description**

Unit test log event mask status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
OFF	0	GOOD
ON	1	CAUTION

Discussion

Indicates whether or not unit test messages are written to the log packet.

Telemetry Source Packets**cips sw_status** (514)**Related Commands**

<i>command</i>	<i>description</i>
enable cips em (63)	Enable given messages in the event log packet.
disable cips em (51)	Disable given messages in the event log packet.
set cips em_mask (99)	Set the software event logging mask.

E.202 cips em_warn_msg**Description**

Generic WARN log event mask status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
OFF	0	CAUTION
ON	1	GOOD

Discussion

Indicates whether or not generic WARN messages are written to the log packet.

Telemetry Source Packets

cips_sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
enable cips_em (63)	Enable given messages in the event log packet.
disable cips_em (51)	Disable given messages in the event log packet.
set cips_em_mask (99)	Set the software event logging mask.

E.203 cips_err_to**Description**

State of downlink enable for CIPS error telemetry packets.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
DISABLED	0	CAUTION
ENABLED	1	GOOD

Telemetry Source Packets

cips_sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips_to_mask (110)	Set the CIPS telemetry output enable mask.
disable cips_tlm_out (56)	Disable downlink of the given CIPS telemetry packet.
enable cips_tlm_out (70)	Enable downlink of the given CIPS telemetry packet.

E.204 cips_event_ct**Description**

Number of events in this event log packet.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips_evt_log (482)

E.205 cips_event_info**Description**

CIPS event log subpacket structure.

Data Type

0 bits, subpacket, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips_evt_log (482)

E.206 cips_event_key**Description**

Event type.

Data Type

32 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
INFO	1	GOOD
POWER	2	GOOD
CMD_RCVD	4	GOOD
TLM_SENT	8	GOOD
ACS_RCVD	16	GOOD
TIME_RCVD	32	GOOD
CDE_TLM_RCVD	64	GOOD
CDE_CMD_SENT	128	GOOD
SEQ_CMD_RCVD	256	GOOD
SDP_STOP	4096	GOOD
SDP_START	8192	GOOD
IMG_INT	16384	GOOD
IMG_TRIGGER	32768	GOOD
MODE_CHNG	65536	GOOD
SEQ_MODE_CHNG	131072	GOOD
WARNING	2097152	CAUTION
SEQ_EVENT	16777216	GOOD
SEQ_ERROR	33554432	BAD
CMD_ERROR	67108864	BAD
TLM_ERROR	134217728	BAD
ERROR	268435456	BAD
DEBUG	1073741824	CAUTION

Telemetry Source Packets

cips evt_log (482)

E.207 cips event_p0**Description**

Event parameter 0.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips evt_log (482)

E.208 cips event_p1**Description**

Event parameter 1.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips evt_log (482)

E.209 cips event_p2**Description**

Event parameter 2.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips evt_log (482)

E.210 cips event_p3**Description**

Event parameter 3.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips evt_log (482)

E.211 cips event_pc**Description**

Software program counter value at event occurrence.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips evt_log (482)

E.212 cips event_time_s**Description**

Event time.

Data Type

32 bits, unsigned integer, engineering units = sec.

Measurement Source

software

Telemetry Source Packets

cips evt_log (482)

E.213 cips event_time_ss**Description**

Event time.

Data Type

32 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips evt_log (482)

E.214 cips evt_log_to**Description**

State of downlink enable for CIPS event log telemetry packets.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
DISABLED	0	CAUTION
ENABLED	1	GOOD

Telemetry Source Packets

cips_sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips to_mask (110)	Set the CIPS telemetry output enable mask.
disable cips_tlm_out (56)	Disable downlink of the given CIPS telemetry packet.
enable cips_tlm_out (70)	Enable downlink of the given CIPS telemetry packet.

E.215 cips_exc_enable**Description**

CIPS FSW exception buffer write enable.

Data Type

16 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
FALSE	0	CAUTION
TRUE	1	GOOD

Telemetry Source Packets

cips_sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
enable cips_exc_pm (64)	Re-enable CIPS exception and post-mortem data recording.

E.216 cips_exception_to**Description**

State of downlink enable for CIPS exception telemetry packets.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
DISABLED	0	CAUTION
ENABLED	1	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips to_mask (110)	Set the CIPS telemetry output enable mask.
disable cips tlm_out (56)	Disable downlink of the given CIPS telemetry packet.
enable cips tlm_out (70)	Enable downlink of the given CIPS telemetry packet.

E.217 cips fast_clears**Description**

CIPS camera fast clear count.

Data Type

3 bits, unsigned integer, engineering units = dn.

Measurement Source

camera fpga

Discussion

Each fast clear takes 586 milliseconds.

Telemetry Source Packets

cips sci_mx_cam (502), **cips sci_my_cam** (505), **cips sci_px_cam** (508), **cips sci_py_cam** (511)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips img_timestamp_s (263)	Time this image was taken, seconds field.
cips img_timestamp_ss (264)	Time this image was taken, subseconds field.

E.218 cips frame_sync_time**Description**

CIPS camera frame sync (delay) time.

Data Type

16 bits, unsigned integer, engineering units = ms.

Measurement Source

camera fpga

Telemetry Source Packets

cips sci_mx_cam (502), cips sci_my_cam (505), cips sci_px_cam (508), cips sci_py_cam (511)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips img_timestamp_s (263)	Time this image was taken, seconds field.
cips img_timestamp_ss (264)	Time this image was taken, subseconds field.

E.219 cips gndv_mon**Description**

CIPS ground voltage monitor.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	0.0001220703125
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Limits

alarm	limit
red high	0.00999999977648258
yellow high	0.00499999988824129
yellow low	0
red low	0
delta	none

Telemetry Source Packets

cips hw_hk (492)

E.220 cips hk_counter**Description**

Housekeeping FPGA 1 Hz counter.

Data Type

8 bits, unsigned integer, engineering units = dn.

Measurement Source

hk fpga

Telemetry Source Packets

cips hw_hk (492)

E.221 cips hk_fpga_ver**Description**

CIPS housekeeping FPGA version.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

hk fpga

Telemetry Source Packets

cips hw_hk (492)

E.222 cips hk_irq_1s**Description**

Housekeeping FPGA one hertz interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.223 cips hk_irq_1s_mk**Description**

Housekeeping FPGA one hertz interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.224 cips hk_irq_lpevt**Description**

Housekeeping FPGA ADC latchup event interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.225 cips hk_irq_lpevt_mk**Description**

Housekeeping FPGA ADC latchup event interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.226 cips hk_irq_lppwr**Description**

Housekeeping FPGA ADC latchup powerdown interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.227 cips hk_irq_lppwr_mk**Description**

Housekeeping FPGA ADC latchup powerdown interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.228 cips hk_irq_mx_bos**Description**

Housekeeping FPGA -X camera bright object sensor interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.229 cips hk_irq_mx_bos_mk**Description**

Housekeeping FPGA -X camera bright object sensor interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets**cips exception** (484)**E.230 cips hk_irq_my_bos****Description**

Housekeeping FPGA -Y camera bright object sensor interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets**cips exception** (484)**E.231 cips hk_irq_my_bos_mk****Description**

Housekeeping FPGA -Y camera bright object sensor interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets**cips exception** (484)

E.232 cips hk_irq_px_bos**Description**

Housekeeping FPGA +X camera bright object sensor interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.233 cips hk_irq_px_bos_mk**Description**

Housekeeping FPGA +X camera bright object sensor interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.234 cips hk_irq_py_bos**Description**

Housekeeping FPGA +Y camera bright object sensor interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.235 cips hk_irq_py_bos_mk**Description**

Housekeeping FPGA +Y camera bright object sensor interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.236 cips hk_irq_sdrami**Description**

Housekeeping FPGA SDRAM overcurrent interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.237 cips hk_irq_sdrami_mk**Description**

Housekeeping FPGA SDRAM overcurrent interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.238 cips hk_lpbit_ena**Description**

Housekeeping ADC latchup bit test enable.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
FALSE	0	GOOD
TRUE	1	CAUTION

Telemetry Source Packets

cips hw_hk (492)

E.239 cips hk_mux_1_addr**Description**

Housekeeping ADC multiplexor address selection.

Data Type

3 bits, unsigned integer, engineering units = dn.

Measurement Source

hk fpga

Discussion

This value is only valid when the ADC is being tested, as shown in the test bit (see related telemetry)

Telemetry Source Packets

cips hw_hk (492)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips hk_mux_test_ena (251)	Housekeeping ADC multiplexor test enable.

E.240 cips hk_mux_2_addr**Description**

Housekeeping ADC multiplexor address selection.

Data Type

3 bits, unsigned integer, engineering units = dn.

Measurement Source

hk fpga

Discussion

This value is only valid when the ADC is being tested, as shown in the test bit (see related telemetry)

Telemetry Source Packets

cips hw_hk (492)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips hk_mux_test_ena (251)	Housekeeping ADC multiplexor test enable.

E.241 cips hk_mux_ena**Description**

Housekeeping ADC multiplexor selection.

Data Type

7 bits, unsigned integer, engineering units = dn.

Measurement Source

hk fpga

Discussion

This value is only valid when the ADC is being tested, as shown in the test bit (see related telemetry)

Telemetry Source Packets

cips hw_hk (492)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips hk_mux_test_ena (251)	Housekeeping ADC multiplexor test enable.

E.242 cips hk_mux_test_ena**Description**

Housekeeping ADC multiplexor test enable.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
FALSE	0	GOOD
TRUE	1	CAUTION

Telemetry Source Packets

cips hw_hk (492)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips hk_mux_2_addr (250)	Housekeeping ADC multiplexor address selection.
cips hk_mux_1_addr (250)	Housekeeping ADC multiplexor address selection.
cips hk_mux_ena (250)	Housekeeping ADC multiplexor selection.

E.243 cips hk_pcb_temp**Description**

CIPS housekeeping board temperature.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Limits

alarm	limit
red high	52
yellow high	47
yellow low	-35
red low	-40
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.244 cips hw_hk_rt**Description**

Packet production period for CIPS hardware housekeeping packets.

Data Type

16 bits, unsigned integer, engineering units = sec.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips hw_hk_rt (101)	Set the cips housekeeping telemetry packet period.

E.245 cips hw_hk_to

Description

State of downlink enable for CIPS hardware housekeeping telemetry packets.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
DISABLED	0	CAUTION
ENABLED	1	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips to_mask (110)	Set the CIPS telemetry output enable mask.
disable cips tlm_out (56)	Disable downlink of the given CIPS telemetry packet.
enable cips tlm_out (70)	Enable downlink of the given CIPS telemetry packet.

E.246 cips i_cs_thresh

Description

Initial (boot) value for CIPS SDRAM current sense threshold.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.247 cips i_l_htr_lthresh

Description

Initial (boot) value for the CIPS cameras' lens heater lower threshold.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

software

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Discussion

The camera lens heaters are controlled based on this threshold. When the lens temperature (lens thermistor 'A' is used to make this determination) is below this value, the HK FPGA turns the lens heater on. This applies individually to all cameras (i.e. if one camera lens is below threshold, only that camera's lens heater is turned on). This does not apply when the lens heaters are disabled (see related telemetry), in which case lens heaters are always turned off

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips my_lens_htr_ena (324)	CIPS -Y camera lens heater power enable.
cips px_lens_htr_ena (403)	CIPS +X camera lens heater power enable.
cips mx_lens_htr_ena (305)	CIPS -X camera lens heater power enable.
cips py_lens_htr_ena (423)	CIPS +Y camera lens heater power enable.

E.248 cips i_l_htr_uthresh**Description**

Initial (boot) value for the CIPS cameras' lens heater upper threshold.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

software

Analog Conversions

Segment 1

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Discussion

The camera lens heaters are controlled based on this threshold. When the lens temperature (lens thermistor 'A' is used to make this determination) is below this value, the HK FPGA turns the lens heater on. This applies individually to all cameras (i.e. if one camera lens is below threshold, only that camera's lens heater is turned on). This does not apply when the lens heaters are disabled (see related telemetry), in which case lens heaters are always turned off

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips my_lens_htr_ena (324)	CIPS -Y camera lens heater power enable.
cips px_lens_htr_ena (403)	CIPS +X camera lens heater power enable.
cips mx_lens_htr_ena (305)	CIPS -X camera lens heater power enable.
cips py_lens_htr_ena (423)	CIPS +Y camera lens heater power enable.

E.249 cips i_mx_bos_level**Description**

Initial (boot) value for CIPS -X camera bright object sensor level.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

software

Analog Conversions

Segment 1

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	-0.00030517578125
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Telemetry Source Packets

cips sw_status (514)

E.250 cips i_my_bos_level**Description**

Initial (boot) value for CIPS -Y camera bright object sensor level.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

software

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	-0.00030517578125
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Telemetry Source Packets

cips sw_status (514)

E.251 cips i_px_bos_level

Description

Initial (boot) value for CIPS +X camera bright object sensor level.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

software

Analog Conversions

Segment 1

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	-0.00030517578125
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Telemetry Source Packets

cips sw_status (514)

E.252 cips i_py_bos_level

Description

Initial (boot) value for CIPS +Y camera bright object sensor level.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

software

Analog Conversions

Segment 1

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	-0.00030517578125
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Telemetry Source Packets

cips sw_status (514)

E.253 cips img_cols**Description**

Number of columns in the image.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sci_mx_cam (502), cips sci_my_cam (505), cips sci_px_cam (508), cips sci_py_cam (511)

E.254 cips img_cols_ccd**Description**

Columns in each image in the current observation when read from the CCD.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips img_size (104)	Set the raw CCD image size for all CIPS cameras.

E.255 cips img_cols_sdram**Description**

Columns in each image in the current observation when read from SDRAM.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Discussion

, Calculated by CIPS FSW based on raw image size and binning options.

Telemetry Source Packets

cips sw_status (514)

E.256 cips img_cols_tlm**Description**

Columns in each image as sent in telemetry.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Discussion

Calculated by CIPS FSW based on raw image size and binning options.

Telemetry Source Packets

cips sw_status (514)

E.257 cips img_delay_tm**Description**

Time between images in the current observation.

Data Type

16 bits, unsigned integer, engineering units = sec.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips img_delay (102)	Set the time interval between CIPS images during the next observation.

E.258 cips img_mode**Description**

Camera FPGA imaging mode for data in this science packet.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

camera fpga

State Conversions

state	value	desirability
SNAPSHOT	0	GOOD
TDI	1	GOOD

Telemetry Source Packets

cips sci_mx_cam (502), cips sci_my_cam (505), cips sci_px_cam (508), cips sci_py_cam (511)

E.259 cips img_offset_col**Description**

CIPS camera image column start offset from CCD origin.

Data Type

12 bits, unsigned integer, engineering units = dn.

Measurement Source

camera fpga

Telemetry Source Packets

cips sci_mx_cam (502), cips sci_my_cam (505), cips sci_px_cam (508), cips sci_py_cam (511)

Related Commands

<i>command</i>	<i>description</i>
set cips img_offset (103)	Set the raw CCD image origin offset from the top and side of the CCD.

E.260 cips img_offset_row**Description**

CIPS camera image row start offset from CCD origin.

Data Type

12 bits, unsigned integer, engineering units = dn.

Measurement Source

camera fpga

Telemetry Source Packets

cips sci_mx_cam (502), **cips sci_my_cam (505)**, **cips sci_px_cam (508)**, **cips sci_py_cam (511)**

Related Commands

<i>command</i>	<i>description</i>
set cips img_offset (103)	Set the raw CCD image origin offset from the top and side of the CCD.

E.261 cips img_remaining**Description**

Number of images remaining to be taken in the current observation.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.262 cips img_rows**Description**

Number of rows in the image.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sci_mx_cam (502), cips sci_my_cam (505), cips sci_px_cam (508), cips sci_py_cam (511)

E.263 cips img_rows_ccd**Description**

Rows in each image in the current observation when read from the CCD.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips img_size (104)	Set the raw CCD image size for all CIPS cameras.

E.264 cips img_rows_sdram**Description**

Number of rows in an image as stored in SDRAM.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Discussion

Calculated by CIPS FSW based on raw image size and binning options.

Telemetry Source Packets

cips sw_status (514)

E.265 cips_img_rows_tlm**Description**

Number of rows in an image as sent in telemetry.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Discussion

Calculated by CIPS FSW based on raw image size and binning options.

Telemetry Source Packets

cips_sw_status (514)

E.266 cips_img_taken**Description**

Total number of images taken since last SDP reset.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Discussion

This value will max out at 50, after which no more images can be taken until science data processing is completed on the existing images.

Telemetry Source Packets

cips_sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips_cur_img_ct (201)	Image number for this camera in the current SDP cycle.

E.267 cips_img_timestamp_s**Description**

Time this image was taken, seconds field.

Data Type

32 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Discussion

Image timestamps are recorded by CIPS FSW when the image acquisition process is triggered. Data are actually recorded by the CIPS cameras a measurable time after this timestamp. The CCD integration period starts S milliseconds later, based on the camera's clear count setting (C) and frame sync (or delay) time (F) where $S = 586 * C + F$. The integration stops at the time indicated in the 'cips shutter.cl.time' telemetry point.

Telemetry Source Packets

cips_sci_mx_cam (502), cips_sci_my_cam (505), cips_sci_px_cam (508), cips_sci_py_cam (511)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips_shutter.cl.time (440)	CIPS camera shutter close time.
cips_frame_sync.time (240)	CIPS camera frame sync (delay) time.
cips_fast_clears (240)	CIPS camera fast clear count.

E.268 cips_img_timestamp_ss**Description**

Time this image was taken, subseconds field.

Data Type

32 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Discussion

Image timestamps are recorded by CIPS FSW when the image acquisition process is triggered. Data are actually recorded by the CIPS cameras a measurable time after this timestamp. The CCD integration period starts S milliseconds later, based on the camera's clear count setting (C) and frame sync (or delay) time (F) where $S = 586 * C + F$. The integration stops at the time indicated in the 'cips shutter.cl.time' telemetry point.

Telemetry Source Packets

cips_sci_mx_cam (502), cips_sci_my_cam (505), cips_sci_px_cam (508), cips_sci_py_cam (511)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips_shutter.cl.time (440)	CIPS camera shutter close time.
cips_frame_sync.time (240)	CIPS camera frame sync (delay) time.
cips_fast_clears (240)	CIPS camera fast clear count.

E.269 cips img_to_be_taken

Description

Number of images to be taken in the current observation.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
init cips imaging (75)	Immediately begin taking images with the indicated CIPS CCDs.

E.270 cips img_with_mx_cam

Description

Flag showing whether the -X camera is to be used in the current observation.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
FALSE	0	GOOD
TRUE	1	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
init cips imaging (75)	Immediately begin taking images with the indicated CIPS CCDs.

E.271 cips img_with_my_cam

Description

Flag showing whether the -Y camera is to be used in the current observation.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
FALSE	0	GOOD
TRUE	1	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
init cips imaging (75)	Immediately begin taking images with the indicated CIPS CCDs.

E.272 cips img_with_px_cam**Description**

Flag showing whether the +X camera is to be used in the current observation.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
FALSE	0	GOOD
TRUE	1	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
init cips imaging (75)	Immediately begin taking images with the indicated CIPS CCDs.

E.273 cips img_with_py_cam**Description**

Flag showing whether the +Y camera is to be used in the current observation.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
FALSE	0	GOOD
TRUE	1	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
init cips imaging (75)	Immediately begin taking images with the indicated CIPS CCDs.

E.274 cips lens_htr_lthresh**Description**

Lens heater lower threshold.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Discussion

The camera lens heaters are controlled based on this threshold. When the lens temperature (lens thermistor 'A' is used to make this determination) is below this value, the HK FPGA turns the lens heater on. This applies individually to all cameras (i.e. if one camera lens is below threshold, only that camera's lens heater is turned on). This does not apply when the lens heaters are disabled (see related telemetry), in which case lens heaters are always turned off

Telemetry Source Packets**cips hw_hk** (492)**Related Telemetry**

<i>telemetry</i>	<i>description</i>
cips my_lens_htr_ena (324)	CIPS -Y camera lens heater power enable.
cips px_lens_htr_ena (403)	CIPS +X camera lens heater power enable.
cips mx_lens_htr_ena (305)	CIPS -X camera lens heater power enable.
cips py_lens_htr_ena (423)	CIPS +Y camera lens heater power enable.

E.275 cips lens_htr_uthresh**Description**

Lens heater upper threshold.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Discussion

The camera lens heaters are controlled based on this threshold. When the lens temperature (lens thermistor 'A' is used to make this determination) is above this value, the HK FPGA turns the lens heater off. This applies individually to all cameras (i.e. if one camera lens is above threshold, only that camera's lens heater is turned off). This does not apply when the lens heaters are disabled (see related telemetry), in which case lens heaters are always turned off.

Telemetry Source Packets**cips hw_hk** (492)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips my_lens_htr_ena (324)	CIPS -Y camera lens heater power enable.
cips px_lens_htr_ena (403)	CIPS +X camera lens heater power enable.
cips mx_lens_htr_ena (305)	CIPS -X camera lens heater power enable.
cips py_lens_htr_ena (423)	CIPS +Y camera lens heater power enable.

E.276 cips m15i_mon**Description**

CIPS -15V current monitor.

Data Type

16 bits, unsigned integer, engineering units = a.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	0.0001220703125
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Limits

alarm	limit
red high	0.649999976158142
yellow high	0.550000011920929
yellow low	-1
red low	-1
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.277 cips m15v_mon**Description**

CIPS -15 volt monitor.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-22.25
c1	0.002716064453125
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Limits

alarm	limit
red high	-14.25
yellow high	-14.5
yellow low	-15.5
red low	-15.75
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.278 cips mem_dump_remain**Description**

Number of words remaining to be dumped from CIPS memory.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets**cips sw_status** (514)**Related Commands**

<i>command</i>	<i>description</i>
dump cips memory (57)	Dump CIPS memory.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips dump_length (206)	Memory dump size in 16-bit words.
cips dump_addr (205)	Memory dump address.
cips dump_type (206)	Memory dump type.

E.279 cips mem_dump_to**Description**

State of downlink enable for CIPS memory dump telemetry packets.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
DISABLED	0	CAUTION
ENABLED	1	GOOD

Telemetry Source Packets**cips sw_status** (514)**Related Commands**

<i>command</i>	<i>description</i>
set cips to_mask (110)	Set the CIPS telemetry output enable mask.
disable cips tlm_out (56)	Disable downlink of the given CIPS telemetry packet.
enable cips tlm_out (70)	Enable downlink of the given CIPS telemetry packet.

E.280 cips mode**Description**

Software mode the instrument is currently in.

Data Type

16 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
SuROM	0	CAUTION
SRAM_INIT	1	CAUTION
SAFE	2	CAUTION
STANDBY	3	GOOD
IMAGING	4	GOOD

Discussion

Imaging and standby are the nominal modes.

Telemetry Source Packets**cips sw_status** (514)**Related Commands**

<i>command</i>	<i>description</i>
init cips sram_code (78)	Command the CIPS SuROM to switch to the SRAM code.
init cips imaging (75)	Immediately begin taking images with the indicated CIPS CCDs.
reset cips cpa (90)	Reset the CIPS Processor Assembly.
safe cips (91)	Place the cips instrument in a safe state.

E.281 cips ms1553_ac_a_s_ct**Description**

Mil-Std-1553 Bus-A message success counter, ACS status channel.

Data Type

32 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets**cips sw_status** (514)**E.282 cips ms1553_ac_b_s_ct****Description**

Mil-Std-1553 Bus-B message success counter, ACS status channel.

Data Type

32 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.283 cips ms1553_b_a_ce_ct**Description**

Mil-Std-1553 bus A illegal command error counter.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.284 cips ms1553_b_a_me_ct**Description**

Mil-Std-1553 bus A manchester error counter.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.285 cips ms1553_b_a_oe_ct**Description**

Mil-Std-1553 bus A overrun error counter.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.286 cips ms1553_b_a_pe_ct**Description**

Mil-Std-1553 bus A parity error counter.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.287 cips ms1553_b_a_te_ct**Description**

Mil-Std-1553 bus A timeout error counter.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.288 cips ms1553_b_b_ce_ct**Description**

Mil-Std-1553 bus B illegal command error counter.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.289 cips ms1553_b_b_me_ct**Description**

Mil-Std-1553 bus B manchester error counter.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.290 cips ms1553_b_b_oe_ct**Description**

Mil-Std-1553 bus B overrun error counter.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.291 cips ms1553_b_b_pe_ct**Description**

Mil-Std-1553 bus B parity error counter.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.292 cips ms1553_b_b_te_ct**Description**

Mil-Std-1553 bus B timeout error counter.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.293 cips ms1553_bit_f_ct**Description**

Mil-Std-1553 Built in test fail counter.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.294 cips ms1553_bit_f_wd**Description**

Mil-Std-1553 Built in last test fail word.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.295 cips ms1553_bit_s_ct**Description**

Mil-Std-1553 Built in test success counter.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.296 cips ms1553_c1_a_s_ct**Description**

Mil-Std-1553 Bus-A message success counter, command channel 1.

Data Type

32 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.297 cips ms1553_c1_b_s_ct**Description**

Mil-Std-1553 Bus-B message success counter, command channel 1.

Data Type

32 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.298 cips ms1553_c2_a_s_ct**Description**

Mil-Std-1553 Bus-A message success counter, command channel 2.

Data Type

32 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.299 cips ms1553_c2_b_s_ct**Description**

Mil-Std-1553 Bus-B message success counter, command channel 2.

Data Type

32 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.300 cips ms1553_c_b_o_ct**Description**

Mil-Std-1553 buffer overrun count (in words), command buffer.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.301 cips ms1553_c_c_p_ct**Description**

Mil-Std-1553 corrupt CCSDS packet counter, command buffer.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.302 cips ms1553_ct_a_s_ct**Description**

Mil-Std-1553 Bus-A message success counter, critical telemetry channel.

Data Type

32 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.303 cips ms1553_ct_b_o_ct**Description**

Mil-Std-1553 buffer overrun count (in words), critical telemetry buffer.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.304 cips ms1553_ct_b_s_ct**Description**

Mil-Std-1553 Bus-B message success counter, critical telemetry channel.

Data Type

32 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.305 cips ms1553_ct_c_p_ct**Description**

Mil-Std-1553 corrupt CCSDS packet counter, critical telemetry buffer.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.306 cips ms1553_ct_wd_ct**Description**

Mil-Std-1553 critical telemetry poll watchdog trip counter.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.307 cips ms1553_db_e_ct**Description**

Mil-Std-1553 descriptor block error count.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.308 cips ms1553_ei_ct

Description

Mil-Std-1553 interrupt error ('You Failed') counter.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.309 cips ms1553_hw_r_ct

Description

Mil-Std-1553 Hardware reset counter.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.310 cips ms1553_iiw_e_ct**Description**

Mil-Std-1553 interrupt information word error counter.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.311 cips ms1553_iiw_e_wd**Description**

Mil-Std-1553 interrupt information word value for last error.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.312 cips ms1553_nt_a_s_ct**Description**

Mil-Std-1553 Bus-A message success counter, telemetry channel.

Data Type

32 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.313 cips ms1553_nt_b_o_ct**Description**

Mil-Std-1553 buffer overrun count (in words), telemetry buffer.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.314 cips ms1553_nt_b_s_ct**Description**

Mil-Std-1553 Bus-B message success counter, telemetry channel.

Data Type

32 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets**cips sw_status** (514)**E.315 cips ms1553_nt_c_p_ct****Description**

Mil-Std-1553 corrupt CCSDS packet counter, telemetry buffer.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets**cips sw_status** (514)**E.316 cips ms1553_nt_wd_ct****Description**

Mil-Std-1553 telemetry poll watchdog trip counter.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets**cips sw_status** (514)

E.317 cips ms1553_to_thresh**Description**

CIPS Mil-Std-1553 communications timeout threshold.

Data Type

16 bits, unsigned integer, engineering units = sec.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.318 cips ms1553_ts_a_s_ct**Description**

Mil-Std-1553 Bus-A message success counter, time sync mode code channel.

Data Type

32 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.319 cips ms1553_ts_b_s_ct**Description**

Mil-Std-1553 Bus-B message success counter, time sync mode code channel.

Data Type

32 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.320 cips ms1553_tv_a_s_ct**Description**

Mil-Std-1553 Bus-A message success counter, time value channel.

Data Type

32 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.321 cips ms1553_tv_b_s_ct**Description**

Mil-Std-1553 Bus-B message success counter, time value channel.

Data Type

32 bits, unsigned integer, engineering units = any.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

E.322 cips ms_delay

Description

Number of times to execute the loop in the waitMs() function to delay for 1 ms.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.323 cips mx_bos_ena

Description

CIPS -X camera bright object sensor enable.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
FALSE	0	CAUTION
TRUE	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
disable cips bos (46)	Disable the CIPS bright object sensor.
enable cips bos (59)	Enable the CIPS bright object sensor.

E.324 cips mx_bos_level

Description

CIPS -X camera BOS trigger level.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	-0.00030517578125
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Telemetry Source Packets

cips hw_hk (492)

E.325 cips mx_bos_thresh**Description**

CIPS -X camera bright object sensor threshold.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	-0.00030517578125
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Discussion

The CIPS housekeeping FPGA will automatically disable high voltage power to the CIPS cameras if the bright object sensor value is greater than this threshold value. The FPGA also prevents these high voltages from being enabled if

the threshold is exceeded at the time the FPGA gets the instruction to enable high voltage power.

Telemetry Source Packets

cips hw_hk (492)

E.326 cips mx_cam_clear_ct

Description

Last commanded CIPS -X camera clear count setting.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

camera fpga

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips clear_ct (96)	Set the number of times the specified CCDs will be cleared before data is taken.

E.327 cips mx_cam_f_sync

Description

Last commanded CIPS -X camera frame synronization delay setting.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

camera fpga

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips frame_sync (100)	Set the delay time between the 'GO' to take an image and CCD readout.

E.328 cips mx_cam_mcp_pwr

Description

CIPS -X camera micro-channel plate power state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OFF	0	GOOD
ON	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
turn off cips mcp (119)	Turns off the MCP high voltage power supply for the given CIPS camera(s).
turn on cips mcp (123)	Turns on the MCP high voltage power supply for the given CIPS camera(s).

E.329 cips mx_cam_mcpv_mon**Description**

CIPS -X camera multi-channel plate voltage monitor.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	0.0726609006524086
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
set cips mcp_volts (105)	Set the microchannel plate voltage level for the given CIPS camera(s).

E.330 cips mx_cam_mode**Description**

Last commanded CIPS -X camera imaging mode setting.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

camera fpga

State Conversions

state	value	desirability
TDI	0	GOOD
SNAPSHOT	1	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
select cips img_mode (94)	Select the type of imaging to be performed with this camera.

E.331 cips mx_cam_p_pwr**Description**

CIPS -X camera phosphor power state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OFF	0	GOOD
ON	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
turn on cips phosphr (124)	Turns on the phosphor high voltage power supply for the given CIPS camera(s).
turn off cips phosphr (120)	Turns off the phosphor high voltage power supply for the given CIPS camera(s).

E.332 cips mx_cam_pv_mon**Description**

CIPS -X camera phosphor voltage monitor.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	0.508626282215118
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Telemetry Source Packets

cips hw_hk (492)

E.333 cips mx_cam_pwr**Description**

CIPS -X camera power status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OFF	0	GOOD
ON	1	GOOD

Telemetry Source Packets**cips hw_hk (492)****Related Commands**

<i>command</i>	<i>description</i>
turn on cips camera (122)	Enable power to the selected CIPS camera(s).
turn off cips camera (117)	Disable power to the selected CIPS camera(s).

E.334 cips mx_cam_ss_time**Description**

Last commanded CIPS -X camera snapshot time setting.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

camera fpga

Telemetry Source Packets**cips sw_status (514)****Related Commands**

<i>command</i>	<i>description</i>
set cips integration (105)	Set the integration time for the given CIPS camera(s).

E.335 cips mx_cam_tdi_rows**Description**

Last commanded CIPS -X camera TDI row count setting.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

camera fpga

Telemetry Source Packets**cips sw_status (514)**

Related Commands

<i>command</i>	<i>description</i>
set cips tdi_row.ct (108)	Set the number of pixel rows over which to perform Time Delayed Integration.

E.336 cips mx_cam_tdi_time**Description**

Last commanded CIPS -X camera TDI time setting.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

camera fpga

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips tdi_tm (109)	Set the amount of time Time Delayed Integration will be performed per pixel row.

E.337 cips mx_ccd_temp**Description**

CIPS -X camera CCD temperature.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Limits

alarm	limit
red high	45
yellow high	5
yellow low	-35
red low	-40
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.338 cips mx_dc_status_pri**Description**

CIPS -X camera dust cover status indicator, primary circuit.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OPEN	0	CAUTION
CLOSED	1	GOOD

Discussion

This telemetry point gives one indication of the dust cover's status, with the other indication given by the related telemetry item. These are two independent observations of the dust cover state, so if they disagree, troubleshooting must be done by CIPS hardware engineers to determine the true state of the CIPS dust cover.

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
open cips dust_cover (86)	Deploy the dust cover protecting the given camera.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips mx_dc_status_sec (300)	CIPS -X camera dust cover status indicator, secondary circuit.

E.339 cips mx_dc_status_sec**Description**

CIPS -X camera dust cover status indicator, secondary circuit.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OPEN	0	CAUTION
CLOSED	1	GOOD

Discussion

This telemetry point gives one indication of the dust cover's status, with the other indication given by the related telemetry item. These are two independent observations of the dust cover state, so if they disagree, troubleshooting must be done by CIPS hardware engineers to determine the true state of the CIPS dust cover.

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
open cips dust_cover (86)	Deploy the dust cover protecting the given camera.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips mx_dc_status_pri (299)	CIPS -X camera dust cover status indicator, primary circuit.

E.340 cips mx_ib_of**Description**

Processor FPGA -X camera image buffer overflow interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets**cips exception (484)****E.341 cips mx_ib_of_mk****Description**

Processor FPGA -X camera image buffer overflow interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets**cips exception (484)****E.342 cips mx_img_end****Description**

Processor FPGA -X camera end of image interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets**cips exception (484)**

E.343 cips mx_img_end_mk

Description

Processor FPGA -X camera end of image interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.344 cips mx_int_temp

Description

CIPS -X camera intensifier temperature.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

hk fpga

Analog Conversions

Segment 1

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Limits

alarm	limit
red high	45
yellow high	40
yellow low	-35
red low	-40
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.345 cips mx_ips_pwr**Description**

CIPS -X camera high voltage power supply enable.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OFF	0	GOOD
ON	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
turn off cips ips (118)	Turn off power to the given CIPS intensifier power supply.
turn on cips ips (122)	Turn on low voltage power to the given CIPS high voltage power supply.

E.346 cips mx_lens_a_temp**Description**

CIPS -X camera lens temperature at sensor A.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

hk fpga

Analog Conversions

Segment 1

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Limits

alarm	limit
red high	45
yellow high	20
yellow low	-20
red low	-40
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.347 cips mx_lens_b_temp**Description**

CIPS -X camera lens temperature at sensor B.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Limits

alarm	limit
red high	45
yellow high	20
yellow low	-20
red low	-40
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.348 cips mx lens htr_ena**Description**

CIPS -X camera lens heater power enable.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
FALSE	0	GOOD
TRUE	1	GOOD

Discussion

Actual application of power for the lens heater is controlled by the CIPS HK FPGA according to the high and low threshold settings (see related telemetry).

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
disable cips lens_htr (52)	Enable the CIPS lens heater for the given camera(s).
enable cips lens_htr (65)	Enable the CIPS lens heater for the given camera(s).

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips i.l_htr_uthresh (254)	Initial (boot) value for the CIPS cameras' lens heater upper threshold.
cips lens_htr_lthresh (267)	Lens heater lower threshold.
cips i.l_htr_lthresh (253)	Initial (boot) value for the CIPS cameras' lens heater lower threshold.
cips lens_htr_uthresh (268)	Lens heater upper threshold.

E.349 cips mx_my_ips_temp**Description**

CIPS -X and -Y camera intensifier power supply temperature.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Limits

alarm	limit
red high	45
yellow high	40
yellow low	-35
red low	-40
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.350 cips mx_opcode_err**Description**

Processor FPGA -X camera opcode error interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.351 cips mx_opcode_err_mk**Description**

Processor FPGA -X camera opcode error interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.352 cips mx_parity_err**Description**

Processor FPGA -X camera parity error interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.353 cips mx_parity_err_mk

Description

Processor FPGA -X camera parity error interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.354 cips mx_rx

Description

Processor FPGA -X camera receive data ready interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.355 cips mx_rx_mk

Description

Processor FPGA -X camera receive data ready interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.356 cips mx_seq_err**Description**

Processor FPGA -X camera row sequence error interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.357 cips mx_seq_err_mk**Description**

Processor FPGA -X camera row sequence error interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets**cips exception** (484)**E.358 cips mx_shutr_disable****Description**

CIPS -X camera shutter disable flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
FALSE	0	GOOD
TRUE	1	GOOD

Discussion

This flag shows wheter the CIPS housekeeping FPGA will act on shutter open/close requests from the camera FPGA or ignore them.

Telemetry Source Packets**cips hw_hk** (492)**Related Commands**

<i>command</i>	<i>description</i>
enable cips shutter (68)	Enable the autonomous opening and closing of the given shuuter(s).
disable cips shutter (54)	Disable the autonomous opening and closing of the given shuuter(s).

E.359 cips mx_shutr_state**Description**

CIPS -X camera shutter state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
CLOSED	0	GOOD
OPEN	1	GOOD

Telemetry Source Packets**cips hw_hk** (492)**E.360 cips my_bos_ena****Description**

CIPS -Y camera bright object sensor enable.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
FALSE	0	CAUTION
TRUE	1	GOOD

Telemetry Source Packets**cips hw_hk** (492)**Related Commands**

<i>command</i>	<i>description</i>
disable cips bos (46)	Disable the CIPS bright object sensor.
enable cips bos (59)	Enable the CIPS bright object sensor.

E.361 cips my_bos_level**Description**

CIPS -Y camera BOS trigger level.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

hk fpga

Analog Conversions

Segment 1

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	-0.00030517578125
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Telemetry Source Packets

cips hw_hk (492)

E.362 cips my_bos_thresh**Description**

CIPS -Y camera bright object sensor threshold.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	-0.00030517578125
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Discussion

The CIPS housekeeping FPGA will automatically disable high voltage power to the CIPS cameras if the bright object sensor value is greater than this threshold value. The FPGA also prevents these high voltages from being enabled if the threshold is exceeded at the time the FPGA gets the instruction to enable high voltage power.

Telemetry Source Packets

cips hw_hk (492)

E.363 cips my_cam_clear_ct**Description**

Last commanded CIPS -Y camera clear count setting.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

camera fpga

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips clear_ct (96)	Set the number of times the specified CCDs will be cleared before data is taken.

E.364 cips my_cam_f_sync**Description**

Last commanded CIPS -Y camera frame synronization delay setting.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

camera fpga

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips frame.sync (100)	Set the delay time between the 'GO' to take an image and CCD readout.

E.365 cips my_cam_mcp_pwr**Description**

CIPS -Y camera micro-channel plate power state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OFF	0	GOOD
ON	1	GOOD

Telemetry Source Packets**cips hw_hk (492)****Related Commands**

<i>command</i>	<i>description</i>
turn off cips mcp (119)	Turns off the MCP high voltage power supply for the given CIPS camera(s).
turn on cips mcp (123)	Turns on the MCP high voltage power supply for the given CIPS camera(s).

E.366 cips my_cam_mcpv_mon**Description**

CIPS -Y camera multi-channel plate voltage monitor.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	0.0726609006524086
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Telemetry Source Packets**cips hw_hk (492)****Related Commands**

<i>command</i>	<i>description</i>
set cips mcp.volts (105)	Set the microchannel plate voltage level for the given CIPS camera(s).

E.367 cips my_cam_mode

Description

Last commanded CIPS -Y camera imaging mode setting.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

camera fpga

State Conversions

state	value	desirability
TDI	0	GOOD
SNAPSHOT	1	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
select cips img_mode (94)	Select the type of imaging to be performed with this camera.

E.368 cips my_cam_p_pwr

Description

CIPS -Y camera phosphor power state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OFF	0	GOOD
ON	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
turn on cips phosphr (124)	Turns on the phosphor high voltage power supply for the given CIPS camera(s).
turn off cips phosphr (120)	Turns off the phosphor high voltage power supply for the given CIPS camera(s).

E.369 cips my_cam_pv_mon**Description**

CIPS -Y camera phosphor voltage monitor.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	0.508626282215118
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Telemetry Source Packets

cips hw_hk (492)

E.370 cips my_cam_pwr**Description**

CIPS -Y camera power status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OFF	0	GOOD
ON	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
turn on cips camera (122)	Enable power to the selected CIPS camera(s).
turn off cips camera (117)	Disable power to the selected CIPS camera(s).

E.371 cips my_cam_ss_time**Description**

Last commanded CIPS -Y camera snapshot time setting.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

camera fpga

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips integration (105)	Set the integration time for the given CIPS camera(s).

E.372 cips my_cam_tdi_rows**Description**

Last commanded CIPS -Y camera TDI row count setting.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

camera fpga

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips tdi_row.ct (108)	Set the number of pixel rows over which to perform Time Delayed Integration.

E.373 cips my_cam_tdi_time**Description**

Last commanded CIPS -Y camera TDI time setting.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

camera fpga

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips tdi_tm (109)	Set the amount of time Time Delayed Integration will be performed per pixel row.

E.374 cips my_ccd_temp**Description**

CIPS -Y camera CCD temperature.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Limits

alarm	limit
red high	45
yellow high	5
yellow low	-35
red low	-40
delta	none

Telemetry Source Packets

cips hw_hk (492)

E.375 cips my_dc_status_pri

Description

CIPS -Y camera dust cover status indicator, primary circuit.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OPEN	0	CAUTION
CLOSED	1	GOOD

Discussion

This telemetry point gives one indication of the dust cover's status, with the other indication given by the related telemetry item. These are two independent observations of the dust cover state, so if they disagree, troubleshooting must be done by CIPS hardware engineers to determine the true state of the CIPS dust cover.

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
open cips dust_cover (86)	Deploy the dust cover protecting the given camera.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips my_dc_status_sec (319)	CIPS -Y camera dust cover status indicator, secondary circuit.

E.376 cips my_dc_status_sec

Description

CIPS -Y camera dust cover status indicator, secondary circuit.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OPEN	0	CAUTION
CLOSED	1	GOOD

Discussion

This telemetry point gives one indication of the dust cover's status, with the other indication given by the related telemetry item. These are two independent observations of the dust cover state, so if they disagree, troubleshooting must be done by CIPS hardware engineers to determine the true state of the CIPS dust cover.

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
open cips dust_cover (86)	Deploy the dust cover protecting the given camera.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips my_dc_status_pri (319)	CIPS -Y camera dust cover status indicator, primary circuit.

E.377 cips my_ib_of**Description**

Processor FPGA -Y camera image buffer overflow interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.378 cips my_ib_of_mk**Description**

Processor FPGA -Y camera image buffer overflow interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.379 cips my_img_end**Description**

Processor FPGA -Y camera end of image interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.380 cips my_img_end_mk**Description**

Processor FPGA -Y camera end of image interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.381 cips my_int_temp**Description**

CIPS -Y camera intensifier temperature.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Limits

alarm	limit
red high	45
yellow high	40
yellow low	-35
red low	-40
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.382 cips my_ips_pwr**Description**

CIPS -Y camera intensifier power enable.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OFF	0	GOOD
ON	1	GOOD

Telemetry Source Packets**cips hw_hk (492)****Related Commands**

<i>command</i>	<i>description</i>
turn off cips ips (118)	Turn off power to the given CIPS intensifier power supply.
turn on cips ips (122)	Turn on low voltage power to the given CIPS high voltage power supply.

E.383 cips my_lens_a_temp**Description**

CIPS -Y camera lens temperature at sensor A.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Limits

alarm	limit
red high	45
yellow high	20
yellow low	-20
red low	-40
delta	none

Telemetry Source Packets**cips hw_hk (492)**

E.384 cips_my_lens_b_temp**Description**

CIPS -Y camera lens temperature at sensor B.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Limits

alarm	limit
red high	45
yellow high	20
yellow low	-20
red low	-40
delta	none

Telemetry Source Packets

cips_hw_hk (492)

E.385 cips_my_lens_htr_ena**Description**

CIPS -Y camera lens heater power enable.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
FALSE	0	GOOD
TRUE	1	GOOD

Discussion

Actual application of power for the lens heater is controlled by the CIPS HK FPGA according to the high and low threshold settings (see related telemetry).

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
disable cips lens_htr (52)	Enable the CIPS lens heater for the given camera(s).
enable cips lens_htr (65)	Enable the CIPS lens heater for the given camera(s).

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips i_l_htr_uthresh (254)	Initial (boot) value for the CIPS cameras' lens heater upper threshold.
cips lens_htr_lthresh (267)	Lens heater lower threshold.
cips i_l_htr_lthresh (253)	Initial (boot) value for the CIPS cameras' lens heater lower threshold.
cips lens_htr_uthresh (268)	Lens heater upper threshold.

E.386 cips my_opcode_err**Description**

Processor FPGA -Y camera opcode error interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.387 cips my_opcode_err_mk**Description**

Processor FPGA -Y camera opcode error interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.388 cips my_parity_err**Description**

Processor FPGA -Y camera parity error interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.389 cips my_parity_err_mk**Description**

Processor FPGA -Y camera parity error interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.390 cips my_rx**Description**

Processor FPGA -Y camera receive data ready interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.391 cips my_rx_mk**Description**

Processor FPGA -Y camera receive data ready interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.392 cips my_seq_err

Description

Processor FPGA -Y camera row sequence error interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.393 cips my_seq_err_mk

Description

Processor FPGA -Y camera row sequence error interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.394 cips my_shutr_disable

Description

CIPS -Y camera shutter disable flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
FALSE	0	GOOD
TRUE	1	GOOD

Discussion

This flag shows wheter the CIPS housekeeping FPGA will act on shutter open/close requests from the camera FPGA or ignore them.

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
enable cips shutter (68)	Enable the autonomous opening and closing of the given shuuter(s).
disable cips shutter (54)	Disable the autonomous opening and closing of the given shuuter(s).

E.395 cips my_shutr_state**Description**

CIPS -Y camera shutter state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
CLOSED	0	GOOD
OPEN	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

E.396 cips o_seq_addr**Description**

Address in memory where the current operational sequence resides.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets**cips sw_status** (514)**Related Commands**

<i>command</i>	<i>description</i>
init cips sequence (77)	Start a CIPS command sequence.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips o_seq_length (333)	Length of the current operational sequence.
cips o_seq_cur_pos (332)	Address of the current command in the current operational sequence.
cips o_seq_page (333)	Page in memory where the current operational sequence resides.

E.397 cips o_seq_cmd_f_code**Description**

Reason for failure of the last command that failed execution.

Data Type

16 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
SUCCESS	0	GOOD
OPCODE_ERR	1	BAD
LEN_ERR	2	BAD
CHKSUM_ERR	3	BAD
FMT_ERR	4	BAD
RANGE_ERR	5	BAD
INVALID_STATE	6	BAD
INVALID_APID	7	BAD
METHOD_ERR	8	BAD

Telemetry Source Packets**cips sw_status** (514)**Related Telemetry**

<i>telemetry</i>	<i>description</i>
cips o_seq_cmd_f.ct (331)	Number of commands that failed execution by the operational sequence engine.
cips o_seq_cmd.fail (331)	Opcode of the last command that failed execution by the operational seq engine.

E.398 cips o_seq_cmd_f_ct**Description**

Number of commands that failed execution by the operational sequence engine.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips o_seq_cmd_f_code (330)	Reason for failure of the last command that failed execution.
cips o_seq_cmd_fail (331)	Opcode of the last command that failed execution by the operational seq engine.
cips o_seq_cmd_s_ct (332)	Number of commands successfully executed by the operational sequence engine.

E.399 cips o_seq_cmd_fail**Description**

Opcode of the last command that failed execution by the operational seq engine.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips o_seq_cmd_f_code (330)	Reason for failure of the last command that failed execution.
cips o_seq_cmd_f_ct (331)	Number of commands that failed execution by the operational sequence engine.
cips o_seq_cmd_succ (332)	Opcode of the last command executed by the operational sequence engine.

E.400 cips o_seq_cmd_s_ct**Description**

Number of commands successfully executed by the operational sequence engine.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips o_seq_cmd_f_ct (331)	Number of commands that failed execution by the operational sequence engine.
cips o_seq_cmd_succ (332)	Opcode of the last command executed by the operational sequence engine.

E.401 cips o_seq_cmd_succ**Description**

Opcode of the last command executed by the operational sequence engine.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips o_seq_cmd_s_ct (332)	Number of commands successfully executed by the operational sequence engine.
cips o_seq_cmd_fail (331)	Opcode of the last command that failed execution by the operational seq engine.

E.402 cips o_seq_cur_pos**Description**

Address of the current command in the current operational sequence.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips o_seq_length (333)	Length of the current operational sequence.
cips o_seq_addr (329)	Address in memory where the current operational sequence resides.
cips o_seq_page (333)	Page in memory where the current operational sequence resides.

E.403 cips o_seq_length**Description**

Length of the current operational sequence.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips o_seq_length (333)	Length of the current operational sequence.
cips o_seq_addr (329)	Address in memory where the current operational sequence resides.
cips o_seq_page (333)	Page in memory where the current operational sequence resides.
cips o_seq_cur_pos (332)	Address of the current command in the current operational sequence.

E.404 cips o_seq_page**Description**

Page in memory where the current operational sequence resides.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
O_SRAM_P0	0	GOOD
O_SRAM_P1	1	BAD
O_SRAM_P2	2	BAD
O_SRAM_P3	3	BAD
O_SRAM_P4	4	GOOD
O_SRAM_P5	5	GOOD
O_SRAM_P6	6	GOOD
O_SRAM_P7	7	GOOD
O_SRAM_P8	8	BAD
O_SRAM_P9	9	BAD
O_SRAM_P10	10	BAD
O_SRAM_P11	11	BAD
O_EEPROM_P12	12	BAD
O_EEPROM_P13	13	BAD
O_EEPROM_P14	14	BAD
O_EEPROM_P15	15	BAD
IO	128	BAD
I_SRAM	255	BAD

Telemetry Source Packets

cips sw_status (514)

Related Commands

command	description
init cips sequence (77)	Start a CIPS command sequence.

Related Telemetry

telemetry	description
cips o_seq_length (333)	Length of the current operational sequence.
cips o_seq_cur_pos (332)	Address of the current command in the current operational sequence.
cips o_seq_page (333)	Page in memory where the current operational sequence resides.
cips o_seq_addr (329)	Address in memory where the current operational sequence resides.

E.405 cips o_seq_state**Description**

Current state of the operational sequence engine.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
IDLE	0	GOOD
INIT	1	GOOD
ACTIVE	2	GOOD
WAIT	3	GOOD

Telemetry Source Packets**cips sw_status (514)****Related Commands**

<i>command</i>	<i>description</i>
suspend cips abs_tm (115)	Wait until an absolute time.
term cips sequence (117)	Stop the current command sequence in the given sequence engine.
suspend cips rel_tm (116)	Wait for a relative time delay.
init cips sequence (77)	Start a CIPS command sequence.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips o_seq_wait_time (336)	Wait time execution of the next command in for the operational sequence engine.
cips o_seq_wait_type (337)	Type of wait currently being executed in the operational sequence engine.

E.406 cips o_seq_stop_code**Description**

Reason command processing in the operational sequence engine stopped.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
NONE	0	GOOD
END_OF_SEQUENCE	1	GOOD
COMMANDED	2	GOOD
STATE_ERROR	3	BAD
LENGTH_ERROR	4	BAD
USE_COUNT_ERROR	5	BAD
PAGE_ERROR	6	BAD
CHECKSUM_ERROR	7	BAD
PARSE_ERROR	8	BAD

Telemetry Source Packets**cips sw_status (514)**

Related Commands

<i>command</i>	<i>description</i>
term cips sequence (117)	Stop the current command sequence in the given sequence engine.

E.407 cips o_seq_use_ct**Description**

Number of uses remaining for the last operational sequence initiated.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.408 cips o_seq_wait_time**Description**

Wait time execution of the next command in for the operational sequence engine.

Data Type

32 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Discussion

This can be a relative time (i.e. number of seconds left in the wait), or an absolute time (i.e. 2005/300-12:15:59), based on the type of 'suspend' command executed.

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
suspend cips abs_tm (115)	Wait until an absolute time.
suspend cips rel_tm (116)	Wait for a relative time delay.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips o_seq_state (334)	Current state of the operational sequence engine.
cips o_seq_wait_type (337)	Type of wait currently being executed in the operational sequence engine.

E.409 cips o_seq_wait_type

Description

Type of wait currently being executed in the operational sequence engine.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
NONE	0	GOOD
REL	1	GOOD
ABS	2	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
suspend cips abs_tm (115)	Wait until an absolute time.
suspend cips rel_tm (116)	Wait for a relative time delay.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips o_seq_state (334)	Current state of the operational sequence engine.
cips o_seq_wait_time (336)	Wait time execution of the next command in for the operational sequence engine.

E.410 cips obs_id

Description

CIPS observation ID for this science image.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sci_mx_cam (502), cips sci_my_cam (505), cips sci_px_cam (508), cips sci_py_cam (511)

Related Commands

<i>command</i>	<i>description</i>
set cips obs_id (106)	Set a unique identifier for the ensuing science images.

E.411 cips opcode_err_ct**Description**

CIPS CAMI FPGA opcode error count.

Data Type

4 bits, unsigned integer, engineering units = dn.

Measurement Source

cam_i fpga

Limits

alarm	limit
red high	1
yellow high	1
yellow low	-1
red low	-1
delta	<i>none</i>

Telemetry Source Packets

cips_sci_mx_cam (502), cips_sci_my_cam (505), cips_sci_px_cam (508), cips_sci_py_cam (511)

E.412 cips p15i_mon**Description**

CIPS +15V current monitor.

Data Type

16 bits, unsigned integer, engineering units = a.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	0.0001220703125
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Limits

alarm	limit
red high	1.25
yellow high	1.125
yellow low	-1
red low	-1
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.413 cips p15v_mon**Description**

CIPS +15 volt monitor.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-22.25
c1	0.002716064453125
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Limits

alarm	limit
red high	15.75
yellow high	15.5
yellow low	14.5
red low	14.25
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.414 cips p20v_mon**Description**

CIPS +20 volt monitor.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-22.25
c1	0.002716064453125
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Limits

alarm	limit
red high	21
yellow high	20.75
yellow low	19.25
red low	19
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.415 cips p2_5v_mon**Description**

CIPS +2.5 voltage monitor.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

hk fpga

Analog Conversions

Segment 1

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-22.25
c1	0.002716064453125
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Limits

alarm	limit
red high	2.61999988555908
yellow high	2.58999991416931
yellow low	2.41000008583069
red low	2.38000011444092
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.416 cips p352flg

Description

CIPS software status packet header, segment flags

Data Type

2 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sw_status (514)

E.417 cips p352hws

Description

CIPS software status secondary header, seconds (high order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
inform cips sc.clock (73)	Notify FSW of spacecraft time at next time sync.

E.418 cips p352hwss**Description**

CIPS software status secondary header, subseconds (high order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sw_status (514)

E.419 cips p352lws**Description**

CIPS software status secondary header, seconds (low order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
inform cips sc.clock (73)	Notify FSW of spacecraft time at next time sync.

E.420 cips p352lwss**Description**

CIPS software status secondary header, subseconds (low order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sw_status (514)

E.421 cips p352pid**Description**

CIPS software status packet header, type and apid

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sw_status (514)

E.422 cips p352pl**Description**

CIPS software status packet header, packet length

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sw_status (514)

E.423 cips p352sct**Description**

CIPS software status packet header, sequence counter

Data Type

14 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sw_status (514)

E.424 cips p353flg**Description**

CIPS hardware housekeeping packet header, segment flags

Data Type

2 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips hw_hk (492)

E.425 cips p353hws**Description**

CIPS hardware housekeeping secondary header, seconds (high order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips hw_hk (492)

E.426 cips p353hwss**Description**

CIPS hardware housekeeping secondary header, subseconds (high order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips hw_hk (492)

E.427 cips p353lws**Description**

CIPS hardware housekeeping secondary header, seconds (low order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips hw_hk (492)

E.428 cips p353lwss**Description**

CIPS hardware housekeeping secondary header, subseconds (low order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips hw_hk (492)

E.429 cips p353pid**Description**

CIPS hardware housekeeping packet header, type and apid

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips hw_hk (492)

E.430 cips p353pl**Description**

CIPS hardware housekeeping packet header, packet length

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips hw_hk (492)

E.431 cips p353sct**Description**

CIPS hardware housekeeping packet header, sequence counter

Data Type

14 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips hw_hk (492)

E.432 cips p354flg**Description**

CIPS error packet header, segment flags

Data Type

2 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips err (481)

E.433 cips p354hws**Description**

CIPS error secondary header, seconds (high order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips err (481)

E.434 cips p354hwss**Description**

CIPS error secondary header, subseconds (high order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips err (481)

E.435 cips p354lws**Description**

CIPS error secondary header, seconds (low order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips err (481)

E.436 cips p354lwss**Description**

CIPS error secondary header, subseconds (low order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips err (481)

E.437 cips p354pid**Description**

CIPS error packet header, type and apid

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips err (481)

E.438 cips p354pl**Description**

CIPS error packet header, packet length

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips err (481)

E.439 cips p354sct**Description**

CIPS error packet header, sequence counter

Data Type

14 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips err (481)

E.440 cips p355flg**Description**

CIPS event log packet header, segment flags

Data Type

2 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips evt_log (482)

E.441 cips p355hws**Description**

CIPS event log secondary header, seconds (high order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips evt_log (482)

E.442 cips p355hwss**Description**

CIPS event log secondary header, subseconds (high order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips evt_log (482)

E.443 cips p355lws**Description**

CIPS event log secondary header, seconds (low order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips evt_log (482)

E.444 cips p355lwss**Description**

CIPS event log secondary header, subseconds (low order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips evt_log (482)

E.445 cips p355pid**Description**

CIPS event log packet header, type and apid

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips evt_log (482)

E.446 cips p355pl**Description**

CIPS event log packet header, packet length

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips evt_log (482)

E.447 cips p355sct**Description**

CIPS event log packet header, sequence counter

Data Type

14 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips evt_log (482)

E.448 cips p356flg**Description**

CIPS memory dump packet header, segment flags

Data Type

2 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips mem_dump (499)

E.449 cips p356hws**Description**

CIPS memory dump secondary header, seconds (high order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips mem_dump (499)

E.450 cips p356hwss**Description**

CIPS memory dump secondary header, subseconds (high order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips mem_dump (499)

E.451 cips p356lws**Description**

CIPS memory dump secondary header, seconds (low order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips mem_dump (499)

E.452 cips p356lwss**Description**

CIPS memory dump secondary header, subseconds (low order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips mem_dump (499)

E.453 cips p356pid**Description**

CIPS memory dump packet header, type and apid

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips mem_dump (499)

E.454 cips p356pl**Description**

CIPS memory dump packet header, packet length

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips mem_dump (499)

E.455 cips p356sct**Description**

CIPS memory dump packet header, sequence counter

Data Type

14 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips mem_dump (499)

E.456 cips p357flg**Description**

CIPS post-mortem packet header, segment flags

Data Type

2 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips post_mortem (500)

E.457 cips p357hws**Description**

CIPS post-mortem secondary header, seconds (high order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips post_mortem (500)

E.458 cips p357hwss**Description**

CIPS post-mortem secondary header, subseconds (high order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips post_mortem (500)

E.459 cips p357lws**Description**

CIPS post-mortem secondary header, seconds (low order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips post_mortem (500)

E.460 cips p357lwss**Description**

CIPS post-mortem secondary header, subseconds (low order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips post_mortem (500)

E.461 cips p357pid**Description**

CIPS post-mortem packet header, type and apid

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips post_mortem (500)

E.462 cips p357pl**Description**

CIPS post-mortem packet header, packet length

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips post_mortem (500)

E.463 cips p357sct**Description**

CIPS post-mortem packet header, sequence counter

Data Type

14 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips post_mortem (500)

E.464 cips p358flg**Description**

CIPS exception packet header, segment flags

Data Type

2 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips exception (484)

E.465 cips p358hws**Description**

CIPS exception secondary header, seconds (high order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips exception (484)

E.466 cips p358hwss**Description**

CIPS exception secondary header, subseconds (high order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips exception (484)

E.467 cips p358lws**Description**

CIPS exception secondary header, seconds (low order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips exception (484)

E.468 cips p358lwss**Description**

CIPS exception secondary header, subseconds (low order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips exception (484)

E.469 cips p358pid**Description**

CIPS exception packet header, type and apid

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips exception (484)

E.470 cips p358pl**Description**

CIPS exception packet header, packet length

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips exception (484)

E.471 cips p358sct**Description**

CIPS exception packet header, sequence counter

Data Type

14 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips exception (484)

E.472 cips p359flg**Description**

CIPS command echo packet header, segment flags

Data Type

2 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips cmd_echo (480)

E.473 cips p359hws**Description**

CIPS command echo secondary header, seconds (high order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips cmd_echo (480)

E.474 cips p359hwss**Description**

CIPS command echo secondary header, subseconds (high order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips cmd_echo (480)

E.475 cips p359lws**Description**

CIPS command echo secondary header, seconds (low order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips cmd_echo (480)

E.476 cips p359lwss**Description**

CIPS command echo secondary header, subseconds (low order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips cmd_echo (480)

E.477 cips p359pid**Description**

CIPS command echo packet header, type and apid

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips cmd_echo (480)

E.478 cips p359pl**Description**

CIPS command echo packet header, packet length

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips cmd_echo (480)

E.479 cips p359sct**Description**

CIPS command echo packet header, sequence counter

Data Type

14 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips cmd_echo (480)

E.480 cips p360flg**Description**

CIPS science +X camera packet header, segment flags

Data Type

2 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_px_cam (508)

E.481 cips p360hws**Description**

CIPS +X camera science data packet secondary header, seconds (high order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_px_cam (508)

E.482 cips p360hwss**Description**

CIPS science +X camera secondary header, subseconds (high order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_px_cam (508)

E.483 cips p360lws**Description**

CIPS science +X camera secondary header, seconds (low order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_px_cam (508)

E.484 cips p360lwss**Description**

CIPS +X camera science data packet secondary header, subseconds (low order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_px_cam (508)

E.485 cips p360pid**Description**

CIPS science +X camera packet header, type and apid

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_px_cam (508)

E.486 cips p360pl**Description**

CIPS +X camera science data packet header, packet length

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_px_cam (508)

E.487 cips p360sct**Description**

CIPS science +X camera packet header, sequence counter

Data Type

14 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_px_cam (508)

E.488 cips p361flg**Description**

CIPS science +Y camera packet header, segment flags

Data Type

2 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_py_cam (511)

E.489 cips p361hws**Description**

CIPS +Y camera science data packet secondary header, seconds (high order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_py_cam (511)

E.490 cips p361hwss**Description**

CIPS +Y camera science data packet secondary header, subseconds (high order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_py_cam (511)

E.491 cips p361lws**Description**

CIPS science +Y camera science data packet secondary header, seconds (low order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_py_cam (511)

E.492 cips p361lwss**Description**

CIPS science +Y camera secondary header, subseconds (low order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_py_cam (511)

E.493 cips p361pid**Description**

CIPS +Y camera science data packet header, type and apid

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_py_cam (511)

E.494 cips p361pl**Description**

CIPS science +Y camera packet header, packet length

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_py_cam (511)

E.495 cips p361sct**Description**

CIPS science +Y camera packet header, sequence counter

Data Type

14 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_py_cam (511)

E.496 cips p362flg**Description**

CIPS -X camera science data packet header, segment flags

Data Type

2 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_mx_cam (502)

E.497 cips p362hws**Description**

CIPS science -X camera secondary header, seconds (high order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_mx_cam (502)

E.498 cips p362hwss**Description**

CIPS science -X camera secondary header, subseconds (high order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_mx_cam (502)

E.499 cips p362lws**Description**

CIPS science -X camera secondary header, seconds (low order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_mx_cam (502)

E.500 cips p362lwss**Description**

CIPS -X camera science data packet secondary header, subseconds (low order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_mx_cam (502)

E.501 cips p362pid**Description**

CIPS -X camera science data packet header, type and apid

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci.mx.cam (502)

E.502 cips p362pl**Description**

CIPS -X camera science data packet header, packet length

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci.mx.cam (502)

E.503 cips p362sct**Description**

CIPS science -X camera packet header, sequence counter

Data Type

14 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci.mx.cam (502)

E.504 cips p363flg**Description**

CIPS -Y camera science data packet header, segment flags

Data Type

2 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_my_cam (505)

E.505 cips p363hws**Description**

CIPS -Y camera science data packet secondary header, seconds (high order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_my_cam (505)

E.506 cips p363hwss**Description**

CIPS -Y camera science data packet secondary header, subseconds (high order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_my_cam (505)

E.507 cips p363lws**Description**

CIPS -Y camera science data packet secondary header, seconds (low order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_my_cam (505)

E.508 cips p363lwss**Description**

CIPS -Y camera science data packet secondary header, subseconds (low order)

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_my_cam (505)

E.509 cips p363pid**Description**

CIPS -Y camera science data packet header, type and apid

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_my_cam (505)

E.510 cips p363pl**Description**

CIPS -Y camera science data packet header, packet length

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_my_cam (505)

E.511 cips p363sct**Description**

CIPS -Y camera science data packet header, sequence counter

Data Type

14 bits, unsigned integer, engineering units = dn.

Measurement Source

CIPS

Telemetry Source Packets

cips sci_my_cam (505)

E.512 cips p5i_mon**Description**

CIPS +5V current monitor.

Data Type

16 bits, unsigned integer, engineering units = a.

Measurement Source

hk fpga

Analog Conversions

Segment 1

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	0.00040690103196539
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Limits

alarm	limit
red high	1.875
yellow high	1.64999997615814
yellow low	-1
red low	-1
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.513 cips p5v_mon**Description**

CIPS +5 volt monitor.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-22.25
c1	0.002716064453125
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Limits

alarm	limit
red high	5.25
yellow high	5.19999980926514
yellow low	4.80000019073486
red low	4.75
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.514 cips parity_err_ct**Description**

CIPS CAMI FPGA parity error count.

Data Type

8 bits, unsigned integer, engineering units = dn.

Measurement Source

cami fpga

Limits

alarm	limit
red high	1
yellow high	1
yellow low	-1
red low	-1
delta	<i>none</i>

Telemetry Source Packets

cips sci_mx_cam (502), cips sci_my_cam (505), cips sci_px_cam (508), cips sci_py_cam (511)

E.515 cips parm_tbl_crc**Description**

CRC of parameter table (less this 16-bit word).

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.516 cips parm_tbl_flag**Description**

Indicates which copy of the FSW parameter table is in use.

Data Type

16 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
EEPROM	0	CAUTION
SRAM	21845	GOOD

Telemetry Source Packets

cips sw_status (514)

E.517 cips pixel_count_err**Description**

Error flag set when number of pixels sent by camera FPGA is not correct.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
NO_ERROR	0	GOOD
ERROR	1	BAD

Discussion

At the end of image collection, CIPS FSW reads the camera FPGA to determine how many pixels it sent to the CPA. If this count is not equal to the number of pixels it should have sent, based on the commanded image size and camera and ccd binning options, this flag is set. The number of pixels read from SDRAM will be the expected number, not the number of pixels written to SDRAM. Thus, if this flag is set, the checksum error flag for this image should also be set.

Telemetry Source Packets

cips sci_mx_cam (502), cips sci_my_cam (505), cips sci_px_cam (508), cips sci_py_cam (511)

E.518 cips pixel_stream

Description

Science data bitstream, contains integral number of image pixels.

Data Type

7744 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Discussion

This bitstream contains science image pixel data in an efficiently packed array. The size in bits per pixel of each element in this array is given by the 'cips bits_per_pixel' telemetry point while the number of elements in this array is given by the 'cips pixels_in_pkt' telemetry point.

Telemetry Source Packets

cips sci_mx_cam (502), cips sci_my_cam (505), cips sci_px_cam (508), cips sci_py_cam (511)

E.519 cips pixels_in_pkt

Description

Number of pixels in this science data packet.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sci_mx_cam (502), cips sci_my_cam (505), cips sci_px_cam (508), cips sci_py_cam (511)

E.520 cips pixels_per_map

Description

Number of pixels in a single image 'map' in SDRAM.

Data Type

32 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.521 cips pixels_per_pkt**Description**

Number of pixels in a full science data packet.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Discussion

Calculated by CIPS FSW based on image binning and compression options. Images span multiple CCSDS telemetry packets.

Telemetry Source Packets

cips sw_status (514)

E.522 cips pkt_crc**Description**

Packet CRC, used for data integrity verification.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips cmd_echo (480), cips err (481), cips evt_log (482), cips exception (484), cips hw_hk (492), cips mem_dump (499), cips post_mortem (500), cips sci_mx_cam (502), cips sci_my_cam (505), cips sci_px_cam (508), cips sci_py_cam (511), cips sw_status (514)

E.523 cips pkt_filler**Description**

cips packet filler.

Data Type

8048 bits, character string, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips_err (481)

E.524 cips_pkt_version**Description**

Packet version number, used for decom mapping.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips_cmd_echo (480), cips_err (481), cips_exception (484), cips_hw_hk (492), cips_mem_dump (499), cips_sci_mx_cam (502), cips_sci_my_cam (505), cips_sci_px_cam (508), cips_sci_py_cam (511), cips_sw_status (514)

E.525 cips_pkt_version_el**Description**

Packet version number, used for decom mapping.

Data Type

8 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips_evt_log (482), cips_post_mortem (500)

E.526 cips_pm_enable**Description**

EEPROM write enable.

Data Type

16 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
FALSE	0	CAUTION
TRUE	1	GOOD

Telemetry Source Packets**cips sw_status** (514)**Related Commands**

<i>command</i>	<i>description</i>
enable cips exc_pm (64)	Re-enable CIPS exception and post-mortem data recording.

E.527 cips pm_event_ct**Description**

Number of events in this post-mortem packet.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets**cips post_mortem** (500)**E.528 cips pm_event_info****Description**

CIPS post-mortem subpacket structure.

Data Type

0 bits, subpacket, engineering units = any.

Measurement Source

software

Telemetry Source Packets**cips post_mortem** (500)

E.529 cips pm_key**Description**

Event type.

Data Type

32 bits, discrete state, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips post_mortem (500)

E.530 cips pm_p0**Description**

Event parameter 0.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips post_mortem (500)

E.531 cips pm_p1**Description**

Event parameter 1.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips post_mortem (500)

E.532 cips pm_p2**Description**

Event parameter 2.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips post_mortem (500)

E.533 cips pm_p3**Description**

Event parameter 3.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips post_mortem (500)

E.534 cips pm_pc**Description**

Software program counter value at event occurrence.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips post_mortem (500)

E.535 cips pm_time_s**Description**

Event time.

Data Type

32 bits, unsigned integer, engineering units = sec.

Measurement Source

software

Telemetry Source Packets

cips post_mortem (500)

E.536 cips pm_time_ss**Description**

Event time.

Data Type

32 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips post_mortem (500)

E.537 cips pm_to**Description**

State of downlink enable for CIPS post-mortem telemetry packets.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
DISABLED	0	CAUTION
ENABLED	1	GOOD

Telemetry Source Packets**cips sw_status** (514)**Related Commands**

<i>command</i>	<i>description</i>
set cips to_mask (110)	Set the CIPS telemetry output enable mask.
disable cips tlm.out (56)	Disable downlink of the given CIPS telemetry packet.
enable cips tlm.out (70)	Enable downlink of the given CIPS telemetry packet.

E.538 cips proc_fpga_ver**Description**

CIPS processor FPGA version.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

processor fpga

Telemetry Source Packets**cips hw_hk** (492)**E.539 cips proc_me_bterr****Description**

Processor FPGA bus timeout machine error state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets**cips exception** (484)

E.540 cips proc_me_mprot**Description**

Processor FPGA memory protection machine error state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.541 cips proc_mode**Description**

Processor FPGA program mode.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
SUROM	0	GOOD
SRAM	1	GOOD

Telemetry Source Packets

cips exception (484)

E.542 cips proc_mode_rset**Description**

Processor FPGA program mode reset flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
OK	0	GOOD
RESET	1	GOOD

Discussion

This telemetry item helps identify the reason for the last processor reset.

Telemetry Source Packets

cips exception (484)

E.543 cips proc_mx_cam_rs**Description**

Processor FPGA counter for image data rows sent by camera 2.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

proc fpga

Telemetry Source Packets

cips hw_hk (492)

E.544 cips proc_my_cam_rs**Description**

Processor FPGA counter for image data rows sent by camera 3.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

proc fpga

Telemetry Source Packets

cips hw_hk (492)

E.545 cips proc_pcb_temp

Description

CIPS processor board temperature.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

hk fpga

Analog Conversions

Segment 1

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Limits

alarm	limit
red high	63
yellow high	58
yellow low	-35
red low	-40
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.546 cips proc_px_cam_rs

Description

Processor FPGA counter for image data rows sent by camera 0.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

proc fpga

Telemetry Source Packets**cips hw_hk** (492)**E.547 cips proc_py_cam_rs****Description**

Processor FPGA counter for image data rows sent by camera 1.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

proc fpga

Telemetry Source Packets**cips hw_hk** (492)**E.548 cips proc_self_rset****Description**

Processor FPGA self reset flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
OK	0	GOOD
RESET	1	GOOD

Discussion

This telemetry item helps identify the reason for the last processor reset.

Telemetry Source Packets**cips exception** (484)**E.549 cips proc_wd_rset****Description**

Processor FPGA watchdog reset flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
OK	0	GOOD
RESET	1	GOOD

Discussion

This telemetry item helps identify the reason for the last processor reset.

Telemetry Source Packets

cips exception (484)

E.550 cips px_bos_ena**Description**

CIPS +X camera bright object sensor enable.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
FALSE	0	CAUTION
TRUE	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
disable cips bos (46)	Disable the CIPS bright object sensor.
enable cips bos (59)	Enable the CIPS bright object sensor.

E.551 cips px_bos_level**Description**

CIPS +X camera BOS trigger level.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	-0.00030517578125
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Telemetry Source Packets

cips hw_hk (492)

E.552 cips px_bos_thresh**Description**

CIPS +X camera bright object sensor threshold.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	-0.00030517578125
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Discussion

The CIPS housekeeping FPGA will automatically disable high voltage power to the CIPS cameras if the bright object sensor value is greater than this threshold value. The FPGA also prevents these high voltages from being enabled if the threshold is exceeded at the time the FPGA gets the instruction to enable high voltage power.

Telemetry Source Packets

cips hw_hk (492)

E.553 cips px_cam_clear_ct**Description**

Last commanded CIPS +X camera clear count setting.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

camera fpga

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips clear_ct (96)	Set the number of times the specified CCDs will be cleared before data is taken.

E.554 cips px_cam_f_sync**Description**

Last commanded CIPS +X camera frame synronization delay setting.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

camera fpga

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips frame_sync (100)	Set the delay time between the 'GO' to take an image and CCD readout.

E.555 cips px_cam_mcp_pwr**Description**

CIPS +X camera micro-channel plate power state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OFF	0	GOOD
ON	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
turn off cips mcp (119)	Turns off the MCP high voltage power supply for the given CIPS camera(s).
turn on cips mcp (123)	Turns on the MCP high voltage power supply for the given CIPS camera(s).

E.556 cips px_cam_mcpv_mon**Description**

CIPS +X camera multi-channel plate voltage monitor.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	0.0726609006524086
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Telemetry Source Packets**cips hw_hk (492)****Related Commands**

<i>command</i>	<i>description</i>
set cips mcp_volts (105)	Set the microchannel plate voltage level for the given CIPS camera(s).

E.557 cips px_cam_mode**Description**

Last commanded imaging mode setting for the +X camera.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

camera fpga

State Conversions

state	value	desirability
TDI	0	GOOD
SNAPSHOT	1	GOOD

Telemetry Source Packets**cips sw_status (514)****Related Commands**

<i>command</i>	<i>description</i>
select cips img_mode (94)	Select the type of imaging to be performed with this camera.

E.558 cips px_cam_p_pwr**Description**

CIPS +X camera phosphor power state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OFF	0	GOOD
ON	1	GOOD

Telemetry Source Packets**cips hw_hk** (492)**Related Commands**

<i>command</i>	<i>description</i>
turn on cips phosphr (124)	Turns on the phosphor high voltage power supply for the given CIPS camera(s).
turn off cips phosphr (120)	Turns off the phosphor high voltage power supply for the given CIPS camera(s).

E.559 cips px_cam_pv_mon**Description**

CIPS +X camera phosphor voltage monitor.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	0.508626282215118
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Telemetry Source Packets**cips hw_hk** (492)**E.560 cips px_cam_pwr****Description**

CIPS +X camera power status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OFF	0	GOOD
ON	1	GOOD

Telemetry Source Packets**cips hw_hk** (492)**Related Commands**

<i>command</i>	<i>description</i>
turn on cips camera (122)	Enable power to the selected CIPS camera(s).
turn off cips camera (117)	Disable power to the selected CIPS camera(s).

E.561 cips px_cam_ss_time**Description**

Last commanded CIPS +X camera snapshot time setting.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

camera fpga

Telemetry Source Packets**cips sw_status** (514)**Related Commands**

<i>command</i>	<i>description</i>
set cips integration (105)	Set the integration time for the given CIPS camera(s).

E.562 cips px_cam_tdi_rows**Description**

Last commanded CIPS +X camera TDI row count setting.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

camera fpga

Telemetry Source Packets**cips sw_status** (514)

Related Commands

<i>command</i>	<i>description</i>
set cips tdi_row.ct (108)	Set the number of pixel rows over which to perform Time Delayed Integration.

E.563 cips px_cam_tdi_time**Description**

Last commanded CIPS +X camera TDI time setting.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

camera fpga

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips tdi_tm (109)	Set the amount of time Time Delayed Integration will be performed per pixel row.

E.564 cips px_ccd_temp**Description**

CIPS +X camera CCD temperature.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Limits

alarm	limit
red high	45
yellow high	5
yellow low	-35
red low	-40
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.565 cips px_dc_status_pri**Description**

CIPS +X camera dust cover status indicator, primary circuit.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OPEN	0	CAUTION
CLOSED	1	GOOD

Discussion

This telemetry point gives one indication of the dust cover's status, with the other indication given by the related telemetry item. These are two independent observations of the dust cover state, so if they disagree, troubleshooting must be done by CIPS hardware engineers to determine the true state of the CIPS dust cover.

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
open cips dust_cover (86)	Deploy the dust cover protecting the given camera.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips px_dc_status_sec (398)	CIPS +X camera dust cover status indicator, secondary circuit.

E.566 cips px_dc_status_sec**Description**

CIPS +X camera dust cover status indicator, secondary circuit.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OPEN	0	CAUTION
CLOSED	1	GOOD

Discussion

This telemetry point gives one indication of the dust cover's status, with the other indication given by the related telemetry item. These are two independent observations of the dust cover state, so if they disagree, troubleshooting must be done by CIPS hardware engineers to determine the true state of the CIPS dust cover.

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
open cips dust_cover (86)	Deploy the dust cover protecting the given camera.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips px_dc_status_pri (397)	CIPS +X camera dust cover status indicator, primary circuit.

E.567 cips px_ib_of**Description**

Processor FPGA +X camera image buffer overflow interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets**cips exception (484)****E.568 cips px_ib_of_mk****Description**

Processor FPGA +X camera image buffer overflow interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets**cips exception (484)****E.569 cips px_img_end****Description**

Processor FPGA +X camera end of image interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets**cips exception (484)**

E.570 cips px_img_end_mk**Description**

Processor FPGA +X camera end of image interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.571 cips px_int_temp**Description**

CIPS +X camera intensifier temperature.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Limits

alarm	limit
red high	45
yellow high	40
yellow low	-35
red low	-40
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.572 cips px_ips_pwr**Description**

CIPS +X camera high voltage power supply enable.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OFF	0	GOOD
ON	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
turn off cips ips (118)	Turn off power to the given CIPS intensifier power supply.
turn on cips ips (122)	Turn on low voltage power to the given CIPS high voltage power supply.

E.573 cips px_lens_a_temp**Description**

CIPS +X camera lens temperature at sensor A.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

hk fpga

Analog Conversions

Segment 1

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Limits

alarm	limit
red high	45
yellow high	20
yellow low	-20
red low	-40
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.574 cips px_lens_b_temp**Description**

CIPS +X camera lens temperature at sensor B.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Limits

alarm	limit
red high	45
yellow high	20
yellow low	-20
red low	-40
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.575 cips px_lens_htr_ena**Description**

CIPS +X camera lens heater power enable.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
FALSE	0	GOOD
TRUE	1	GOOD

Discussion

Actual application of power for the lens heater is controlled by the CIPS HK FPGA according to the high and low threshold settings (see related telemetry).

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
disable cips lens_htr (52)	Enable the CIPS lens heater for the given camera(s).
enable cips lens_htr (65)	Enable the CIPS lens heater for the given camera(s).

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips i.l_htr_uthresh (254)	Initial (boot) value for the CIPS cameras' lens heater upper threshold.
cips lens_htr_lthresh (267)	Lens heater lower threshold.
cips i.l_htr_lthresh (253)	Initial (boot) value for the CIPS cameras' lens heater lower threshold.
cips lens_htr_uthresh (268)	Lens heater upper threshold.

E.576 cips px_opcode_err**Description**

Processor FPGA +X camera opcode error interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.577 cips px_opcode_err_mk**Description**

Processor FPGA +X camera opcode error interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.578 cips px_parity_err**Description**

Processor FPGA +X camera parity error interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.579 cips px_parity_err_mk**Description**

Processor FPGA +X camera parity error interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.580 cips px_py_ips_temp**Description**

CIPS PX and PY camera intensifier power supply temperature.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

hk fpga

Analog Conversions

Segment 1

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Limits

alarm	limit
red high	45
yellow high	40
yellow low	-35
red low	-40
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.581 cips px_rx**Description**

Processor FPGA +X camera receive data ready interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.582 cips px_rx_mk

Description

Processor FPGA +X camera receive data ready interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.583 cips px_seq_err

Description

Processor FPGA +X camera row sequence error interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.584 cips px_seq_err_mk

Description

Processor FPGA +X camera row sequence error interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.585 cips px_shutr_disable**Description**

CIPS +X camera shutter disable flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
FALSE	0	GOOD
TRUE	1	GOOD

Discussion

This flag shows whether the CIPS housekeeping FPGA will act on shutter open/close requests from the camera FPGA or ignore them.

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
enable cips shutter (68)	Enable the autonomous opening and closing of the given shutter(s).
disable cips shutter (54)	Disable the autonomous opening and closing of the given shutter(s).

E.586 cips px_shutr_state**Description**

CIPS +X camera shutter state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
CLOSED	0	GOOD
OPEN	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

E.587 cips py_bos_ena**Description**

CIPS +Y camera bright object sensor enable.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
FALSE	0	CAUTION
TRUE	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
disable cips bos (46)	Disable the CIPS bright object sensor.
enable cips bos (59)	Enable the CIPS bright object sensor.

E.588 cips py_bos_level**Description**

CIPS +Y camera BOS trigger level.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

hk fpga

Analog Conversions

Segment 1

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	-0.00030517578125
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Telemetry Source Packets

cips hw_hk (492)

E.589 cips py_bos_thresh

Description

CIPS +Y camera bright object sensor threshold.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

hk fpga

Analog Conversions

Segment 1

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	-0.00030517578125
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Discussion

The CIPS housekeeping FPGA will automatically disable high voltage power to the CIPS cameras if the bright object sensor value is greater than this threshold value. The FPGA also prevents these high voltages from being enabled if the threshold is exceeded at the time the FPGA gets the instruction to enable high voltage power.

Telemetry Source Packets**cips hw_hk (492)****E.590 cips py_cam_clear_ct****Description**

Last commanded CIPS +Y camera clear count setting.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

camera fpga

Telemetry Source Packets**cips sw_status (514)****Related Commands**

<i>command</i>	<i>description</i>
set cips clear_ct (96)	Set the number of times the specified CCDs will be cleared before data is taken.

E.591 cips py_cam_f_sync**Description**

Last commanded CIPS +Y camera frame synronization delay setting.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

camera fpga

Telemetry Source Packets**cips sw_status (514)****Related Commands**

<i>command</i>	<i>description</i>
set cips frame_sync (100)	Set the delay time between the 'GO' to take an image and CCD readout.

E.592 cips py_cam_mcp_pwr**Description**

CIPS +Y camera micro-channel plate power state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OFF	0	GOOD
ON	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
turn off cips mcp (119)	Turns off the MCP high voltage power supply for the given CIPS camera(s).
turn on cips mcp (123)	Turns on the MCP high voltage power supply for the given CIPS camera(s).

E.593 cips py_cam_mcpv_mon**Description**

CIPS +Y camera multi-channel plate voltage monitor.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	0.0726609006524086
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
set cips mcp_volts (105)	Set the microchannel plate voltage level for the given CIPS camera(s).

E.594 cips py_cam_mode**Description**

Last commanded CIPS +Y camera imaging mode setting.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

camera fpga

State Conversions

state	value	desirability
TDI	0	GOOD
SNAPSHOT	1	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
select cips img_mode (94)	Select the type of imaging to be performed with this camera.

E.595 cips py_cam_p_pwr**Description**

CIPS +Y camera phosphor power state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OFF	0	GOOD
ON	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
turn on cips phosphr (124)	Turns on the phosphor high voltage power supply for the given CIPS camera(s).
turn off cips phosphr (120)	Turns off the phosphor high voltage power supply for the given CIPS camera(s).

E.596 cips py_cam_pv_mon**Description**

CIPS +Y camera phosphor voltage monitor.

Data Type

16 bits, unsigned integer, engineering units = V.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	0.508626282215118
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Telemetry Source Packets

cips hw_hk (492)

E.597 cips py_cam_pwr**Description**

CIPS +Y camera power status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OFF	0	GOOD
ON	1	GOOD

Telemetry Source Packets**cips hw_hk** (492)**Related Commands**

<i>command</i>	<i>description</i>
turn on cips camera (122)	Enable power to the selected CIPS camera(s).
turn off cips camera (117)	Disable power to the selected CIPS camera(s).

E.598 cips py_cam_ss_time**Description**

Last commanded CIPS +Y camera snapshot time setting.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

camera fpga

Telemetry Source Packets**cips sw_status** (514)**Related Commands**

<i>command</i>	<i>description</i>
set cips integration (105)	Set the integration time for the given CIPS camera(s).

E.599 cips py_cam_tdi_rows**Description**

Last commanded CIPS +Y camera TDI row count setting.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

camera fpga

Telemetry Source Packets**cips sw_status** (514)

Related Commands

<i>command</i>	<i>description</i>
set cips tdi_row.ct (108)	Set the number of pixel rows over which to perform Time Delayed Integration.

E.600 cips py_cam_tdi_time**Description**

Last commanded CIPS +Y camera TDI time setting.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

camera fpga

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips tdi_tm (109)	Set the amount of time Time Delayed Integration will be performed per pixel row.

E.601 cips py_ccd_temp**Description**

CIPS +Y camera CCD temperature.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Limits

alarm	limit
red high	45
yellow high	5
yellow low	-35
red low	-40
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.602 cips py_dc_status_pri**Description**

CIPS +Y camera dust cover status, primary circuit.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OPEN	0	CAUTION
CLOSED	1	GOOD

Discussion

This telemetry point gives one indication of the dust cover's status, with the other indication given by the related telemetry item. These are two independent observations of the dust cover state, so if they disagree, troubleshooting must be done by CIPS hardware engineers to determine the true state of the CIPS dust cover.

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
open cips dust_cover (86)	Deploy the dust cover protecting the given camera.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips py_dc_status_sec (418)	CIPS +Y camera dust cover status indicator, secondary circuit.

E.603 cips py_dc_status_sec**Description**

CIPS +Y camera dust cover status indicator, secondary circuit.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OPEN	0	CAUTION
CLOSED	1	GOOD

Discussion

This telemetry point gives one indication of the dust cover's status, with the other indication given by the related telemetry item. These are two independent observations of the dust cover state, so if they disagree, troubleshooting must be done by CIPS hardware engineers to determine the true state of the CIPS dust cover.

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
open cips dust_cover (86)	Deploy the dust cover protecting the given camera.

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips py_dc_status_pri (417)	CIPS +Y camera dust cover status, primary circuit.

E.604 cips py_ib_of**Description**

Processor FPGA +Y camera image buffer overflow interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets**cips exception (484)****E.605 cips py_ib_of_mk****Description**

Processor FPGA +Y camera image buffer overflow interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets**cips exception (484)****E.606 cips py_img_end****Description**

Processor FPGA +Y camera end of image interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets**cips exception (484)**

E.607 cips py_img_end_mk**Description**

Processor FPGA +Y camera end of image interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.608 cips py_int_temp**Description**

CIPS +Y camera intensifier temperature.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Limits

alarm	limit
red high	45
yellow high	40
yellow low	-35
red low	-40
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.609 cips py_ips_pwr**Description**

CIPS +Y camera high voltage power supply enable.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OFF	0	GOOD
ON	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
turn off cips ips (118)	Turn off power to the given CIPS intensifier power supply.
turn on cips ips (122)	Turn on low voltage power to the given CIPS high voltage power supply.

E.610 cips py_lens_a_temp**Description**

CIPS +Y camera lens temperature at sensor A.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

hk fpga

Analog Conversions

Segment 1

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Limits

alarm	limit
red high	45
yellow high	20
yellow low	-20
red low	-40
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.611 cips py_lens_b_temp**Description**

CIPS +Y camera lens temperature at sensor B.

Data Type

16 bits, unsigned integer, engineering units = C.

Measurement Source

hk fpga

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	65535
c0	-69.3040008544922
c1	0.0214385986328125
c2	-3.86424380849348e-06
c3	4.53405840650234e-10
c4	-2.66966997282663e-14
c5	6.4052632505346e-19
c6	0
c7	0

Limits

alarm	limit
red high	45
yellow high	20
yellow low	-20
red low	-40
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.612 cips py_lens_htr_ena**Description**

CIPS +Y camera lens heater power enable.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
FALSE	0	GOOD
TRUE	1	GOOD

Discussion

Actual application of power for the lens heater is controlled by the CIPS HK FPGA according to the high and low threshold settings (see related telemetry).

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
disable cips lens_htr (52)	Enable the CIPS lens heater for the given camera(s).
enable cips lens_htr (65)	Enable the CIPS lens heater for the given camera(s).

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips i.l_htr_uthresh (254)	Initial (boot) value for the CIPS cameras' lens heater upper threshold.
cips lens_htr_lthresh (267)	Lens heater lower threshold.
cips i.l_htr_lthresh (253)	Initial (boot) value for the CIPS cameras' lens heater lower threshold.
cips lens_htr_uthresh (268)	Lens heater upper threshold.

E.613 cips py_opcode_err

Description

Processor FPGA +Y camera opcode error interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.614 cips py_opcode_err_mk

Description

Processor FPGA +Y camera opcode error interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.615 cips py_parity_err

Description

Processor FPGA +Y camera parity error interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.616 cips py_parity_err_mk**Description**

Processor FPGA +Y camera parity error interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.617 cips py_rx**Description**

Processor FPGA +Y camera receive data ready interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets**cips exception (484)****E.618 cips py_rx_mk****Description**

Processor FPGA +Y camera receive data ready interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets**cips exception (484)****E.619 cips py_seq_err****Description**

Processor FPGA +Y camera row sequence error interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets**cips exception (484)**

E.620 cips py_seq_err_mk**Description**

Processor FPGA +Y camera row sequence error interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
UNMASKED	0	GOOD
MASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.621 cips py_shutr_disable**Description**

CIPS +Y camera shutter disable flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
FALSE	0	GOOD
TRUE	1	GOOD

Discussion

This flag shows whether the CIPS housekeeping FPGA will act on shutter open/close requests from the camera FPGA or ignore them.

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
enable cips shutter (68)	Enable the autonomous opening and closing of the given shutter(s).
disable cips shutter (54)	Disable the autonomous opening and closing of the given shutter(s).

E.622 cips py_shutr_state**Description**

CIPS +Y camera shutter state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
CLOSED	0	GOOD
OPEN	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

E.623 cips quick_boot_flag**Description**

Software flag enabling the automatic transition to STANDBY state after boot.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
LONG	0	GOOD
QUICK	21845	CAUTION

Telemetry Source Packets

cips sw_status (514)

E.624 cips reserved_1_to**Description**

State of downlink enable for CIPS reserved 1 telemetry packets.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
DISABLED	0	GOOD
ENABLED	1	CAUTION

Telemetry Source Packets**cips sw_status** (514)**Related Commands**

<i>command</i>	<i>description</i>
set cips to_mask (110)	Set the CIPS telemetry output enable mask.
disable cips tlm_out (56)	Disable downlink of the given CIPS telemetry packet.
enable cips tlm_out (70)	Enable downlink of the given CIPS telemetry packet.

E.625 cips_reserved_2_to**Description**

State of downlink enable for CIPS reserved 2 telemetry packets.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
DISABLED	0	GOOD
ENABLED	1	CAUTION

Telemetry Source Packets**cips sw_status** (514)**Related Commands**

<i>command</i>	<i>description</i>
set cips to_mask (110)	Set the CIPS telemetry output enable mask.
disable cips tlm_out (56)	Disable downlink of the given CIPS telemetry packet.
enable cips tlm_out (70)	Enable downlink of the given CIPS telemetry packet.

E.626 cips_reserved_3_to**Description**

State of downlink enable for CIPS reserved 3 telemetry packets.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
DISABLED	0	GOOD
ENABLED	1	CAUTION

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips to_mask (110)	Set the CIPS telemetry output enable mask.
disable cips tlm_out (56)	Disable downlink of the given CIPS telemetry packet.
enable cips tlm_out (70)	Enable downlink of the given CIPS telemetry packet.

E.627 cips reserved_4_to**Description**

State of downlink enable for CIPS reserved 4 telemetry packets.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
DISABLED	0	GOOD
ENABLED	1	CAUTION

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips to_mask (110)	Set the CIPS telemetry output enable mask.
disable cips tlm_out (56)	Disable downlink of the given CIPS telemetry packet.
enable cips tlm_out (70)	Enable downlink of the given CIPS telemetry packet.

E.628 cips_reserved_bit

Description

Null (empty) bit. Ignore.

Data Type

1 bits, fill, engineering units = any.

Measurement Source

software

Discussion

Used to 'pad' telemetry packets so that they are an even number of bytes long.

Telemetry Source Packets

This item is contained in many (virtually all) telemetry packets, but is of no real interest.

E.629 cips_row_seq_err_ct

Description

CIPS CAMI FPGA row sequence error count.

Data Type

4 bits, unsigned integer, engineering units = dn.

Measurement Source

cam_i fpga

Limits

alarm	limit
red high	1
yellow high	1
yellow low	-1
red low	-1
delta	<i>none</i>

Telemetry Source Packets

cips_sci_mx_cam (502), cips_sci_my_cam (505), cips_sci_px_cam (508), cips_sci_py_cam (511)

E.630 cips_safe_type

Description

Reason for last entry into CIPS safe hold.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
BOOT	0	GOOD
COMMAND	1	CAUTION
BOS	2	BAD
CMD_LOSS	3	BAD
NONE	4	GOOD

Telemetry Source Packets

cips sw_status (514)

E.631 cips sci_mx_cam_to**Description**

State of downlink enable for CIPS -X camera science telemetry packets.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
DISABLED	0	CAUTION
ENABLED	1	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips to_mask (110)	Set the CIPS telemetry output enable mask.
disable cips tlm_out (56)	Disable downlink of the given CIPS telemetry packet.
enable cips tlm_out (70)	Enable downlink of the given CIPS telemetry packet.

E.632 cips sci_my_cam_to**Description**

State of downlink enable for CIPS -Y camera science telemetry packets.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
DISABLED	0	CAUTION
ENABLED	1	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips to_mask (110)	Set the CIPS telemetry output enable mask.
disable cips tlm_out (56)	Disable downlink of the given CIPS telemetry packet.
enable cips tlm_out (70)	Enable downlink of the given CIPS telemetry packet.

E.633 cips sci_px_cam_to**Description**

State of downlink enable for CIPS +X camera science telemetry packets.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
DISABLED	0	CAUTION
ENABLED	1	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips to_mask (110)	Set the CIPS telemetry output enable mask.
disable cips tlm_out (56)	Disable downlink of the given CIPS telemetry packet.
enable cips tlm_out (70)	Enable downlink of the given CIPS telemetry packet.

E.634 cips sci_py_cam_to

Description

State of downlink enable for CIPS +Y camera science telemetry packets.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
DISABLED	0	CAUTION
ENABLED	1	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
set cips to_mask (110)	Set the CIPS telemetry output enable mask.
disable cips tlm_out (56)	Disable downlink of the given CIPS telemetry packet.
enable cips tlm_out (70)	Enable downlink of the given CIPS telemetry packet.

E.635 cips sdp_enable

Description

Software data processing enable flag.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
FALSE	0	GOOD
TRUE	21845	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
enable cips sdp (66)	Enable CIPS science data processing and downlink.
disable cips sdp (53)	Disable CIPS science data processing and downlink.

E.636 cips sdp_in_progress

Description

Flag showing SDP activity.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
COMPLETE	0	GOOD
IN_PROGRESS	1	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips cur_img_ct (201)	Image number for this camera in the current SDP cycle.

E.637 cips sdram_a_pwr

Description

SDRAM power status, bank A.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OFF	0	CAUTION
ON	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
turn off cips sdram (121)	Disable power to the specified bank(s) of SDRAM memory devices.
turn on cips sdram (125)	Enable power to the specified bank(s) of SDRAM memory devices.

E.638 cips sdram_b_pwr

Description

SDRAM power status, bank B.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
OFF	0	CAUTION
ON	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
turn off cips sdram (121)	Disable power to the specified bank(s) of SDRAM memory devices.
turn on cips sdram (125)	Enable power to the specified bank(s) of SDRAM memory devices.

E.639 cips sdram_chip_remap

Description

Processor FPGA top of image buffer mapping in SDRAM.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
CHIP_0	0	GOOD
CHIP_2	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

E.640 cips sdram i_mon

Description

SDRAM current sense.

Data Type

16 bits, unsigned integer, engineering units = a.

Measurement Source

hk fpga

Analog Conversions

Segment 1

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	3.0517578125e-05
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Limits

alarm	limit
red high	0.100000001490116
yellow high	0.0299999993294477
yellow low	-1
red low	-1
delta	<i>none</i>

Discussion

This value is only updated if the update enable bit is set (see related telemetry).

Telemetry Source Packets

cips hw_hk (492)

Related Commands

command	description
enable cips sdram_ud (67)	Re-enable CIPS SDRAM current monitor telemetry.

Related Telemetry

telemetry	description
cips sdram_update_ena (439)	This bit shows whether or not the SDRAM current measurement is being updated..

E.641 cips_sdram_i_thresh

Description

Threshold for FPGA safing due to SRAM current draw.

Data Type

16 bits, unsigned integer, engineering units = a.

Measurement Source

hk_fpga

Analog Conversions

Segment 1

type	UNSEGMENTED_5D
start	0
stop	65535
c0	0
c1	3.0517578125e-05
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Discussion

The CIPS housekeeping FPGA will automatically turn off the SDRAM chips if they enter an overcurrent condition, with the current threshold defined by this value.

Telemetry Source Packets

cips_hw_hk (492)

E.642 cips_sdram_pixel_map

Description

Processor FPGA mapping mode for images placed in SDRAM.

Data Type

2 bits, discrete state, engineering units = dn.

Measurement Source

proc_fpga

State Conversions

state	value	desirability
S_768x384	0	GOOD
M_768x768	1	GOOD
L_1024x1024	2	GOOD
SS_2048x2048	3	GOOD

Telemetry Source Packets

cips_hw_hk (492)

E.643 cips_sdram_trip_ct**Description**

Cumulative count of the number of times the SDRAM current sense has tripped.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Limits

alarm	limit
red high	-1
yellow high	-1
yellow low	-1
red low	-1
delta	0

Telemetry Source Packets

cips_sw_status (514)

E.644 cips_sdram_update_ena**Description**

This bit shows whether or not the SDRAM current measurement is being updated..

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk_fpga

State Conversions

state	value	desirability
FALSE	0	BAD
TRUE	1	GOOD

Discussion

The SDRAM current measurement will stop updating if the SDRAM current exceeds the programmed threshold and will only be re-enabled when the SDRAM has been commandded back on (an overcurrent event results in the SDRAM being powered off by the FPGA). A command to turn off the SDRAM will not result in SDRAM current measurement updates being disabled.

Telemetry Source Packets

cips hw_hk (492)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips sdram.i.mon (437)	SDRAM current sense.

E.645 cips shutter_cl_time**Description**

CIPS camera shutter close time.

Data Type

16 bits, unsigned integer, engineering units = ms.

Measurement Source

camera fpga

Discussion

This is the time of the camera shutter closure in number of milliseconds after the cameras received the 'Go take an image' instruction. This time includes any frame sync (delay) time, the time taken for camera clear operations (number of clear counts times the duration of one clear count) plus the actual integration time.

Telemetry Source Packets

cips sci_mx_cam (502), **cips sci_my_cam** (505), **cips sci_px_cam** (508), **cips sci_py_cam** (511)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips img_timestamp.s (263)	Time this image was taken, seconds field.
cips img_timestamp.ss (264)	Time this image was taken, subseconds field.

E.646 cips shutter_disable**Description**

Camera shutter open disable flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

hk fpga

State Conversions

state	value	desirability
TRUE	0	GOOD
FALSE	1	GOOD

Discussion

This flag shows whether the CIPS housekeeping FPGA acted on shutter open/close requests from the camera FPGAs during imaging. Nominally, the cameras send open/close requests to the housekeeping FPGA to control integration times.

Telemetry Source Packets

cips sci_mx_cam (502), cips sci_my_cam (505), cips sci_px_cam (508), cips sci_py_cam (511)

E.647 cips sram_img_1_addr**Description**

Starting address of SRAM Code image 1.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.648 cips sram_img_1_crc**Description**

CRC of SRAM Code image 1.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.649 cips sram_img_1_len**Description**

Length in 16-bit words of SRAM Code image 1.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.650 cips sram_img_1_pg**Description**

Page in memory where SRAM Code image 1 starts.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
O_SRAM_P0	0	BAD
O_SRAM_P1	1	BAD
O_SRAM_P2	2	BAD
O_SRAM_P3	3	BAD
O_SRAM_P4	4	BAD
O_SRAM_P5	5	BAD
O_SRAM_P6	6	BAD
O_SRAM_P7	7	BAD
O_SRAM_P8	8	BAD
O_SRAM_P9	9	BAD
O_SRAM_P10	10	BAD
O_SRAM_P11	11	BAD
O_EEPROM_P12	12	BAD
O_EEPROM_P13	13	BAD
O_EEPROM_P14	14	GOOD
O_EEPROM_P15	15	BAD
IO	128	BAD
I_SRAM	255	BAD

Telemetry Source Packets**cips sw_status** (514)**Related Telemetry**

<i>telemetry</i>	<i>description</i>
cips a_seq_page (141)	Page in memory where the current autonomous sequence resides.
cips a_seq_cur_pos (140)	Address of the current command in the current autonomous sequence.
cips a_seq_length (140)	Length of the current autonomous sequence.

E.651 cips sram_img_2_addr**Description**

Starting address of SRAM Code image 2.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets**cips sw_status** (514)**E.652 cips sram_img_2_crc****Description**

CRC of SRAM Code image 2.

Data Type

16 bits, unsigned integer, engineering units = sec.

Measurement Source

software

Telemetry Source Packets**cips sw_status** (514)**E.653 cips sram_img_2_len****Description**

Length in 16-bit words of SRAM Code image 2.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.654 cips sram_img_2_pg**Description**

Page in memory where SRAM Code image 2 starts.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
O_SRAM_P0	0	BAD
O_SRAM_P1	1	BAD
O_SRAM_P2	2	BAD
O_SRAM_P3	3	BAD
O_SRAM_P4	4	BAD
O_SRAM_P5	5	BAD
O_SRAM_P6	6	BAD
O_SRAM_P7	7	BAD
O_SRAM_P8	8	BAD
O_SRAM_P9	9	BAD
O_SRAM_P10	10	BAD
O_SRAM_P11	11	BAD
O_EEPROM_P12	12	GOOD
O_EEPROM_P13	13	BAD
O_EEPROM_P14	14	BAD
O_EEPROM_P15	15	BAD
IO	128	BAD
I_SRAM	255	BAD

Telemetry Source Packets

cips sw_status (514)

Related Telemetry

<i>telemetry</i>	<i>description</i>
cips a_seq_page (141)	Page in memory where the current autonomous sequence resides.
cips a_seq_cur_pos (140)	Address of the current command in the current autonomous sequence.
cips a_seq_length (140)	Length of the current autonomous sequence.

E.655 cips sram_img_to_ld**Description**

Flag that tells the SuROM which SRAM Code image to load on boot.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
IMAGE_1	0	GOOD
IMAGE_2	21845	GOOD

Telemetry Source Packets

cips sw_status (514)

E.656 cips srt_bdrcv**Description**

SuMMIT chip broadcast command received interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.657 cips srt_bdrcv_mk**Description**

SuMMIT chip broadcast command received interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
MASKED	0	GOOD
UNMASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.658 cips srt_bitf**Description**

SuMMIT chip built-in test fail interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.659 cips srt_bitf_mk**Description**

SuMMIT chip built-in test fail interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
MASKED	0	GOOD
UNMASKED	1	GOOD

Telemetry Source Packets**cips exception (484)****E.660 cips srt_brx_mc_ill****Description**

SuMMIT chip broadcast receive mode code illegalization flags.

Data Type

32 bits, unsigned integer, engineering units = dn.

Measurement Source

summit chip

Limits

alarm	limit
red high	0
yellow high	0
yellow low	-2
red low	-2
delta	<i>none</i>

Telemetry Source Packets**cips hw_hk (492)****E.661 cips srt_brx_sa_ill****Description**

SuMMIT chip broadcast receive subaddress illegalization flags.

Data Type

32 bits, unsigned integer, engineering units = dn.

Measurement Source

summit chip

Limits

alarm	limit
red high	0
yellow high	0
yellow low	-2
red low	-2
delta	<i>none</i>

Telemetry Source Packets**cips hw_hk (492)**

E.662 cips srt_btx_mc_ill

Description

SuMMIT chip broadcast transmit mode code illegalization flags.

Data Type

32 bits, unsigned integer, engineering units = dn.

Measurement Source

summit chip

Limits

alarm	limit
red high	0
yellow high	0
yellow low	-2
red low	-2
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.663 cips srt_btx_sa_ill

Description

SuMMIT chip broadcast transmit subaddress illegalization flags.

Data Type

32 bits, unsigned integer, engineering units = dn.

Measurement Source

summit chip

Limits

alarm	limit
red high	0
yellow high	0
yellow low	-2
red low	-2
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.664 cips srt_cr_bcen**Description**

SuMMIT chip broadcast enable flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
DISABLED	0	GOOD
ENABLED	1	CAUTION

Telemetry Source Packets

cips hw_hk (492)

E.665 cips srt_cr_bufc**Description**

SuMMIT chip buffer mode enable flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
DISABLED	0	CAUTION
ENABLED	1	GOOD

Discussion

This buffer mode is NOT the circular buffer mode, but rather the 32-word DMA 'burst' buffer mode.

Telemetry Source Packets

cips hw_hk (492)

E.666 cips srt_cr_chaen**Description**

SuMMIT chip channel A enable flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
DISABLED	0	BAD
ENABLED	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

E.667 cips srt_cr_chben**Description**

SuMMIT chip channel B enable flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
DISABLED	0	BAD
ENABLED	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

E.668 cips srt_cr_dynbc**Description**

SuMMIT chip dynamic bus control acceptance flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
DISABLED	0	GOOD
ENABLED	1	BAD

Telemetry Source Packets

cips hw_hk (492)

E.669 cips srt_cr_etce**Description**

SuMMIT chip external timer clock enable flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
DISABLED	0	GOOD
ENABLED	1	CAUTION

Telemetry Source Packets

cips hw_hk (492)

E.670 cips srt_cr_inten**Description**

SuMMIT chip interrupt logging enable flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
DISABLED	0	BAD
ENABLED	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

E.671 cips srt_cr_ppen**Description**

SuMMIT chip ping-pong mode enable flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
DISABLED	0	GOOD
ENABLED	1	BAD

Telemetry Source Packets

cips hw_hk (492)

E.672 cips srt_cr_stex**Description**

SuMMIT chip execution enable flag.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
DISABLED	0	BAD
ENABLED	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

E.673 cips srt_desc_ptr**Description**

SuMMIT chip descriptor pointer.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

summit chip

Limits

alarm	limit
red high	32769
yellow high	32769
yellow low	32767
red low	32767
delta	0

Telemetry Source Packets

cips hw_hk (492)

E.674 cips srt_dmaf**Description**

SuMMIT chip DMA fail interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.675 cips srt_dmaf_mk**Description**

SuMMIT chip DMA fail interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
MASKED	0	GOOD
UNMASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.676 cips srt_ill_ptr**Description**

SuMMIT chip interrupt log list pointer.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

summit chip

Limits

alarm	limit
red high	33312
yellow high	33312
yellow low	33279
red low	33279
delta	33

Telemetry Source Packets

cips hw_hk (492)

E.677 cips srt_illcmd**Description**

SuMMIT chip illegal command interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.678 cips srt_illcmd_mk**Description**

SuMMIT chip illegal command interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
MASKED	0	GOOD
UNMASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.679 cips srt_ixeq0**Description**

SuMMIT chip index equal zero command interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.680 cips srt_ixeq0_mk**Description**

SuMMIT chip index equal zero command interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
MASKED	0	GOOD
UNMASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.681 cips srt_merr**Description**

SuMMIT chip message error interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.682 cips srt_merr_mk**Description**

SuMMIT chip message error interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
MASKED	0	GOOD
UNMASKED	1	GOOD

Telemetry Source Packets

cips_exception (484)

E.683 cips_srt_rx_mc_ill**Description**

SuMMIT chip receive mode code illegalization flags.

Data Type

32 bits, unsigned integer, engineering units = dn.

Measurement Source

summit chip

Limits

alarm	limit
red high	0
yellow high	0
yellow low	-2
red low	-2
delta	none

Telemetry Source Packets

cips_hw_hk (492)

E.684 cips_srt_rx_sa_ill**Description**

SuMMIT chip receive subaddress illegalization flags.

Data Type

32 bits, unsigned integer, engineering units = dn.

Measurement Source

summit chip

Limits

alarm	limit
red high	-31
yellow high	-31
yellow low	-33
red low	-33
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.685 cips srt_subad**Description**

SuMMIT chip subaddress accessed interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.686 cips srt_subad_mk**Description**

SuMMIT chip subaddress accessed interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
MASKED	0	GOOD
UNMASKED	1	GOOD

Telemetry Source Packets**cips exception (484)****E.687 cips srt_tapf****Description**

SuMMIT chip terminal address parity fail interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets**cips exception (484)****E.688 cips srt_tapf_mk****Description**

SuMMIT chip terminal address parity fail interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
MASKED	0	GOOD
UNMASKED	1	GOOD

Telemetry Source Packets**cips exception (484)**

E.689 cips srt_tx_mc_ill**Description**

SuMMIT chip transmit mode code illegalization flags.

Data Type

32 bits, unsigned integer, engineering units = dn.

Measurement Source

summit chip

Limits

alarm	limit
red high	-54
yellow high	-54
yellow low	-56
red low	-56
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.690 cips srt_tx_sa_ill**Description**

SuMMIT chip transmit subaddress illegalization flags.

Data Type

32 bits, unsigned integer, engineering units = dn.

Measurement Source

summit chip

Limits

alarm	limit
red high	-120
yellow high	-120
yellow low	-122
red low	-122
delta	<i>none</i>

Telemetry Source Packets

cips hw_hk (492)

E.691 cips srt_wrapf**Description**

SuMMIT chip wrap fail interrupt state.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
INACTIVE	0	GOOD
ACTIVE	1	GOOD

Telemetry Source Packets

cips exception (484)

E.692 cips srt_wrapf_mk**Description**

SuMMIT chip wrap fail interrupt mask.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

summit chip

State Conversions

state	value	desirability
MASKED	0	GOOD
UNMASKED	1	GOOD

Telemetry Source Packets

cips exception (484)

E.693 cips stack_util_max**Description**

Maximum CIPS stack memory utilization measured since last reset/reboot.

Data Type

16 bits, unsigned integer, engineering units = pc.

Measurement Source

software

Analog Conversions**Segment 1**

type	UNSEGMENTED_5D
start	0
stop	0
c0	0
c1	0.00305185094475746
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Telemetry Source Packets

cips sw_status (514)

E.694 cips start_pixel**Description**

Position of first pixel in image data within this packet.

Data Type

32 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sci_mx_cam (502), cips sci_my_cam (505), cips sci_px_cam (508), cips sci_py_cam (511)

E.695 cips sw_bin_ena**Description**

Describes if software binning was enabled for this image.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
DIS	0	GOOD
ENA	1	GOOD

Telemetry Source Packets

cips sci_mx_cam (502), cips sci_my_cam (505), cips sci_px_cam (508), cips sci_py_cam (511)

E.696 cips sw_bin_enable**Description**

Software binning enable flag.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
FALSE	0	GOOD
TRUE	21845	GOOD

Telemetry Source Packets

cips sw_status (514)

Related Commands

<i>command</i>	<i>description</i>
enable cips sw_bin (69)	Enable science data binning capability in CIPS FSW.
disable cips sw_bin (55)	Disable science data binning capability in CIPS FSW.

E.697 cips sw_comp_enable**Description**

Software compression enable flag.

Data Type

16 bits, discrete state, engineering units = dn.

Measurement Source

software

State Conversions

state	value	desirability
FALSE	0	GOOD
TRUE	21845	GOOD

Telemetry Source Packets**cips sw_status** (514)**Related Commands**

<i>command</i>	<i>description</i>
disable cips comp (49)	Disable science data compression capability in CIPS FSW.
enable cips comp (61)	Enable science data compression capability in CIPS FSW.

E.698 cips sw_stat_rt**Description**

Packet production period for CIPS software status packets.

Data Type

16 bits, unsigned integer, engineering units = sec.

Measurement Source

software

Telemetry Source Packets**cips sw_status** (514)**E.699 cips sw_stat_to****Description**

State of downlink enable for CIPS software status telemetry packets.

Data Type

1 bits, discrete state, engineering units = any.

Measurement Source

software

State Conversions

state	value	desirability
DISABLED	0	CAUTION
ENABLED	1	GOOD

Telemetry Source Packets**cips sw_status** (514)

Related Commands

<i>command</i>	<i>description</i>
set cips to_mask (110)	Set the CIPS telemetry output enable mask.
disable cips tlm_out (56)	Disable downlink of the given CIPS telemetry packet.
enable cips tlm_out (70)	Enable downlink of the given CIPS telemetry packet.

E.700 cips tdi_dir**Description**

CIPS camera TDI direction setting.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

camera fpga

State Conversions

state	value	desirability
TO_A	0	GOOD
TO_B	1	GOOD

Telemetry Source Packets

cips sci_mx_cam (502), cips sci_my_cam (505), cips sci_px_cam (508), cips sci_py_cam (511)

E.701 cips tdi_dwell**Description**

CIPS camera TDI dwell time.

Data Type

10 bits, unsigned integer, engineering units = ms.

Measurement Source

camera fpga

Analog Conversions

Segment 1

type	UNSEGMENTED_5D
start	0
stop	0
c0	0
c1	0.100000001490116
c2	0
c3	0
c4	0
c5	0
c6	0
c7	0

Telemetry Source Packets

cips sci_mx_cam (502), cips sci_my_cam (505), cips sci_px_cam (508), cips sci_py_cam (511)

E.702 cips tdi_rows**Description**

TDI row count setting for this image.

Data Type

8 bits, unsigned integer, engineering units = dn.

Measurement Source

camera fpga

Telemetry Source Packets

cips sci_mx_cam (502), cips sci_my_cam (505), cips sci_px_cam (508), cips sci_py_cam (511)

E.703 cips test_cam_select**Description**

Flag showing which camera's data is routed to the GSE test port.

Data Type

2 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
PX	0	GOOD
PY	1	GOOD
MX	2	GOOD
MY	3	GOOD

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
select cips gse_ccd (93)	Selects the CCD whose data output is streamed to the GSE port.

E.704 cips tlm_pkts_per_blk**Description**

Maximum number of CCSDS telemetry packets to write per 1024-byte block transfer.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.705 cips tlm_rbuf_wds**Description**

Number of words of data currently in the telemetry ring buffer.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.706 cips tlm_words_sent**Description**

Cumulative count of the number of words downlinked in telemetry.

Data Type

32 bits, unsigned integer, engineering units = any.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

E.707 cips uart_ab_sel**Description**

CDE UART channel selection status.

Data Type

1 bits, discrete state, engineering units = dn.

Measurement Source

proc fpga

State Conversions

state	value	desirability
B	0	GOOD
A	1	GOOD

Telemetry Source Packets

cips hw_hk (492)

Related Commands

<i>command</i>	<i>description</i>
select cips cde_chnl (92)	Select the channel CIPS uses to communicate with CDE.

E.708 cips uart_fifo_level**Description**

Number of words received on the CDE UART before interrupting CIPS FSW.

Data Type

4 bits, unsigned integer, engineering units = dn.

Measurement Source

proc fpga

Telemetry Source Packets

cips hw_hk (492)

E.709 cips ut69r_acc**Description**

Value contained in the UTMC 69R000 microcontroller accumulator register.

Data Type

32 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Telemetry Source Packets

cips exception (484)

E.710 cips ut69r_ft**Description**

Value contained in the UTMC 69R000 microcontroller fault register.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Telemetry Source Packets

cips exception (484)

E.711 cips ut69r_ics**Description**

Value in the UTMC 69R000 microcontroller instruction counter save register.

Data Type

32 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Telemetry Source Packets**cips exception** (484)**E.712 cips ut69r_mk****Description**

Value contained in the UTMIC 69R000 microcontroller interrupt mask register.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Telemetry Source Packets**cips exception** (484)**E.713 cips ut69r_pi****Description**

Value contained in the UTMIC 69R000 microcontroller pending interrupt register.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Telemetry Source Packets**cips exception** (484)**E.714 cips ut69r_r10****Description**

Value contained in the UTMIC 69R000 microcontroller register 10.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Telemetry Source Packets**cips exception** (484)

E.715 cips ut69r_r11**Description**

Value contained in the UTMC 69R000 microcontroller register 11.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Telemetry Source Packets

cips exception (484)

E.716 cips ut69r_r12**Description**

Value contained in the UTMC 69R000 microcontroller register 12.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Telemetry Source Packets

cips exception (484)

E.717 cips ut69r_r13**Description**

Value contained in the UTMC 69R000 microcontroller register 13.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Telemetry Source Packets

cips exception (484)

E.718 cips ut69r_r14**Description**

Value contained in the UTMC 69R000 microcontroller register 14.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Discussion

The value in this register should be the stack pointer at the entry to the exception ISR.

Telemetry Source Packets

cips exception (484)

E.719 cips ut69r_r15**Description**

Value contained in the UTMC 69R000 microcontroller register 15.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Telemetry Source Packets

cips exception (484)

E.720 cips ut69r_r16**Description**

Value contained in the UTMC 69R000 microcontroller register 16.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Discussion

The value in this register should be zero.

Telemetry Source Packets**cips exception** (484)**E.721 cips ut69r_r17****Description**

Value contained in the UTMIC 69R000 microcontroller register 17.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Discussion

The value in this register should be the address of the instruction immediately after the one where the interrupt occurred.

Telemetry Source Packets**cips exception** (484)**E.722 cips ut69r_r2****Description**

Value contained in the UTMIC 69R000 microcontroller register 2.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Discussion

The value in this register should be pInfo.

Telemetry Source Packets**cips exception** (484)**E.723 cips ut69r_r3****Description**

Value contained in the UTMIC 69R000 microcontroller register 3.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Discussion

The value in this register should be the page number in use at the time the exception occurred.

Telemetry Source Packets

cips exception (484)

E.724 cips ut69r_r4**Description**

Value contained in the UTM 69R000 microcontroller register 4.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Discussion

The value in this register should be the value of the UT69R000 status register at the time the exception occurred.

Telemetry Source Packets

cips exception (484)

E.725 cips ut69r_r5**Description**

Value contained in the UTM 69R000 microcontroller register 5.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Telemetry Source Packets

cips exception (484)

E.726 cips ut69r_r6**Description**

Value contained in the UTMC 69R000 microcontroller register 6.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Telemetry Source Packets

cips exception (484)

E.727 cips ut69r_r7**Description**

Value contained in the UTMC 69R000 microcontroller register 7.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Telemetry Source Packets

cips exception (484)

E.728 cips ut69r_r8**Description**

Value contained in the UTMC 69R000 microcontroller register 8.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Telemetry Source Packets

cips exception (484)

E.729 cips ut69r_r9**Description**

Value contained in the UTMC 69R000 microcontroller register 9.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Telemetry Source Packets

cips exception (484)

E.730 cips ut69r_rcvr**Description**

Value contained in the UTMC 69R000 microcontroller GSE UART receive register.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Telemetry Source Packets

cips exception (484)

E.731 cips ut69r_status**Description**

Value contained in the UTMC 69R000 microcontroller system status register.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Telemetry Source Packets

cips exception (484)

E.732 cips ut69r_sw**Description**

Value contained in the UTMC 69R000 microcontroller status register.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Telemetry Source Packets

cips exception (484)

E.733 cips ut69r_ta**Description**

Value contained in the UTMC 69R000 microcontroller timer A register.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Telemetry Source Packets

cips exception (484)

E.734 cips ut69r_tb**Description**

Value contained in the UTMC 69R000 microcontroller timer B register.

Data Type

16 bits, unsigned integer, engineering units = any.

Measurement Source

69R000

Telemetry Source Packets

cips exception (484)

E.735 cips wind_sync_err_ct**Description**

Cumulative count of window synchronization errors encountered.

Data Type

16 bits, unsigned integer, engineering units = dn.

Measurement Source

software

Telemetry Source Packets

cips sw_status (514)

F

Telemetry Packet List

OASIS-CC/FSW database version TBD, Mon Feb 19 11:16:31 2007.

F.1 cips cmd_echo

Packet Source Application Identifier

0x167 (359)

OASIS Stream Name

cips359

Description

cips cmd echo packet.

Discussion

CIPS cmd echo packet.

Default Packet Production Status: Output of this packet is nominally disabled. When enabled, packets are produced asynchronously on command receipt.

Format

Packet **cips cmd_echo** Format

start bit	stop bit	size bits	field	control	method
0	15	16	cips p359pid (362)	—	—
16	17	2	cips p359flg (360)	—	—
18	31	14	cips p359sct (362)	—	—
32	47	16	cips p359pl (362)	—	—
48	63	16	cips p359hws (360)	—	—
64	79	16	cips p359lws (361)	—	—
80	95	16	cips p359hwss (361)	—	—
96	111	16	cips p359lwss (361)	—	—
112	127	16	cips pkt_version (379)	—	—
128	130	3	cips ce_version (169)	—	—
131	131	1	cips ce_type (169)	—	—
132	132	1	cips ce_sec_hdr (168)	—	—
133	143	11	cips ce_apid (166)	—	—
144	145	2	cips ce_seq_flags (168)	—	—
146	159	14	cips ce_seq_ct (168)	—	—
160	175	16	cips ce_pkt_len (167)	—	—
176	176	1	cips reserved_bit (431)	—	—
177	183	7	cips ce_func_code (167)	—	—
184	191	8	cips ce_chksum (166)	—	—
192	199	8	cips ce_opcode (167)	—	—
200	2175	1976	cips ce_cmd_data (166)	—	—
2176	2191	16	cips pkt_crc (378)	—	—

Packet size = 2192 bits (274 bytes) .

F.2 cips err

Packet Source Application Identifier

0x162 (354)

OASIS Stream Name

cips354

Description

cips error packet.

Discussion

CIPS error packet.

Default Packet Production Status: Packets produced asynchronously on error detection.

Format

Packet **cips err** Format

start bit	stop bit	size bits	field	control	method
0	15	16	cips p354pid (348)	—	—
16	17	2	cips p354flg (347)	—	—
18	31	14	cips p354sct (349)	—	—
32	47	16	cips p354pl (349)	—	—
48	63	16	cips p354hws (347)	—	—
64	79	16	cips p354lws (348)	—	—
80	95	16	cips p354hwss (347)	—	—
96	111	16	cips p354lwss (348)	—	—
112	127	16	cips pkt.version (379)	—	—
128	8175	8048	cips pkt.filler (378)	—	—
8176	8191	16	cips pkt.crc (378)	—	—

Packet size = 8192 bits (1024 bytes) .

F.3 cips evt_log

Packet Source Application Identifier

0x163 (355)

OASIS Stream Name

cips355

Description

cips event log packet.

Discussion

CIPS event log packet.

Default Packet Production Status: Packets produced asynchronously on command or packet completion.

Related Commands

<i>command</i>	<i>description</i>
enable cips em (63)	Enable given messages in the event log packet.
disable cips em (51)	Disable given messages in the event log packet.
set cips em_mask (99)	Set the software event logging mask.

Format

Packet **cips evt_log** Format

start bit	stop bit	size bits	field	control	method
0	15	16	cips p355pid (351)	—	—
16	17	2	cips p355flg (349)	—	—
18	31	14	cips p355sct (352)	—	—
32	47	16	cips p355pl (351)	—	—
48	63	16	cips p355hws (350)	—	—
64	79	16	cips p355lws (350)	—	—
80	95	16	cips p355hwss (350)	—	—
96	111	16	cips p355lwss (351)	—	—
112	119	8	cips csci (199)	—	—
120	127	8	cips pkt_version_el (379)	—	—
128	143	16	cips event_ct (234)	—	—
144	143	0	cips event_info (235)	cips event_ct (234)	COUNT
144	159	16	cips pkt_crc (378)	—	—

Packet size = 160 bits (20 bytes) plus N*P, where N is the number of subpackets in the packet and P is the size of the subpacket (which, of course, may need to be calculated based on sub-subpacket size). .

Event Info Format

Event Info Format

start bit	stop bit	size bits	field	control	method
0	31	32	cips event_time_s (238)	—	—
32	63	32	cips event_time_ss (238)	—	—

Event Info Format, cont'd

start bit	stop bit	size bits	field		
64	95	32	cips event_key (235)	—	—
96	111	16	cips event_pc (237)	—	—
112	127	16	cips event_p0 (236)	—	—
128	143	16	cips event_p1 (236)	—	—
144	159	16	cips event_p2 (237)	—	—
160	175	16	cips event_p3 (237)	—	—

Maximum Subpacket size = 176 bits (22 bytes)

F.4 cips exception

Packet Source Application Identifier

0x166 (358)

OASIS Stream Name

cips358

Description

cips exception packet.

Discussion

CIPS exception packet.

Default Packet Production Status: Packets produced asynchronously on a processor reset.

Related Commands

<i>command</i>	<i>description</i>
issue cips exception (80)	Send an exception telemetry packet.

Format

Packet **cips exception** Format

start bit	stop bit	size bits	field	control	method
0	15	16	cips p358pid (359)	—	—
16	17	2	cips p358flg (357)	—	—
18	31	14	cips p358sct (360)	—	—
32	47	16	cips p358pl (359)	—	—
48	63	16	cips p358hws (358)	—	—
64	79	16	cips p358lws (358)	—	—
80	95	16	cips p358hwss (358)	—	—
96	111	16	cips p358lwss (359)	—	—
112	127	16	cips pkt_version (379)	—	—
128	143	16	cips ut69r_r2 (473)	—	—
144	159	16	cips ut69r_r3 (473)	—	—
160	175	16	cips ut69r_r4 (474)	—	—
176	191	16	cips ut69r_r5 (474)	—	—
192	207	16	cips ut69r_r6 (475)	—	—
208	223	16	cips ut69r_r7 (475)	—	—
224	239	16	cips ut69r_r8 (475)	—	—
240	255	16	cips ut69r_r9 (476)	—	—
256	271	16	cips ut69r_r10 (470)	—	—
272	287	16	cips ut69r_r11 (471)	—	—
288	303	16	cips ut69r_r12 (471)	—	—
304	319	16	cips ut69r_r13 (471)	—	—
320	335	16	cips ut69r_r14 (472)	—	—
336	351	16	cips ut69r_r15 (472)	—	—
352	367	16	cips ut69r_r16 (472)	—	—
368	383	16	cips ut69r_r17 (473)	—	—
384	399	16	cips ut69r_status (476)	—	—

Packet **cips exception** Format, cont'd

start bit	stop bit	size bits	field		
400	431	32	cips ut69r_ics (469)	—	—
432	447	16	cips ut69r_rcvr (476)	—	—
448	463	16	cips ut69r_ta (477)	—	—
464	479	16	cips ut69r_tb (477)	—	—
480	511	32	cips ut69r_acc (469)	—	—
512	527	16	cips ut69r_sw (477)	—	—
528	543	16	cips ut69r_ft (469)	—	—
544	559	16	cips ut69r_mk (470)	—	—
560	575	16	cips ut69r_pi (470)	—	—
576	591	16	cips e_proc_fpga_ver (212)	—	—
592	607	16	cips e_cami_fpga_ver (207)	—	—
608	623	16	cips e_hk_fpga_ver (212)	—	—
624	624	1	cips reserved_bit (431)	—	—
625	625	1	cips reserved_bit (431)	—	—
626	626	1	cips reserved_bit (431)	—	—
627	627	1	cips reserved_bit (431)	—	—
628	628	1	cips reserved_bit (431)	—	—
629	629	1	cips reserved_bit (431)	—	—
630	630	1	cips reserved_bit (431)	—	—
631	631	1	cips reserved_bit (431)	—	—
632	632	1	cips hk_irq_1s (242)	—	—
633	633	1	cips hk_irq_lppwr (244)	—	—
634	634	1	cips hk_irq_lpevt (243)	—	—
635	635	1	cips hk_irq_sdrami (248)	—	—
636	636	1	cips hk_irq_my_bos (246)	—	—
637	637	1	cips hk_irq_mx_bos (245)	—	—
638	638	1	cips hk_irq_py_bos (247)	—	—
639	639	1	cips hk_irq_px_bos (247)	—	—
640	640	1	cips reserved_bit (431)	—	—
641	641	1	cips reserved_bit (431)	—	—
642	642	1	cips reserved_bit (431)	—	—
643	643	1	cips reserved_bit (431)	—	—
644	644	1	cips reserved_bit (431)	—	—
645	645	1	cips reserved_bit (431)	—	—
646	646	1	cips reserved_bit (431)	—	—
647	647	1	cips reserved_bit (431)	—	—
648	648	1	cips hk_irq_1s_mk (243)	—	—
649	649	1	cips hk_irq_lppwr_mk (244)	—	—
650	650	1	cips hk_irq_lpevt_mk (244)	—	—
651	651	1	cips hk_irq_sdrami_mk (249)	—	—
652	652	1	cips hk_irq_my_bos_mk (246)	—	—
653	653	1	cips hk_irq_mx_bos_mk (245)	—	—
654	654	1	cips hk_irq_py_bos_mk (248)	—	—
655	655	1	cips hk_irq_px_bos_mk (247)	—	—
656	656	1	cips reserved_bit (431)	—	—
657	657	1	cips reserved_bit (431)	—	—
658	658	1	cips reserved_bit (431)	—	—

Packet **cips exception** Format, cont'd

start bit	stop bit	size bits	field		
659	659	1	cips reserved_bit (431)	—	—
660	660	1	cips reserved_bit (431)	—	—
661	661	1	cips reserved_bit (431)	—	—
662	662	1	cips reserved_bit (431)	—	—
663	663	1	cips reserved_bit (431)	—	—
664	664	1	cips reserved_bit (431)	—	—
665	665	1	cips reserved_bit (431)	—	—
666	666	1	cips reserved_bit (431)	—	—
667	667	1	cips reserved_bit (431)	—	—
668	668	1	cips reserved_bit (431)	—	—
669	669	1	cips reserved_bit (431)	—	—
670	670	1	cips proc_me_mprot (385)	—	—
671	671	1	cips proc_me_bterr (384)	—	—
672	672	1	cips reserved_bit (431)	—	—
673	673	1	cips reserved_bit (431)	—	—
674	674	1	cips reserved_bit (431)	—	—
675	675	1	cips reserved_bit (431)	—	—
676	676	1	cips reserved_bit (431)	—	—
677	677	1	cips reserved_bit (431)	—	—
678	678	1	cips reserved_bit (431)	—	—
679	679	1	cips reserved_bit (431)	—	—
680	680	1	cips reserved_bit (431)	—	—
681	681	1	cips reserved_bit (431)	—	—
682	682	1	cips px_ib_of (398)	—	—
683	683	1	cips px_img_end (399)	—	—
684	684	1	cips px_seq_err (407)	—	—
685	685	1	cips px_opcode_err (404)	—	—
686	686	1	cips px_parity_err (404)	—	—
687	687	1	cips px_rx (406)	—	—
688	688	1	cips reserved_bit (431)	—	—
689	689	1	cips reserved_bit (431)	—	—
690	690	1	cips reserved_bit (431)	—	—
691	691	1	cips reserved_bit (431)	—	—
692	692	1	cips reserved_bit (431)	—	—
693	693	1	cips reserved_bit (431)	—	—
694	694	1	cips reserved_bit (431)	—	—
695	695	1	cips reserved_bit (431)	—	—
696	696	1	cips reserved_bit (431)	—	—
697	697	1	cips reserved_bit (431)	—	—
698	698	1	cips px_ib_of_mk (399)	—	—
699	699	1	cips px_img_end_mk (400)	—	—
700	700	1	cips px_seq_err_mk (407)	—	—
701	701	1	cips px_opcode_err_mk (404)	—	—
702	702	1	cips px_parity_err_mk (405)	—	—
703	703	1	cips px_rx_mk (407)	—	—
704	704	1	cips reserved_bit (431)	—	—
705	705	1	cips reserved_bit (431)	—	—

Packet **cips exception** Format, cont'd

start bit	stop bit	size bits	field		
706	706	1	cips reserved_bit (431)	—	—
707	707	1	cips reserved_bit (431)	—	—
708	708	1	cips reserved_bit (431)	—	—
709	709	1	cips reserved_bit (431)	—	—
710	710	1	cips reserved_bit (431)	—	—
711	711	1	cips reserved_bit (431)	—	—
712	712	1	cips reserved_bit (431)	—	—
713	713	1	cips reserved_bit (431)	—	—
714	714	1	cips py_ib_of (418)	—	—
715	715	1	cips py_img_end (419)	—	—
716	716	1	cips py_seq_err (426)	—	—
717	717	1	cips py_opcode_err (424)	—	—
718	718	1	cips py_parity_err (424)	—	—
719	719	1	cips py_rx (425)	—	—
720	720	1	cips reserved_bit (431)	—	—
721	721	1	cips reserved_bit (431)	—	—
722	722	1	cips reserved_bit (431)	—	—
723	723	1	cips reserved_bit (431)	—	—
724	724	1	cips reserved_bit (431)	—	—
725	725	1	cips reserved_bit (431)	—	—
726	726	1	cips reserved_bit (431)	—	—
727	727	1	cips reserved_bit (431)	—	—
728	728	1	cips reserved_bit (431)	—	—
729	729	1	cips reserved_bit (431)	—	—
730	730	1	cips py_ib_of_mk (419)	—	—
731	731	1	cips py_img_end_mk (420)	—	—
732	732	1	cips py_seq_err_mk (427)	—	—
733	733	1	cips py_opcode_err_mk (424)	—	—
734	734	1	cips py_parity_err_mk (425)	—	—
735	735	1	cips py_rx_mk (426)	—	—
736	736	1	cips reserved_bit (431)	—	—
737	737	1	cips reserved_bit (431)	—	—
738	738	1	cips reserved_bit (431)	—	—
739	739	1	cips reserved_bit (431)	—	—
740	740	1	cips reserved_bit (431)	—	—
741	741	1	cips reserved_bit (431)	—	—
742	742	1	cips reserved_bit (431)	—	—
743	743	1	cips reserved_bit (431)	—	—
744	744	1	cips reserved_bit (431)	—	—
745	745	1	cips reserved_bit (431)	—	—
746	746	1	cips mx_ib_of (300)	—	—
747	747	1	cips mx_img_end (301)	—	—
748	748	1	cips mx_seq_err (309)	—	—
749	749	1	cips mx_opcode_err (306)	—	—
750	750	1	cips mx_parity_err (307)	—	—
751	751	1	cips mx_rx (308)	—	—
752	752	1	cips reserved_bit (431)	—	—

Packet **cips exception** Format, cont'd

start bit	stop bit	size bits	field		
753	753	1	cips reserved_bit (431)	—	—
754	754	1	cips reserved_bit (431)	—	—
755	755	1	cips reserved_bit (431)	—	—
756	756	1	cips reserved_bit (431)	—	—
757	757	1	cips reserved_bit (431)	—	—
758	758	1	cips reserved_bit (431)	—	—
759	759	1	cips reserved_bit (431)	—	—
760	760	1	cips reserved_bit (431)	—	—
761	761	1	cips reserved_bit (431)	—	—
762	762	1	cips mx_ib_of_mk (301)	—	—
763	763	1	cips mx_img_end_mk (302)	—	—
764	764	1	cips mx_seq_err_mk (309)	—	—
765	765	1	cips mx_opcode_err_mk (307)	—	—
766	766	1	cips mx_parity_err_mk (308)	—	—
767	767	1	cips mx_rx_mk (308)	—	—
768	768	1	cips reserved_bit (431)	—	—
769	769	1	cips reserved_bit (431)	—	—
770	770	1	cips reserved_bit (431)	—	—
771	771	1	cips reserved_bit (431)	—	—
772	772	1	cips reserved_bit (431)	—	—
773	773	1	cips reserved_bit (431)	—	—
774	774	1	cips reserved_bit (431)	—	—
775	775	1	cips reserved_bit (431)	—	—
776	776	1	cips reserved_bit (431)	—	—
777	777	1	cips reserved_bit (431)	—	—
778	778	1	cips my_ib_of (320)	—	—
779	779	1	cips my_img_end (321)	—	—
780	780	1	cips my_seq_err (328)	—	—
781	781	1	cips my_opcode_err (325)	—	—
782	782	1	cips my_parity_err (326)	—	—
783	783	1	cips my_rx (327)	—	—
784	784	1	cips reserved_bit (431)	—	—
785	785	1	cips reserved_bit (431)	—	—
786	786	1	cips reserved_bit (431)	—	—
787	787	1	cips reserved_bit (431)	—	—
788	788	1	cips reserved_bit (431)	—	—
789	789	1	cips reserved_bit (431)	—	—
790	790	1	cips reserved_bit (431)	—	—
791	791	1	cips reserved_bit (431)	—	—
792	792	1	cips reserved_bit (431)	—	—
793	793	1	cips reserved_bit (431)	—	—
794	794	1	cips my_ib_of_mk (320)	—	—
795	795	1	cips my_img_end_mk (321)	—	—
796	796	1	cips my_seq_err_mk (328)	—	—
797	797	1	cips my_opcode_err_mk (325)	—	—
798	798	1	cips my_parity_err_mk (326)	—	—
799	799	1	cips my_rx_mk (327)	—	—

Packet **cips exception** Format, cont'd

start bit	stop bit	size bits	field		
800	800	1	cips reserved.bit (431)	—	—
801	801	1	cips reserved.bit (431)	—	—
802	802	1	cips e_cde_uart.f_err (207)	—	—
803	803	1	cips e_cde_uart.rx_or (209)	—	—
804	804	1	cips e_cde_uart.rx_af (207)	—	—
805	805	1	cips e_cde_uart.rx.f (208)	—	—
806	806	1	cips e_cde_uart.rx.r (209)	—	—
807	807	1	cips e_cde_uart.rx.e (208)	—	—
808	808	1	cips reserved.bit (431)	—	—
809	809	1	cips reserved.bit (431)	—	—
810	810	1	cips reserved.bit (431)	—	—
811	811	1	cips e_cde_uart.tx.b (210)	—	—
812	812	1	cips e_cde_uart.tx.af (210)	—	—
813	813	1	cips e_cde_uart.tx.f (211)	—	—
814	814	1	cips e_cde_uart.tx.ae (210)	—	—
815	815	1	cips e_cde_uart.tx.e (211)	—	—
816	816	1	cips reserved.bit (431)	—	—
817	817	1	cips reserved.bit (431)	—	—
818	818	1	cips reserved.bit (431)	—	—
819	819	1	cips cde_uart.rxor.mk (163)	—	—
820	820	1	cips reserved.bit (431)	—	—
821	821	1	cips reserved.bit (431)	—	—
822	822	1	cips cde_uart.rx.r.mk (162)	—	—
823	823	1	cips reserved.bit (431)	—	—
824	824	1	cips reserved.bit (431)	—	—
825	825	1	cips reserved.bit (431)	—	—
826	826	1	cips reserved.bit (431)	—	—
827	827	1	cips reserved.bit (431)	—	—
828	828	1	cips reserved.bit (431)	—	—
829	829	1	cips reserved.bit (431)	—	—
830	830	1	cips reserved.bit (431)	—	—
831	831	1	cips cde_uart.tx.e.mk (165)	—	—
832	832	1	cips reserved.bit (431)	—	—
833	833	1	cips reserved.bit (431)	—	—
834	834	1	cips reserved.bit (431)	—	—
835	835	1	cips reserved.bit (431)	—	—
836	836	1	cips reserved.bit (431)	—	—
837	837	1	cips reserved.bit (431)	—	—
838	838	1	cips reserved.bit (431)	—	—
839	839	1	cips reserved.bit (431)	—	—
840	840	1	cips reserved.bit (431)	—	—
841	841	1	cips reserved.bit (431)	—	—
842	842	1	cips reserved.bit (431)	—	—
843	843	1	cips reserved.bit (431)	—	—
844	844	1	cips proc_mode.rset (385)	—	—
845	845	1	cips proc_self.rset (388)	—	—
846	846	1	cips proc.wd.rset (388)	—	—

Packet **cips exception** Format, cont'd

start bit	stop bit	size bits	field		
847	847	1	cips proc_mode (385)	—	—
848	848	1	cips e_srt_cr_stex (217)	—	—
849	849	1	cips reserved.bit (431)	—	—
850	850	1	cips reserved.bit (431)	—	—
851	851	1	cips e_srt_cr_chaen (214)	—	—
852	852	1	cips e_srt_cr_chben (215)	—	—
853	853	1	cips e_srt_cr_etce (216)	—	—
854	854	1	cips reserved.bit (431)	—	—
855	855	1	cips reserved.bit (431)	—	—
856	856	1	cips reserved.bit (431)	—	—
857	857	1	cips e_srt_cr_bufrr (214)	—	—
858	858	1	cips reserved.bit (431)	—	—
859	859	1	cips e_srt_cr_bcen (214)	—	—
860	860	1	cips e_srt_cr_dynbc (215)	—	—
861	861	1	cips e_srt_cr_ppen (217)	—	—
862	862	1	cips e_srt_cr_inten (216)	—	—
863	863	1	cips reserved.bit (431)	—	—
864	864	1	cips srt_dmaf (453)	—	—
865	865	1	cips srt_wrapf (461)	—	—
866	866	1	cips srt_tapf (459)	—	—
867	867	1	cips srt_bitf (446)	—	—
868	868	1	cips srt_merr (456)	—	—
869	869	1	cips srt_subad (458)	—	—
870	870	1	cips srt_bdrvc (445)	—	—
871	871	1	cips srt_ixeq0 (455)	—	—
872	872	1	cips srt_illcmd (454)	—	—
873	873	1	cips reserved.bit (431)	—	—
874	874	1	cips reserved.bit (431)	—	—
875	875	1	cips reserved.bit (431)	—	—
876	876	1	cips reserved.bit (431)	—	—
877	877	1	cips reserved.bit (431)	—	—
878	878	1	cips reserved.bit (431)	—	—
879	879	1	cips reserved.bit (431)	—	—
880	880	1	cips srt_dmaf_mk (453)	—	—
881	881	1	cips srt_wrapf_mk (461)	—	—
882	882	1	cips srt_tapf_mk (459)	—	—
883	883	1	cips srt_bitf_mk (446)	—	—
884	884	1	cips srt_merr_mk (456)	—	—
885	885	1	cips srt_subad_mk (458)	—	—
886	886	1	cips srt_bdrvc_mk (445)	—	—
887	887	1	cips srt_ixeq0_mk (456)	—	—
888	888	1	cips srt_illcmd_mk (455)	—	—
889	889	1	cips reserved.bit (431)	—	—
890	890	1	cips reserved.bit (431)	—	—
891	891	1	cips reserved.bit (431)	—	—
892	892	1	cips reserved.bit (431)	—	—
893	893	1	cips reserved.bit (431)	—	—

Packet **cips exception** Format, cont'd

start bit	stop bit	size bits	field		
894	894	1	cips reserved_bit (431)	—	—
895	895	1	cips reserved_bit (431)	—	—
896	911	16	cips e_srt_ill_ptr (218)	—	—
912	927	16	cips e_srt_desc_ptr (218)	—	—
928	959	32	cips e_srt_rx_sa_ill (219)	—	—
960	991	32	cips e_srt_tx_sa_ill (219)	—	—
992	1023	32	cips e_srt_brx_sa_ill (213)	—	—
1024	1055	32	cips e_srt_btx_sa_ill (213)	—	—
1056	1087	32	cips e_srt_rx_mc_ill (218)	—	—
1088	1119	32	cips e_srt_tx_mc_ill (219)	—	—
1120	1151	32	cips e_srt_brx_mc_ill (212)	—	—
1152	1183	32	cips e_srt_btx_mc_ill (213)	—	—
1184	1199	16	cips pkt_crc (378)	—	—

Packet size = 1200 bits (150 bytes) .

F.5 cips hw_hk

Packet Source Application Identifier

0x161 (353)

OASIS Stream Name

cips353

Description

cips housekeeping packet.

Discussion

CIPS hardware housekeeping packet.

Default Packet Production Status: Packets produced every 60 seconds.

Related Commands

<i>command</i>	<i>description</i>
set cips hw_hk_rt (101)	Set the cips housekeeping telemetry packet period.
issue cips hw_hk (81)	Command the CIPS FSW to issue a hardware housekeeping packet.

Format

Packet **cips hw_hk** Format

start bit	stop bit	size bits	field	control	method
0	15	16	cips p353pid (346)	—	—
16	17	2	cips p353flg (344)	—	—
18	31	14	cips p353sct (346)	—	—
32	47	16	cips p353pl (346)	—	—
48	63	16	cips p353hws (344)	—	—
64	79	16	cips p353lws (345)	—	—
80	95	16	cips p353hwss (345)	—	—
96	111	16	cips p353lwss (345)	—	—
112	127	16	cips pkt_version (379)	—	—
128	143	16	cips proc_fpga_ver (384)	—	—
144	159	16	cips cami_fpga_ver (152)	—	—
160	175	16	cips hk_fpga_ver (242)	—	—
176	176	1	cips reserved_bit (431)	—	—
177	177	1	cips reserved_bit (431)	—	—
178	178	1	cips reserved_bit (431)	—	—
179	179	1	cips reserved_bit (431)	—	—
180	180	1	cips reserved_bit (431)	—	—
181	181	1	cips reserved_bit (431)	—	—
182	182	1	cips reserved_bit (431)	—	—
183	185	3	cips eeprom_reset (220)	—	—
186	186	1	cips comm_sram_a16 (176)	—	—
187	187	1	cips uart_ab_sel (468)	—	—
188	191	4	cips uart_fifo_level (468)	—	—
192	192	1	cips reserved_bit (431)	—	—
193	193	1	cips reserved_bit (431)	—	—

Packet **cips hw_hk** Format, cont'd

start bit	stop bit	size bits	field		
194	194	1	cips cde_uart_f_err (160)	—	—
195	195	1	cips cde_uart_rx_or (161)	—	—
196	196	1	cips cde_uart_rx_af (160)	—	—
197	197	1	cips cde_uart_rx_f (161)	—	—
198	198	1	cips cde_uart_rx_r (162)	—	—
199	199	1	cips cde_uart_rx_e (160)	—	—
200	200	1	cips reserved_bit (431)	—	—
201	201	1	cips reserved_bit (431)	—	—
202	202	1	cips reserved_bit (431)	—	—
203	203	1	cips cde_uart_tx_b (164)	—	—
204	204	1	cips cde_uart_tx_af (164)	—	—
205	205	1	cips cde_uart_tx_f (165)	—	—
206	206	1	cips cde_uart_tx_ae (163)	—	—
207	207	1	cips cde_uart_tx_e (164)	—	—
208	208	1	cips reserved_bit (431)	—	—
209	209	1	cips reserved_bit (431)	—	—
210	210	1	cips reserved_bit (431)	—	—
211	211	1	cips reserved_bit (431)	—	—
212	212	1	cips check_ccd_cmd_p (170)	—	—
213	213	1	cips sdram_chip_remap (436)	—	—
214	215	2	cips sdram_pixel_map (438)	—	—
216	216	1	cips reserved_bit (431)	—	—
217	217	1	cips reserved_bit (431)	—	—
218	218	1	cips reserved_bit (431)	—	—
219	219	1	cips reserved_bit (431)	—	—
220	220	1	cips reserved_bit (431)	—	—
221	221	1	cips reserved_bit (431)	—	—
222	223	2	cips test_cam_select (466)	—	—
224	224	1	cips srt_cr_stex (452)	—	—
225	225	1	cips reserved_bit (431)	—	—
226	226	1	cips reserved_bit (431)	—	—
227	227	1	cips srt_cr_chaen (449)	—	—
228	228	1	cips srt_cr_chben (450)	—	—
229	229	1	cips srt_cr_etce (451)	—	—
230	230	1	cips reserved_bit (431)	—	—
231	231	1	cips reserved_bit (431)	—	—
232	232	1	cips reserved_bit (431)	—	—
233	233	1	cips srt_cr_bufc (449)	—	—
234	234	1	cips reserved_bit (431)	—	—
235	235	1	cips srt_cr_bcen (449)	—	—
236	236	1	cips srt_cr_dynbc (450)	—	—
237	237	1	cips srt_cr_ppen (452)	—	—
238	238	1	cips srt_cr_inten (451)	—	—
239	239	1	cips reserved_bit (431)	—	—
240	255	16	cips srt_ill_ptr (454)	—	—
256	271	16	cips srt_desc_ptr (452)	—	—
272	303	32	cips srt_rx_sa_ill (457)	—	—

Packet **cips hw_hk** Format, cont'd

start bit	stop bit	size bits	field		
304	335	32	cips srt_tx_sa_ill (460)	—	—
336	367	32	cips srt_brx_sa_ill (447)	—	—
368	399	32	cips srt_btx_sa_ill (448)	—	—
400	431	32	cips srt_rx_mc_ill (457)	—	—
432	463	32	cips srt_tx_mc_ill (460)	—	—
464	495	32	cips srt_brx_mc_ill (447)	—	—
496	527	32	cips srt_btx_mc_ill (448)	—	—
528	543	16	cips proc_px_cam_rs (387)	—	—
544	559	16	cips proc_py_cam_rs (388)	—	—
560	575	16	cips proc_mx_cam_rs (386)	—	—
576	591	16	cips proc_my_cam_rs (386)	—	—
592	592	1	cips reserved_bit (431)	—	—
593	593	1	cips reserved_bit (431)	—	—
594	594	1	cips reserved_bit (431)	—	—
595	595	1	cips reserved_bit (431)	—	—
596	596	1	cips reserved_bit (431)	—	—
597	597	1	cips reserved_bit (431)	—	—
598	598	1	cips reserved_bit (431)	—	—
599	599	1	cips reserved_bit (431)	—	—
600	600	1	cips my_lens_htr_ena (324)	—	—
601	601	1	cips mx_lens_htr_ena (305)	—	—
602	602	1	cips py_lens_htr_ena (423)	—	—
603	603	1	cips px_lens_htr_ena (403)	—	—
604	604	1	cips my_cam_pwr (316)	—	—
605	605	1	cips mx_cam_pwr (296)	—	—
606	606	1	cips py_cam_pwr (414)	—	—
607	607	1	cips px_cam_pwr (394)	—	—
608	623	16	cips lens_htr_uthresh (268)	—	—
624	639	16	cips lens_htr_lthresh (267)	—	—
640	640	1	cips reserved_bit (431)	—	—
641	641	1	cips reserved_bit (431)	—	—
642	642	1	cips reserved_bit (431)	—	—
643	643	1	cips reserved_bit (431)	—	—
644	644	1	cips reserved_bit (431)	—	—
645	645	1	cips reserved_bit (431)	—	—
646	646	1	cips reserved_bit (431)	—	—
647	647	1	cips reserved_bit (431)	—	—
648	648	1	cips reserved_bit (431)	—	—
649	649	1	cips reserved_bit (431)	—	—
650	650	1	cips reserved_bit (431)	—	—
651	651	1	cips reserved_bit (431)	—	—
652	655	4	cips dc_activated (202)	—	—
656	671	16	cips dc_timeout (205)	—	—
672	672	1	cips reserved_bit (431)	—	—
673	673	1	cips reserved_bit (431)	—	—
674	674	1	cips reserved_bit (431)	—	—
675	675	1	cips reserved_bit (431)	—	—

Packet **cips hw_hk** Format, cont'd

start bit	stop bit	size bits	field		
676	676	1	cips reserved_bit (431)	—	—
677	677	1	cips reserved_bit (431)	—	—
678	678	1	cips reserved_bit (431)	—	—
679	679	1	cips reserved_bit (431)	—	—
680	680	1	cips reserved_bit (431)	—	—
681	681	1	cips reserved_bit (431)	—	—
682	682	1	cips reserved_bit (431)	—	—
683	683	1	cips sdram_update_ena (439)	—	—
684	684	1	cips reserved_bit (431)	—	—
685	685	1	cips reserved_bit (431)	—	—
686	686	1	cips sdram_b_pwr (436)	—	—
687	687	1	cips sdram_a_pwr (435)	—	—
688	703	16	cips sdram_i_thresh (438)	—	—
704	704	1	cips reserved_bit (431)	—	—
705	705	1	cips reserved_bit (431)	—	—
706	706	1	cips reserved_bit (431)	—	—
707	707	1	cips reserved_bit (431)	—	—
708	708	1	cips reserved_bit (431)	—	—
709	709	1	cips reserved_bit (431)	—	—
710	710	1	cips reserved_bit (431)	—	—
711	711	1	cips reserved_bit (431)	—	—
712	712	1	cips my_dc_status_sec (319)	—	—
713	713	1	cips mx_dc_status_sec (300)	—	—
714	714	1	cips py_dc_status_sec (418)	—	—
715	715	1	cips px_dc_status_sec (398)	—	—
716	716	1	cips my_dc_status_pri (319)	—	—
717	717	1	cips mx_dc_status_pri (299)	—	—
718	718	1	cips py_dc_status_pri (417)	—	—
719	719	1	cips px_dc_status_pri (397)	—	—
720	720	1	cips reserved_bit (431)	—	—
721	721	1	cips reserved_bit (431)	—	—
722	722	1	cips reserved_bit (431)	—	—
723	723	1	cips reserved_bit (431)	—	—
724	724	1	cips reserved_bit (431)	—	—
725	725	1	cips reserved_bit (431)	—	—
726	726	1	cips reserved_bit (431)	—	—
727	727	1	cips reserved_bit (431)	—	—
728	728	1	cips reserved_bit (431)	—	—
729	729	1	cips reserved_bit (431)	—	—
730	730	1	cips reserved_bit (431)	—	—
731	731	1	cips reserved_bit (431)	—	—
732	732	1	cips my_ips_pwr (322)	—	—
733	733	1	cips mx_ips_pwr (303)	—	—
734	734	1	cips py_ips_pwr (421)	—	—
735	735	1	cips px_ips_pwr (401)	—	—
736	736	1	cips reserved_bit (431)	—	—
737	737	1	cips reserved_bit (431)	—	—

Packet **cips hw_hk** Format, cont'd

start bit	stop bit	size bits	field		
738	738	1	cips reserved_bit (431)	—	—
739	739	1	cips reserved_bit (431)	—	—
740	740	1	cips reserved_bit (431)	—	—
741	741	1	cips reserved_bit (431)	—	—
742	742	1	cips reserved_bit (431)	—	—
743	743	1	cips reserved_bit (431)	—	—
744	744	1	cips reserved_bit (431)	—	—
745	745	1	cips reserved_bit (431)	—	—
746	746	1	cips reserved_bit (431)	—	—
747	747	1	cips reserved_bit (431)	—	—
748	748	1	cips my_bos_ena (311)	—	—
749	749	1	cips mx_bos_ena (291)	—	—
750	750	1	cips py_bos_ena (409)	—	—
751	751	1	cips px_bos_ena (389)	—	—
752	752	1	cips reserved_bit (431)	—	—
753	753	1	cips reserved_bit (431)	—	—
754	754	1	cips reserved_bit (431)	—	—
755	755	1	cips reserved_bit (431)	—	—
756	756	1	cips reserved_bit (431)	—	—
757	757	1	cips reserved_bit (431)	—	—
758	758	1	cips reserved_bit (431)	—	—
759	759	1	cips reserved_bit (431)	—	—
760	760	1	cips my_cam_mcp_pwr (313)	—	—
761	761	1	cips mx_cam_mcp_pwr (293)	—	—
762	762	1	cips py_cam_mcp_pwr (411)	—	—
763	763	1	cips px_cam_mcp_pwr (392)	—	—
764	764	1	cips my_cam_p_pwr (315)	—	—
765	765	1	cips mx_cam_p_pwr (295)	—	—
766	766	1	cips py_cam_p_pwr (413)	—	—
767	767	1	cips px_cam_p_pwr (393)	—	—
768	768	1	cips reserved_bit (431)	—	—
769	769	1	cips reserved_bit (431)	—	—
770	770	1	cips reserved_bit (431)	—	—
771	771	1	cips reserved_bit (431)	—	—
772	772	1	cips my_shutr_disable (328)	—	—
773	773	1	cips mx_shutr_disable (310)	—	—
774	774	1	cips py_shutr_disable (427)	—	—
775	775	1	cips px_shutr_disable (408)	—	—
776	776	1	cips reserved_bit (431)	—	—
777	777	1	cips reserved_bit (431)	—	—
778	778	1	cips reserved_bit (431)	—	—
779	779	1	cips reserved_bit (431)	—	—
780	780	1	cips my_shutr_state (329)	—	—
781	781	1	cips mx_shutr_state (310)	—	—
782	782	1	cips py_shutr_state (428)	—	—
783	783	1	cips px_shutr_state (408)	—	—
784	784	1	cips reserved_bit (431)	—	—

Packet **cips hw_hk** Format, cont'd

start bit	stop bit	size bits	field		
785	791	7	cips hk_mux_ena (250)	—	—
792	792	1	cips reserved_bit (431)	—	—
793	795	3	cips hk_mux_2_addr (250)	—	—
796	796	1	cips reserved_bit (431)	—	—
797	799	3	cips hk_mux_1_addr (250)	—	—
800	800	1	cips reserved_bit (431)	—	—
801	801	1	cips reserved_bit (431)	—	—
802	802	1	cips reserved_bit (431)	—	—
803	803	1	cips reserved_bit (431)	—	—
804	804	1	cips reserved_bit (431)	—	—
805	805	1	cips reserved_bit (431)	—	—
806	806	1	cips reserved_bit (431)	—	—
807	807	1	cips reserved_bit (431)	—	—
808	808	1	cips reserved_bit (431)	—	—
809	809	1	cips reserved_bit (431)	—	—
810	810	1	cips reserved_bit (431)	—	—
811	811	1	cips hk_mux_test_ena (251)	—	—
812	812	1	cips reserved_bit (431)	—	—
813	813	1	cips reserved_bit (431)	—	—
814	814	1	cips reserved_bit (431)	—	—
815	815	1	cips hk_lpbit_ena (249)	—	—
816	816	1	cips reserved_bit (431)	—	—
817	817	1	cips reserved_bit (431)	—	—
818	818	1	cips reserved_bit (431)	—	—
819	819	1	cips reserved_bit (431)	—	—
820	820	1	cips reserved_bit (431)	—	—
821	821	1	cips reserved_bit (431)	—	—
822	822	1	cips reserved_bit (431)	—	—
823	823	1	cips reserved_bit (431)	—	—
824	831	8	cips hk_counter (242)	—	—
832	847	16	cips px_bos_thresh (390)	—	—
848	863	16	cips py_bos_thresh (410)	—	—
864	879	16	cips mx_bos_thresh (292)	—	—
880	895	16	cips my_bos_thresh (312)	—	—
896	911	16	cips px_bos_level (389)	—	—
912	927	16	cips py_bos_level (409)	—	—
928	943	16	cips mx_bos_level (291)	—	—
944	959	16	cips my_bos_level (311)	—	—
960	975	16	cips hk_pcb_temp (251)	—	—
976	991	16	cips dc_conv_1_temp (203)	—	—
992	1007	16	cips dc_conv_2_temp (204)	—	—
1008	1023	16	cips proc_pcb_temp (387)	—	—
1024	1039	16	cips cami_pcb_temp (152)	—	—
1040	1055	16	cips px_py_ips_temp (405)	—	—
1056	1071	16	cips mx_my_ips_temp (306)	—	—
1072	1087	16	cips px_ccd_temp (396)	—	—
1088	1103	16	cips py_ccd_temp (416)	—	—

Packet **cips hw_hk** Format, cont'd

start bit	stop bit	size bits	field		
1104	1119	16	cips mx_ccd_temp (298)	—	—
1120	1135	16	cips my_ccd_temp (318)	—	—
1136	1151	16	cips px_lens_a_temp (401)	—	—
1152	1167	16	cips py_lens_a_temp (421)	—	—
1168	1183	16	cips mx_lens_a_temp (303)	—	—
1184	1199	16	cips my_lens_a_temp (323)	—	—
1200	1215	16	cips px_lens_b_temp (402)	—	—
1216	1231	16	cips py_lens_b_temp (422)	—	—
1232	1247	16	cips mx_lens_b_temp (304)	—	—
1248	1263	16	cips my_lens_b_temp (324)	—	—
1264	1279	16	cips px_int_temp (400)	—	—
1280	1295	16	cips py_int_temp (420)	—	—
1296	1311	16	cips mx_int_temp (302)	—	—
1312	1327	16	cips my_int_temp (322)	—	—
1328	1343	16	cips gndv_mon (241)	—	—
1344	1359	16	cips p20v_mon (340)	—	—
1360	1375	16	cips p15v_mon (339)	—	—
1376	1391	16	cips m15v_mon (270)	—	—
1392	1407	16	cips p5v_mon (374)	—	—
1408	1423	16	cips p2_5v_mon (340)	—	—
1424	1439	16	cips cpa_p3_3v_mon (177)	—	—
1440	1455	16	cips px_cam_pv_mon (394)	—	—
1456	1471	16	cips py_cam_pv_mon (414)	—	—
1472	1487	16	cips mx_cam_pv_mon (296)	—	—
1488	1503	16	cips my_cam_pv_mon (316)	—	—
1504	1519	16	cips px_cam_mcpv_mon (392)	—	—
1520	1535	16	cips py_cam_mcpv_mon (412)	—	—
1536	1551	16	cips mx_cam_mcpv_mon (294)	—	—
1552	1567	16	cips my_cam_mcpv_mon (314)	—	—
1568	1583	16	cips sdram_i_mon (437)	—	—
1584	1599	16	cips p5i_mon (373)	—	—
1600	1615	16	cips p15i_mon (338)	—	—
1616	1631	16	cips m15i_mon (269)	—	—
1632	1647	16	cips pkt_crc (378)	—	—

Packet size = 1648 bits (206 bytes) .

F.6 cips mem_dump

Packet Source Application Identifier

0x164 (356)

OASIS Stream Name

cips356

Description

cips memory dump packet.

Discussion

CIPS memory dump packet.

Default Packet Production Status: Packets produced asynchronously on command.

Related Commands

<i>command</i>	<i>description</i>
dump cips memory (57)	Dump CIPS memory.

Format

Packet **cips mem_dump** Format

start bit	stop bit	size bits	field	control	method
0	15	16	cips p356pid (354)	—	—
16	17	2	cips p356flg (352)	—	—
18	31	14	cips p356sct (354)	—	—
32	47	16	cips p356pl (354)	—	—
48	63	16	cips p356hws (352)	—	—
64	79	16	cips p356lws (353)	—	—
80	95	16	cips p356hwss (353)	—	—
96	111	16	cips p356lwss (353)	—	—
112	127	16	cips pkt.version (379)	—	—
128	159	32	cips dump.length (206)	—	—
160	175	16	cips dump.type (206)	—	—
176	191	16	cips dump.addr (205)	—	—
192	207	16	cips dump.data (205)	cips dump.length (206)	COUNT
208	223	16	cips pkt.crc (378)	—	—

Packet size = 224 bits (28 bytes) plus N*P, where N is the number of subpackets in the packet and P is the size of the subpacket (which, of course, may need to be calculated based on sub-subpacket size). .

F.7 cips post_mortem

Packet Source Application Identifier

0x165 (357)

OASIS Stream Name

cips357

Description

cips post-mortem packet.

Discussion

CIPS post-mortem packet.

Default Packet Production Status: Packets produced asynchronously on a processor reset.

Related Commands

<i>command</i>	<i>description</i>
issue cips pm (82)	Send a post-mortem telemetry packet.

Format

Packet **cips post_mortem** Format

start bit	stop bit	size bits	field	control	method
0	15	16	cips p357pid (356)	—	—
16	17	2	cips p357flg (355)	—	—
18	31	14	cips p357sct (357)	—	—
32	47	16	cips p357pl (357)	—	—
48	63	16	cips p357hws (355)	—	—
64	79	16	cips p357lws (356)	—	—
80	95	16	cips p357hwss (355)	—	—
96	111	16	cips p357lwss (356)	—	—
112	119	8	cips csci (199)	—	—
120	127	8	cips pkt_version_el (379)	—	—
128	143	16	cips pm_event_ct (380)	—	—
144	143	0	cips pm_event_info (380)	cips pm_event_ct (380)	COUNT
144	159	16	cips pkt_crc (378)	—	—

Packet size = 160 bits (20 bytes) plus N*P, where N is the number of subpackets in the packet and P is the size of the subpacket (which, of course, may need to be calculated based on sub-subpacket size). .

Pm Event Info Format

Pm Event Info Format

start bit	stop bit	size bits	field	control	method
0	31	32	cips pm_time_s (383)	—	—
32	63	32	cips pm_time_ss (383)	—	—
64	95	32	cips pm_key (381)	—	—
96	111	16	cips pm_pc (382)	—	—

Pm Event Info Format, cont'd

start bit	stop bit	size bits	field		
112	127	16	cips pm.p0 (381)	—	—
128	143	16	cips pm.p1 (381)	—	—
144	159	16	cips pm.p2 (382)	—	—
160	175	16	cips pm.p3 (382)	—	—

Maximum Subpacket size = 176 bits (22 bytes)

F.8 cips sci_mx_cam

Packet Source Application Identifier

0x16a (362)

OASIS Stream Name

cips362

Description

CIPS -X camera science data packet.

Discussion

Contains CIPS science detector data and corresponding instrument configuration.

Packets will contain data for at most one CCD image.

A single CCD image may span multiple CCSDS telemetry packets, depending on how large the image is (i.e. will it fit within the AIM CCSDS telemetry packet size limit). If this is the case, the 'checksum_error' telemetry point should have a value of 'UNKNOWN' for all packets except the last packet in the image. The value of this telemetry point in the last packet will be either 'TRUE' or 'FALSE', depending on the actual status of the checksum comparison being done by CIPS FSW.

Most of the data items in this packet should be constant for all packets associated with the same image. The exceptions are the CCSDS header information and the 'start_pixel', 'chksum_error', 'pixels_in_pkt', and 'pkt_crc' telemetry points. Of course, the 'pixel_stream' telemetry item, which contains the actual science image data, will also change with each packet.

Images taken simultaneously on multiple cameras can be correlated through the 'img_timestamp' telemetry points. These telemetry points can also be used to piece together the multiple packets that may constitute up a single image.

While the 'img_timestamp' telemetry items give the approximate CCD exposure time, the exact CCD exposure time can be more correctly calculated using the 'fast_clears', 'frame_sync_time', and 'shutter_cl.time' telemetry points according to the following equations:

$$\text{Shutter Open Time} = \text{img_timestamp} + \text{frame_sync_time} + (583 * \text{fast_clears})$$

$$\text{Shutter Close Time} = \text{img_timestamp} + \text{shutter_cl.time}$$

Note that all of the telemetry points used in these calculations, with the exception of the timestamp, are in milliseconds.

Default Packet Production Status: Enabled and produced synchronously after enough CIPS cycles have completed such that the packet is full.

Related Commands

<i>command</i>	<i>description</i>
enable cips sdp (66)	Enable CIPS science data processing and downlink.
enable cips sw_bin (69)	Enable science data binning capability in CIPS FSW.
disable cips comp (49)	Disable science data compression capability in CIPS FSW.
set cips img_size (104)	Set the raw CCD image size for all CIPS cameras.
disable cips sdp (53)	Disable CIPS science data processing and downlink.
disable cips cam_bin (47)	Disable the cameras' science data binning capability.
enable cips cam_bin (60)	Enable the cameras' science data binning capability.
enable cips comp (61)	Enable science data compression capability in CIPS FSW.
disable cips sw_bin (55)	Disable science data binning capability in CIPS FSW.

FormatPacket **cips sci_mx_cam** Format

start bit	stop bit	size bits	field	control	method
0	15	16	cips p362pid (370)	—	—
16	17	2	cips p362flg (368)	—	—
18	31	14	cips p362sct (370)	—	—
32	47	16	cips p362pl (370)	—	—
48	63	16	cips p362hws (368)	—	—
64	79	16	cips p362lws (369)	—	—
80	95	16	cips p362hwss (369)	—	—
96	111	16	cips p362lwss (369)	—	—
112	127	16	cips pkt_version (379)	—	—
128	159	32	cips img_timestamp_s (263)	—	—
160	191	32	cips img_timestamp_ss (264)	—	—
192	207	16	cips obs_id (337)	—	—
208	239	32	cips start_pixel (462)	—	—
240	247	8	cips bits_per_pixel (148)	—	—
248	255	8	cips cur_img_ct (201)	—	—
256	256	1	cips adc_lup (147)	—	—
257	258	2	cips chksum_error (170)	—	—
259	261	3	cips comp_scheme (177)	—	—
262	262	1	cips img_mode (260)	—	—
263	263	1	cips tdi_dir (465)	—	—
264	264	1	cips sw_bin_ena (462)	—	—
265	265	1	cips cam_bin_ena (151)	—	—
266	268	3	cips ccd_col_bin (153)	—	—
269	271	3	cips ccd_row_bin (154)	—	—
272	279	8	cips cam_fpga_ver (152)	—	—
280	287	8	cips tdi_rows (466)	—	—
288	290	3	cips fast_clears (240)	—	—
291	291	1	cips pixel_count_err (376)	—	—
292	292	1	cips shutter_disable (440)	—	—
293	293	1	cips reserved_bit (431)	—	—
294	303	10	cips tdi_dwell (465)	—	—
304	319	16	cips shutter_cl_time (440)	—	—
320	335	16	cips frame_sync_time (240)	—	—
336	336	1	cips reserved_bit (431)	—	—
337	337	1	cips reserved_bit (431)	—	—
338	338	1	cips reserved_bit (431)	—	—
339	339	1	cips reserved_bit (431)	—	—
340	351	12	cips img_offset_col (260)	—	—
352	352	1	cips reserved_bit (431)	—	—
353	353	1	cips reserved_bit (431)	—	—
354	354	1	cips reserved_bit (431)	—	—
355	355	1	cips reserved_bit (431)	—	—
356	367	12	cips img_offset_row (261)	—	—
368	371	4	cips opcode_err_ct (338)	—	—
372	375	4	cips row_seq_err_ct (431)	—	—
376	383	8	cips parity_err_ct (375)	—	—

Packet **cips sci_mx_cam** Format, cont'd

start bit	stop bit	size bits	field		
384	399	16	cips img_cols (258)	–	–
400	415	16	cips img_rows (261)	–	–
416	431	16	cips pixels_in_pkt (377)	–	–
432	8175	7744	cips pixel_stream (377)	cips pixels_in_pkt (377)	COUNT
8176	8191	16	cips pkt_crc (378)	–	–

Packet size = 8192 bits (1024 bytes) plus N*P, where N is the number of subpackets in the packet and P is the size of the subpacket (which, of course, may need to be calculated based on sub-subpacket size). .

F.9 cips sci_my_cam

Packet Source Application Identifier

0x16b (363)

OASIS Stream Name

cips363

Description

CIPS -Y camera science data packet.

Discussion

Contains CIPS science detector data and corresponding instrument configuration.

Packets will contain data for at most one CCD image.

A single CCD image may span multiple CCSDS telemetry packets, depending on how large the image is (i.e. will it fit within the AIM CCSDS telemetry packet size limit). If this is the case, the 'checksum_error' telemetry point should have a value of 'UNKNOWN' for all packets except the last packet in the image. The value of this telemetry point in the last packet will be either 'TRUE' or 'FALSE', depending on the actual status of the checksum comparison being done by CIPS FSW.

Most of the data items in this packet should be constant for all packets associated with the same image. The exceptions are the CCSDS header information and the 'start_pixel', 'chksum_error', 'pixels_in_pkt', and 'pkt_crc' telemetry points. Of course, the 'pixel_stream' telemetry item, which contains the actual science image data, will also change with each packet.

Images taken simultaneously on multiple cameras can be correlated through the 'img_timestamp' telemetry points. These telemetry points can also be used to piece together the multiple packets that may constitute up a single image.

While the 'img_timestamp' telemetry items give the approximate CCD exposure time, the exact CCD exposure time can be more correctly calculated using the 'fast_clears', 'frame_sync_time', and 'shutter_cl.time' telemetry points according to the following equations:

$$\text{Shutter Open Time} = \text{img_timestamp} + \text{frame_sync_time} + (583 * \text{fast_clears})$$

$$\text{Shutter Close Time} = \text{img_timestamp} + \text{shutter_cl.time}$$

Note that all of the telemetry points used in these calculations, with the exception of the timestamp, are in milliseconds.

Default Packet Production Status: Enabled and produced synchronously after enough CIPS cycles have completed such that the packet is full.

Related Commands

<i>command</i>	<i>description</i>
enable cips sdp (66)	Enable CIPS science data processing and downlink.
enable cips sw_bin (69)	Enable science data binning capability in CIPS FSW.
disable cips comp (49)	Disable science data compression capability in CIPS FSW.
set cips img_size (104)	Set the raw CCD image size for all CIPS cameras.
disable cips sdp (53)	Disable CIPS science data processing and downlink.
disable cips cam_bin (47)	Disable the cameras' science data binning capability.
enable cips cam_bin (60)	Enable the cameras' science data binning capability.
enable cips comp (61)	Enable science data compression capability in CIPS FSW.
disable cips sw_bin (55)	Disable science data binning capability in CIPS FSW.

FormatPacket **cips sci_my_cam** Format

start bit	stop bit	size bits	field	control	method
0	15	16	cips p363pid (372)	—	—
16	17	2	cips p363flg (371)	—	—
18	31	14	cips p363sct (373)	—	—
32	47	16	cips p363pl (373)	—	—
48	63	16	cips p363hws (371)	—	—
64	79	16	cips p363lws (372)	—	—
80	95	16	cips p363hwss (371)	—	—
96	111	16	cips p363lwss (372)	—	—
112	127	16	cips pkt_version (379)	—	—
128	159	32	cips img_timestamp_s (263)	—	—
160	191	32	cips img_timestamp_ss (264)	—	—
192	207	16	cips obs_id (337)	—	—
208	239	32	cips start_pixel (462)	—	—
240	247	8	cips bits_per_pixel (148)	—	—
248	255	8	cips cur_img_ct (201)	—	—
256	256	1	cips adc_lup (147)	—	—
257	258	2	cips chksum_error (170)	—	—
259	261	3	cips comp_scheme (177)	—	—
262	262	1	cips img_mode (260)	—	—
263	263	1	cips tdi_dir (465)	—	—
264	264	1	cips sw_bin_ena (462)	—	—
265	265	1	cips cam_bin_ena (151)	—	—
266	268	3	cips ccd_col_bin (153)	—	—
269	271	3	cips ccd_row_bin (154)	—	—
272	279	8	cips cam_fpga_ver (152)	—	—
280	287	8	cips tdi_rows (466)	—	—
288	290	3	cips fast_clears (240)	—	—
291	291	1	cips pixel_count_err (376)	—	—
292	292	1	cips shutter_disable (440)	—	—
293	293	1	cips reserved_bit (431)	—	—
294	303	10	cips tdi_dwell (465)	—	—
304	319	16	cips shutter_cl_time (440)	—	—
320	335	16	cips frame_sync_time (240)	—	—
336	336	1	cips reserved_bit (431)	—	—
337	337	1	cips reserved_bit (431)	—	—
338	338	1	cips reserved_bit (431)	—	—
339	339	1	cips reserved_bit (431)	—	—
340	351	12	cips img_offset_col (260)	—	—
352	352	1	cips reserved_bit (431)	—	—
353	353	1	cips reserved_bit (431)	—	—
354	354	1	cips reserved_bit (431)	—	—
355	355	1	cips reserved_bit (431)	—	—
356	367	12	cips img_offset_row (261)	—	—
368	371	4	cips opcode_err_ct (338)	—	—
372	375	4	cips row_seq_err_ct (431)	—	—
376	383	8	cips parity_err_ct (375)	—	—

Packet **cips sci_my_cam** Format, cont'd

start bit	stop bit	size bits	field		
384	399	16	cips img_cols (258)	–	–
400	415	16	cips img_rows (261)	–	–
416	431	16	cips pixels_in_pkt (377)	–	–
432	8175	7744	cips pixel_stream (377)	cips pixels_in_pkt (377)	COUNT
8176	8191	16	cips pkt_crc (378)	–	–

Packet size = 8192 bits (1024 bytes) plus N*P, where N is the number of subpackets in the packet and P is the size of the subpacket (which, of course, may need to be calculated based on sub-subpacket size). .

F.10 cips sci_px_cam

Packet Source Application Identifier

0x168 (360)

OASIS Stream Name

cips360

Description

CIPS +X camera science data packet.

Discussion

Contains CIPS science detector data and corresponding instrument configuration.

Packets will contain data for at most one CCD image.

A single CCD image may span multiple CCSDS telemetry packets, depending on how large the image is (i.e. will it fit within the AIM CCSDS telemetry packet size limit). If this is the case, the 'checksum_error' telemetry point should have a value of 'UNKNOWN' for all packets except the last packet in the image. The value of this telemetry point in the last packet will be either 'TRUE' or 'FALSE', depending on the actual status of the checksum comparison being done by CIPS FSW.

Most of the data items in this packet should be constant for all packets associated with the same image. The exceptions are the CCSDS header information and the 'start_pixel', 'chksum_error', 'pixels_in_pkt', and 'pkt_crc' telemetry points. Of course, the 'pixel_stream' telemetry item, which contains the actual science image data, will also change with each packet.

Images taken simultaneously on multiple cameras can be correlated through the 'img_timestamp' telemetry points. These telemetry points can also be used to piece together the multiple packets that may constitute up a single image.

While the 'img_timestamp' telemetry items give the approximate CCD exposure time, the exact CCD exposure time can be more correctly calculated using the 'fast_clears', 'frame_sync_time', and 'shutter_cl.time' telemetry points according to the following equations:

$$\text{Shutter Open Time} = \text{img_timestamp} + \text{frame_sync_time} + (583 * \text{fast_clears})$$

$$\text{Shutter Close Time} = \text{img_timestamp} + \text{shutter_cl.time}$$

Note that all of the telemetry points used in these calculations, with the exception of the timestamp, are in milliseconds.

Default Packet Production Status: Enabled and produced synchronously after enough CIPS cycles have completed such that the packet is full.

Related Commands

<i>command</i>	<i>description</i>
enable cips sdp (66)	Enable CIPS science data processing and downlink.
enable cips sw_bin (69)	Enable science data binning capability in CIPS FSW.
disable cips comp (49)	Disable science data compression capability in CIPS FSW.
set cips img_size (104)	Set the raw CCD image size for all CIPS cameras.
disable cips sdp (53)	Disable CIPS science data processing and downlink.
disable cips cam_bin (47)	Disable the cameras' science data binning capability.
enable cips cam_bin (60)	Enable the cameras' science data binning capability.
enable cips comp (61)	Enable science data compression capability in CIPS FSW.
disable cips sw_bin (55)	Disable science data binning capability in CIPS FSW.

FormatPacket **cips sci_px_cam** Format

start bit	stop bit	size bits	field	control	method
0	15	16	cips p360pid (364)	—	—
16	17	2	cips p360flg (363)	—	—
18	31	14	cips p360sct (365)	—	—
32	47	16	cips p360pl (365)	—	—
48	63	16	cips p360hws (363)	—	—
64	79	16	cips p360lws (364)	—	—
80	95	16	cips p360hwss (363)	—	—
96	111	16	cips p360lwss (364)	—	—
112	127	16	cips pkt_version (379)	—	—
128	159	32	cips img_timestamp_s (263)	—	—
160	191	32	cips img_timestamp_ss (264)	—	—
192	207	16	cips obs_id (337)	—	—
208	239	32	cips start_pixel (462)	—	—
240	247	8	cips bits_per_pixel (148)	—	—
248	255	8	cips cur_img_ct (201)	—	—
256	256	1	cips adc_lup (147)	—	—
257	258	2	cips chksum_error (170)	—	—
259	261	3	cips comp_scheme (177)	—	—
262	262	1	cips img_mode (260)	—	—
263	263	1	cips tdi_dir (465)	—	—
264	264	1	cips sw_bin_ena (462)	—	—
265	265	1	cips cam_bin_ena (151)	—	—
266	268	3	cips ccd_col_bin (153)	—	—
269	271	3	cips ccd_row_bin (154)	—	—
272	279	8	cips cam_fpga_ver (152)	—	—
280	287	8	cips tdi_rows (466)	—	—
288	290	3	cips fast_clears (240)	—	—
291	291	1	cips pixel_count_err (376)	—	—
292	292	1	cips shutter_disable (440)	—	—
293	293	1	cips reserved_bit (431)	—	—
294	303	10	cips tdi_dwell (465)	—	—
304	319	16	cips shutter_cl_time (440)	—	—
320	335	16	cips frame_sync_time (240)	—	—
336	336	1	cips reserved_bit (431)	—	—
337	337	1	cips reserved_bit (431)	—	—
338	338	1	cips reserved_bit (431)	—	—
339	339	1	cips reserved_bit (431)	—	—
340	351	12	cips img_offset_col (260)	—	—
352	352	1	cips reserved_bit (431)	—	—
353	353	1	cips reserved_bit (431)	—	—
354	354	1	cips reserved_bit (431)	—	—
355	355	1	cips reserved_bit (431)	—	—
356	367	12	cips img_offset_row (261)	—	—
368	371	4	cips opcode_err_ct (338)	—	—
372	375	4	cips row_seq_err_ct (431)	—	—
376	383	8	cips parity_err_ct (375)	—	—

Packet **cips sci_px_cam** Format, cont'd

start bit	stop bit	size bits	field		
384	399	16	cips img_cols (258)	–	–
400	415	16	cips img_rows (261)	–	–
416	431	16	cips pixels_in_pkt (377)	–	–
432	8175	7744	cips pixel_stream (377)	cips pixels_in_pkt (377)	COUNT
8176	8191	16	cips pkt_crc (378)	–	–

Packet size = 8192 bits (1024 bytes) plus N*P, where N is the number of subpackets in the packet and P is the size of the subpacket (which, of course, may need to be calculated based on sub-subpacket size). .

F.11 cips sci_py_cam

Packet Source Application Identifier

0x169 (361)

OASIS Stream Name

cips361

Description

CIPS +Y camera science data packet.

Discussion

Contains CIPS science detector data and corresponding instrument configuration.

Packets will contain data for at most one CCD image.

A single CCD image may span multiple CCSDS telemetry packets, depending on how large the image is (i.e. will it fit within the AIM CCSDS telemetry packet size limit). If this is the case, the 'checksum_error' telemetry point should have a value of 'UNKNOWN' for all packets except the last packet in the image. The value of this telemetry point in the last packet will be either 'TRUE' or 'FALSE', depending on the actual status of the checksum comparison being done by CIPS FSW.

Most of the data items in this packet should be constant for all packets associated with the same image. The exceptions are the CCSDS header information and the 'start_pixel', 'chksum_error', 'pixels_in_pkt', and 'pkt_crc' telemetry points. Of course, the 'pixel_stream' telemetry item, which contains the actual science image data, will also change with each packet.

Images taken simultaneously on multiple cameras can be correlated through the 'img_timestamp' telemetry points. These telemetry points can also be used to piece together the multiple packets that may constitute up a single image.

While the 'img_timestamp' telemetry items give the approximate CCD exposure time, the exact CCD exposure time can be more correctly calculated using the 'fast_clears', 'frame_sync_time', and 'shutter_cl.time' telemetry points according to the following equations:

$$\text{Shutter Open Time} = \text{img_timestamp} + \text{frame_sync_time} + (583 * \text{fast_clears})$$

$$\text{Shutter Close Time} = \text{img_timestamp} + \text{shutter_cl.time}$$

Note that all of the telemetry points used in these calculations, with the exception of the timestamp, are in milliseconds.

Default Packet Production Status: Enabled and produced synchronously after enough CIPS cycles have completed such that the packet is full.

Related Commands

<i>command</i>	<i>description</i>
enable cips sdp (66)	Enable CIPS science data processing and downlink.
enable cips sw_bin (69)	Enable science data binning capability in CIPS FSW.
disable cips comp (49)	Disable science data compression capability in CIPS FSW.
set cips img_size (104)	Set the raw CCD image size for all CIPS cameras.
disable cips sdp (53)	Disable CIPS science data processing and downlink.
disable cips cam_bin (47)	Disable the cameras' science data binning capability.
enable cips cam_bin (60)	Enable the cameras' science data binning capability.
enable cips comp (61)	Enable science data compression capability in CIPS FSW.
disable cips sw_bin (55)	Disable science data binning capability in CIPS FSW.

FormatPacket **cips sci_py_cam** Format

start bit	stop bit	size bits	field	control	method
0	15	16	cips p361pid (367)	—	—
16	17	2	cips p361flg (365)	—	—
18	31	14	cips p361sct (368)	—	—
32	47	16	cips p361pl (367)	—	—
48	63	16	cips p361hws (366)	—	—
64	79	16	cips p361lws (366)	—	—
80	95	16	cips p361hwss (366)	—	—
96	111	16	cips p361lwss (367)	—	—
112	127	16	cips pkt_version (379)	—	—
128	159	32	cips img_timestamp_s (263)	—	—
160	191	32	cips img_timestamp_ss (264)	—	—
192	207	16	cips obs_id (337)	—	—
208	239	32	cips start_pixel (462)	—	—
240	247	8	cips bits_per_pixel (148)	—	—
248	255	8	cips cur_img_ct (201)	—	—
256	256	1	cips adc_lup (147)	—	—
257	258	2	cips chksum_error (170)	—	—
259	261	3	cips comp_scheme (177)	—	—
262	262	1	cips img_mode (260)	—	—
263	263	1	cips tdi_dir (465)	—	—
264	264	1	cips sw_bin_ena (462)	—	—
265	265	1	cips cam_bin_ena (151)	—	—
266	268	3	cips ccd_col_bin (153)	—	—
269	271	3	cips ccd_row_bin (154)	—	—
272	279	8	cips cam_fpga_ver (152)	—	—
280	287	8	cips tdi_rows (466)	—	—
288	290	3	cips fast_clears (240)	—	—
291	291	1	cips pixel_count_err (376)	—	—
292	292	1	cips shutter_disable (440)	—	—
293	293	1	cips reserved_bit (431)	—	—
294	303	10	cips tdi_dwell (465)	—	—
304	319	16	cips shutter_cl_time (440)	—	—
320	335	16	cips frame_sync_time (240)	—	—
336	336	1	cips reserved_bit (431)	—	—
337	337	1	cips reserved_bit (431)	—	—
338	338	1	cips reserved_bit (431)	—	—
339	339	1	cips reserved_bit (431)	—	—
340	351	12	cips img_offset_col (260)	—	—
352	352	1	cips reserved_bit (431)	—	—
353	353	1	cips reserved_bit (431)	—	—
354	354	1	cips reserved_bit (431)	—	—
355	355	1	cips reserved_bit (431)	—	—
356	367	12	cips img_offset_row (261)	—	—
368	371	4	cips opcode_err_ct (338)	—	—
372	375	4	cips row_seq_err_ct (431)	—	—
376	383	8	cips parity_err_ct (375)	—	—

Packet **cips sci_py_cam** Format, cont'd

start bit	stop bit	size bits	field		
384	399	16	cips img_cols (258)	–	–
400	415	16	cips img_rows (261)	–	–
416	431	16	cips pixels_in_pkt (377)	–	–
432	8175	7744	cips pixel_stream (377)	cips pixels_in_pkt (377)	COUNT
8176	8191	16	cips pkt_crc (378)	–	–

Packet size = 8192 bits (1024 bytes) plus N*P, where N is the number of subpackets in the packet and P is the size of the subpacket (which, of course, may need to be calculated based on sub-subpacket size). .

F.12 cips sw_status

Packet Source Application Identifier

0x160 (352)

OASIS Stream Name

cips352

Description

cips software status.

Discussion

CIPS software status, state, counters, etc. This telemetry packet can be produced independent of CIPS instrument configuration.

Default Packet Production Status: Packets produced every 60 seconds.

Related Commands

<i>command</i>	<i>description</i>
set cips sw_stat_rt (107)	Set period of the cips software status telemetry packet.
issue cips sw_stat (84)	Send a cips software status telemetry packet.

Format

Packet **cips sw_status** Format

start bit	stop bit	size bits	field	control	method
0	15	16	cips p352pid (343)	—	—
16	17	2	cips p352flg (341)	—	—
18	31	14	cips p352sct (344)	—	—
32	47	16	cips p352pl (343)	—	—
48	63	16	cips p352hws (341)	—	—
64	79	16	cips p352lws (342)	—	—
80	95	16	cips p352hwss (342)	—	—
96	111	16	cips p352lwss (343)	—	—
112	127	16	cips pkt_version (379)	—	—
128	143	16	cips mode (271)	—	—
144	159	16	cips armed (148)	—	—
160	175	16	cips safe_type (431)	—	—
176	191	16	cips cmd_succ_ct (175)	—	—
192	207	16	cips cmd_fail_ct (173)	—	—
208	215	8	cips cmd_succ (175)	—	—
216	223	8	cips cmd_fail (172)	—	—
224	239	16	cips cmd_fail_code (173)	—	—
240	255	16	cips sw_stat_rt (464)	—	—
256	271	16	cips hw_hk_rt (252)	—	—
272	287	16	cips cde_itf_rt (157)	—	—
288	303	16	cips cmd_loss_time (174)	—	—
304	319	16	cips bos_trip_ct (150)	—	—
320	335	16	cips sdram_trip_ct (439)	—	—
336	399	64	cips acs_sun_uv_x (145)	—	—

Packet **cips sw_status** Format, cont'd

start bit	stop bit	size bits	field		
400	463	64	cips acs_sun_uv_y (146)	—	—
464	527	64	cips acs_sun_uv_z (146)	—	—
528	543	16	cips img_to_be_taken (265)	—	—
544	559	16	cips img_remaining (261)	—	—
560	560	1	cips reserved_bit (431)	—	—
561	561	1	cips reserved_bit (431)	—	—
562	562	1	cips reserved_bit (431)	—	—
563	563	1	cips reserved_bit (431)	—	—
564	564	1	cips reserved_bit (431)	—	—
565	565	1	cips reserved_bit (431)	—	—
566	566	1	cips reserved_bit (431)	—	—
567	567	1	cips reserved_bit (431)	—	—
568	568	1	cips reserved_bit (431)	—	—
569	569	1	cips reserved_bit (431)	—	—
570	570	1	cips reserved_bit (431)	—	—
571	571	1	cips reserved_bit (431)	—	—
572	572	1	cips img_with_my_cam (265)	—	—
573	573	1	cips img_with_mx_cam (265)	—	—
574	574	1	cips img_with_py_cam (266)	—	—
575	575	1	cips img_with_px_cam (266)	—	—
576	591	16	cips img_delay_tm (259)	—	—
592	607	16	cips img_taken (263)	—	—
608	623	16	cips current_img (202)	—	—
624	639	16	cips current_cam (201)	—	—
640	655	16	cips sdp_in_progress (435)	—	—
656	671	16	cips img_cols_ccd (258)	—	—
672	687	16	cips img_rows_ccd (262)	—	—
688	703	16	cips img_cols_sdram (259)	—	—
704	719	16	cips img_rows_sdram (262)	—	—
720	735	16	cips img_cols_tlm (259)	—	—
736	751	16	cips img_rows_tlm (263)	—	—
752	783	32	cips pixels_per_map (377)	—	—
784	799	16	cips pixels_per_pkt (378)	—	—
800	815	16	cips bits_per_pixel_c (149)	—	—
816	831	16	cips columns_per_map (176)	—	—
832	847	16	cips sdp_enable (434)	—	—
848	863	16	cips sw_comp_enable (463)	—	—
864	879	16	cips sw_bin_enable (463)	—	—
880	895	16	cips cam_bin_enable (151)	—	—
896	911	16	cips px_cam_mode (393)	—	—
912	927	16	cips px_cam_f_sync (391)	—	—
928	943	16	cips px_cam_clear_ct (391)	—	—
944	959	16	cips px_cam_tdi_rows (395)	—	—
960	975	16	cips px_cam_tdi_time (396)	—	—
976	991	16	cips px_cam_ss_time (395)	—	—
992	1007	16	cips py_cam_mode (413)	—	—
1008	1023	16	cips py_cam_f_sync (411)	—	—

Packet **cips sw_status** Format, cont'd

start bit	stop bit	size bits	field		
1024	1039	16	cips py_cam_clear_ct (411)	—	—
1040	1055	16	cips py_cam_tdi_rows (415)	—	—
1056	1071	16	cips py_cam_tdi_time (416)	—	—
1072	1087	16	cips py_cam_ss_time (415)	—	—
1088	1103	16	cips mx_cam_mode (295)	—	—
1104	1119	16	cips mx_cam_f_sync (293)	—	—
1120	1135	16	cips mx_cam_clear_ct (293)	—	—
1136	1151	16	cips mx_cam_tdi_rows (297)	—	—
1152	1167	16	cips mx_cam_tdi_time (298)	—	—
1168	1183	16	cips mx_cam_ss_time (297)	—	—
1184	1199	16	cips my_cam_mode (315)	—	—
1200	1215	16	cips my_cam_f_sync (313)	—	—
1216	1231	16	cips my_cam_clear_ct (313)	—	—
1232	1247	16	cips my_cam_tdi_rows (317)	—	—
1248	1263	16	cips my_cam_tdi_time (317)	—	—
1264	1279	16	cips my_cam_ss_time (317)	—	—
1280	1295	16	cips wind_sync_err_ct (478)	—	—
1296	1311	16	cips a_seq_page (141)	—	—
1312	1327	16	cips a_seq_addr (137)	—	—
1328	1343	16	cips a_seq_state (142)	—	—
1344	1359	16	cips a_seq_length (140)	—	—
1360	1375	16	cips a_seq_use_ct (144)	—	—
1376	1391	16	cips a_seq_cur_pos (140)	—	—
1392	1423	32	cips a_seq_wait_time (144)	—	—
1424	1439	16	cips a_seq_wait_type (145)	—	—
1440	1455	16	cips a_seq_stop_code (143)	—	—
1456	1471	16	cips a_seq_cmd_s_ct (139)	—	—
1472	1487	16	cips a_seq_cmd_f_ct (138)	—	—
1488	1503	16	cips a_seq_cmd_succ (140)	—	—
1504	1519	16	cips a_seq_cmd_fail (139)	—	—
1520	1535	16	cips a_seq_cmd_f_code (137)	—	—
1536	1551	16	cips o_seq_page (333)	—	—
1552	1567	16	cips o_seq_addr (329)	—	—
1568	1583	16	cips o_seq_state (334)	—	—
1584	1599	16	cips o_seq_length (333)	—	—
1600	1615	16	cips o_seq_use_ct (336)	—	—
1616	1631	16	cips o_seq_cur_pos (332)	—	—
1632	1663	32	cips o_seq_wait_time (336)	—	—
1664	1679	16	cips o_seq_wait_type (337)	—	—
1680	1695	16	cips o_seq_stop_code (335)	—	—
1696	1711	16	cips o_seq_cmd_s_ct (332)	—	—
1712	1727	16	cips o_seq_cmd_f_ct (331)	—	—
1728	1743	16	cips o_seq_cmd_succ (332)	—	—
1744	1759	16	cips o_seq_cmd_fail (331)	—	—
1760	1775	16	cips o_seq_cmd_f_code (330)	—	—
1776	1791	16	cips mem_dump_remain (270)	—	—
1792	1807	16	cips ee.write_enable (220)	—	—

Packet **cips sw_status** Format, cont'd

start bit	stop bit	size bits	field		
1808	1823	16	cips ctlm_rbuf_wds (201)	—	—
1824	1839	16	cips tlm_rbuf_wds (467)	—	—
1840	1855	16	cips cmd_rbuf_wds (175)	—	—
1856	1871	16	cips cde_tlm_rbuf_wds (159)	—	—
1872	1887	16	cips cde_cmd_rbuf_wds (155)	—	—
1888	1903	16	cips stack_util_max (461)	—	—
1904	1919	16	cips cpu_util (178)	—	—
1920	1935	16	cips cpu_util_max (178)	—	—
1936	1951	16	cips csci_version_num (200)	—	—
1952	1967	16	cips csci_version_pl (200)	—	—
1968	1983	16	cips active_csci (147)	—	—
1984	1999	16	cips boot_error (150)	—	—
2000	2031	32	cips tlm_words_sent (468)	—	—
2032	2047	16	cips ms1553_nt_wd.ct (288)	—	—
2048	2063	16	cips ms1553_ct_wd.ct (284)	—	—
2064	2079	16	cips ms1553_bit_f.ct (278)	—	—
2080	2095	16	cips ms1553_ei.ct (285)	—	—
2096	2111	16	cips ms1553_hw_r.ct (285)	—	—
2112	2127	16	cips ms1553_bit_s.ct (279)	—	—
2128	2143	16	cips ms1553_bit_f.wd (278)	—	—
2144	2159	16	cips ms1553_iiw_e.ct (286)	—	—
2160	2175	16	cips ms1553_iiw_e.wd (286)	—	—
2176	2191	16	cips ms1553_db_e.ct (284)	—	—
2192	2207	16	cips ms1553_c_b.o.ct (281)	—	—
2208	2223	16	cips ms1553_c.c.p.ct (281)	—	—
2224	2239	16	cips ms1553_nt_b.o.ct (287)	—	—
2240	2255	16	cips ms1553_nt.c.p.ct (288)	—	—
2256	2271	16	cips ms1553_ct.b.o.ct (282)	—	—
2272	2287	16	cips ms1553_ct.c.p.ct (283)	—	—
2288	2303	16	cips ms1553_b.a.pe.ct (274)	—	—
2304	2319	16	cips ms1553_b.a.me.ct (273)	—	—
2320	2335	16	cips ms1553_b.a.ce.ct (273)	—	—
2336	2351	16	cips ms1553_b.a.oe.ct (274)	—	—
2352	2367	16	cips ms1553_b.a.te.ct (275)	—	—
2368	2383	16	cips ms1553_b.b.pe.ct (277)	—	—
2384	2399	16	cips ms1553_b.b.me.ct (276)	—	—
2400	2415	16	cips ms1553_b.b.ce.ct (275)	—	—
2416	2431	16	cips ms1553_b.b.oe.ct (276)	—	—
2432	2447	16	cips ms1553_b.b.te.ct (277)	—	—
2448	2479	32	cips ms1553_ts.a.s.ct (289)	—	—
2480	2511	32	cips ms1553_ts.b.s.ct (289)	—	—
2512	2543	32	cips ms1553_tv.a.s.ct (290)	—	—
2544	2575	32	cips ms1553_tv.b.s.ct (290)	—	—
2576	2607	32	cips ms1553.c1.a.s.ct (279)	—	—
2608	2639	32	cips ms1553.c1.b.s.ct (280)	—	—
2640	2671	32	cips ms1553.c2.a.s.ct (280)	—	—
2672	2703	32	cips ms1553.c2.b.s.ct (280)	—	—

Packet **cips sw_status** Format, cont'd

start bit	stop bit	size bits	field		
2704	2735	32	cips ms1553_ac_a_s.ct (272)	—	—
2736	2767	32	cips ms1553_ac_b_s.ct (272)	—	—
2768	2799	32	cips ms1553_nt_a_s.ct (286)	—	—
2800	2831	32	cips ms1553_nt_b_s.ct (287)	—	—
2832	2863	32	cips ms1553_ct_a_s.ct (282)	—	—
2864	2895	32	cips ms1553_ct_b_s.ct (283)	—	—
2896	2911	16	cips cde_hb_fluttr.ct (155)	—	—
2912	2927	16	cips pm.enable (379)	—	—
2928	2943	16	cips exc.enable (239)	—	—
2944	2959	16	cips clock_tick.bits (171)	—	—
2960	2991	32	cips clock_tick.ssec (171)	—	—
2992	3007	16	cips ms.delay (291)	—	—
3008	3023	16	cips ee.write_ipd (220)	—	—
3024	3039	16	cips ms1553_to_thresh (289)	—	—
3040	3055	16	cips boot.ct_thresh (149)	—	—
3056	3071	16	cips quick_boot.flag (428)	—	—
3072	3087	16	cips sram_img_1_crc (441)	—	—
3088	3103	16	cips sram_img_1_len (442)	—	—
3104	3119	16	cips sram_img_1_pg (442)	—	—
3120	3135	16	cips sram_img_1_addr (441)	—	—
3136	3151	16	cips sram_img_2_crc (443)	—	—
3152	3167	16	cips sram_img_2_len (443)	—	—
3168	3183	16	cips sram_img_2_pg (444)	—	—
3184	3199	16	cips sram_img_2_addr (443)	—	—
3200	3215	16	cips sram_img_to_ld (445)	—	—
3216	3231	16	cips cde_init_flag (156)	—	—
3232	3247	16	cips crc.type (198)	—	—
3248	3263	16	cips crc.addr (197)	—	—
3264	3279	16	cips crc.length (197)	—	—
3280	3295	16	cips crc.valid (198)	—	—
3296	3311	16	cips crc (179)	—	—
3312	3327	16	cips crc_1.type (181)	—	—
3328	3343	16	cips crc_1.addr (180)	—	—
3344	3359	16	cips crc_1.length (180)	—	—
3360	3375	16	cips crc_1.valid (181)	—	—
3376	3391	16	cips crc_1 (180)	—	—
3392	3407	16	cips crc_2.type (183)	—	—
3408	3423	16	cips crc_2.addr (182)	—	—
3424	3439	16	cips crc_2.length (183)	—	—
3440	3455	16	cips crc_2.valid (184)	—	—
3456	3471	16	cips crc_2 (182)	—	—
3472	3487	16	cips crc_3.type (186)	—	—
3488	3503	16	cips crc_3.addr (185)	—	—
3504	3519	16	cips crc_3.length (185)	—	—
3520	3535	16	cips crc_3.valid (186)	—	—
3536	3551	16	cips crc_3 (184)	—	—
3552	3567	16	cips crc_4.type (188)	—	—

Packet **cips sw_status** Format, cont'd

start bit	stop bit	size bits	field		
3568	3583	16	cips crc_4_addr (187)	—	—
3584	3599	16	cips crc_4_length (188)	—	—
3600	3615	16	cips crc_4_valid (189)	—	—
3616	3631	16	cips crc_4 (187)	—	—
3632	3647	16	cips crc_5_type (191)	—	—
3648	3663	16	cips crc_5_addr (190)	—	—
3664	3679	16	cips crc_5_length (190)	—	—
3680	3695	16	cips crc_5_valid (191)	—	—
3696	3711	16	cips crc_5 (189)	—	—
3712	3727	16	cips crc_6_type (193)	—	—
3728	3743	16	cips crc_6_addr (192)	—	—
3744	3759	16	cips crc_6_length (193)	—	—
3760	3775	16	cips crc_6_valid (194)	—	—
3776	3791	16	cips crc_6 (192)	—	—
3792	3807	16	cips crc_7_type (196)	—	—
3808	3823	16	cips crc_7_addr (195)	—	—
3824	3839	16	cips crc_7_length (195)	—	—
3840	3855	16	cips crc_7_valid (196)	—	—
3856	3871	16	cips crc_7 (194)	—	—
3872	3887	16	cips crc_wpc (199)	—	—
3888	3888	1	cips em_u_test (233)	—	—
3889	3889	1	cips em_debug (224)	—	—
3890	3890	1	cips reserved_bit (431)	—	—
3891	3891	1	cips em_err (225)	—	—
3892	3892	1	cips em_tlm_err (232)	—	—
3893	3893	1	cips em_cmd_err (223)	—	—
3894	3894	1	cips em_seq_err (229)	—	—
3895	3895	1	cips em_seq_evt (230)	—	—
3896	3896	1	cips reserved_bit (431)	—	—
3897	3897	1	cips reserved_bit (431)	—	—
3898	3898	1	cips em_warn_msg (233)	—	—
3899	3899	1	cips reserved_bit (431)	—	—
3900	3900	1	cips reserved_bit (431)	—	—
3901	3901	1	cips reserved_bit (431)	—	—
3902	3902	1	cips em_seq_mode_chng (230)	—	—
3903	3903	1	cips em_mode_chng (227)	—	—
3904	3904	1	cips em_img_trigger (226)	—	—
3905	3905	1	cips em_img_int (225)	—	—
3906	3906	1	cips em_sdp_start (228)	—	—
3907	3907	1	cips em_sdp_stop (228)	—	—
3908	3908	1	cips reserved_bit (431)	—	—
3909	3909	1	cips reserved_bit (431)	—	—
3910	3910	1	cips reserved_bit (431)	—	—
3911	3911	1	cips em_seq_cmd_rx (229)	—	—
3912	3912	1	cips em_cde_cmd_tx (222)	—	—
3913	3913	1	cips em_cde_tlm_rx (222)	—	—
3914	3914	1	cips em_time_rx (231)	—	—

Packet **cips sw_status** Format, cont'd

start bit	stop bit	size bits	field		
3915	3915	1	cips em_acs_stat_rx (221)	—	—
3916	3916	1	cips em_tlm_tx (232)	—	—
3917	3917	1	cips em_cmd_rx (223)	—	—
3918	3918	1	cips em_pwr (227)	—	—
3919	3919	1	cips em_info (226)	—	—
3920	3935	16	cips tlm_pkts_per_blk (467)	—	—
3936	3936	1	cips reserved_bit (431)	—	—
3937	3937	1	cips reserved_bit (431)	—	—
3938	3938	1	cips reserved_bit (431)	—	—
3939	3939	1	cips reserved_bit (431)	—	—
3940	3940	1	cips reserved_bit (431)	—	—
3941	3941	1	cips reserved_bit (431)	—	—
3942	3942	1	cips reserved_bit (431)	—	—
3943	3943	1	cips reserved_bit (431)	—	—
3944	3944	1	cips reserved_bit (431)	—	—
3945	3945	1	cips cde_itf_to (157)	—	—
3946	3946	1	cips cde_flash_to (155)	—	—
3947	3947	1	cips cde_cal_to (154)	—	—
3948	3948	1	cips cde_hist_log_to (156)	—	—
3949	3949	1	cips cde_sys_data_to (159)	—	—
3950	3950	1	cips cde_mem_dump_to (158)	—	—
3951	3951	1	cips cde_sci_to (158)	—	—
3952	3952	1	cips reserved_4_to (430)	—	—
3953	3953	1	cips reserved_3_to (429)	—	—
3954	3954	1	cips reserved_2_to (429)	—	—
3955	3955	1	cips reserved_1_to (428)	—	—
3956	3956	1	cips sci_my_cam_to (432)	—	—
3957	3957	1	cips sci_mx_cam_to (432)	—	—
3958	3958	1	cips sci_py_cam_to (434)	—	—
3959	3959	1	cips sci_px_cam_to (433)	—	—
3960	3960	1	cips cmd_echo_to (171)	—	—
3961	3961	1	cips exception_to (239)	—	—
3962	3962	1	cips pm_to (383)	—	—
3963	3963	1	cips mem_dump_to (271)	—	—
3964	3964	1	cips evt_log_to (238)	—	—
3965	3965	1	cips err_to (234)	—	—
3966	3966	1	cips hw_hk_to (253)	—	—
3967	3967	1	cips sw_stat_to (464)	—	—
3968	3983	16	cips i_px_bos_level (257)	—	—
3984	3999	16	cips i_py_bos_level (257)	—	—
4000	4015	16	cips i_mx_bos_level (255)	—	—
4016	4031	16	cips i_my_bos_level (256)	—	—
4032	4047	16	cips i_l_htr_uthresh (254)	—	—
4048	4063	16	cips i_l_htr_lthresh (253)	—	—
4064	4079	16	cips i_cs_thresh (253)	—	—
4080	4095	16	cips parm_tbl_crc (375)	—	—
4096	4111	16	cips parm_tbl_flag (376)	—	—

Packet **cips sw_status** Format, cont'd

start bit	stop bit	size bits	field		
4112	4127	16	cips pkt_crc (378)	–	–

Packet size = 4128 bits (516 bytes) .

G

Telemetry Packet Summary

Telemetry Packet Summary

pkt no.	apid (hex)	packet
1	0x160 (352)	cips sw_status (514)
2	0x161 (353)	cips hw_hk (492)
3	0x162 (354)	cips err (481)
4	0x163 (355)	cips evt_log (482)
5	0x164 (356)	cips mem_dump (499)
6	0x165 (357)	cips post_mortem (500)
7	0x166 (358)	cips exception (484)
8	0x167 (359)	cips cmd_echo (480)
9	0x168 (360)	cips sci_px_cam (508)
10	0x169 (361)	cips sci_py_cam (511)
11	0x16a (362)	cips sci_mx_cam (502)
12	0x16b (363)	cips sci_my_cam (505)

G. TELEMETRY PACKET SUMMARY

H

Glossary

ACS *Attitude Control Subsystem* Spacecraft subsystem responsible for spacecraft and instrument pointing.

ADC *analog/digital converter*

Aeronomy of Ice in the Mesosphere NASA mission to study the formation, morphology, and life cycle of polar mesospheric clouds (PMCs). These clouds are also called noctilucent due to their tendency to 'glow' during the polar winter months. These clouds are so high in the atmosphere that their under sides (sides facing the earth) are illuminated by sunshine passing across the pole through the atmosphere from the daylit side of the Earth.

B *byte* See byte

b *bit* See bit.

bit Basic unit of information storage.

BOS *bright object sensor* Hardware for sensing presence of bright objects (such as the sun).

byte 8 bits of storage. CCSDS uses the term *octet* instead.

CCSDS *Consultative Committee for Space Data Systems* An [international organization](#) of space agencies interested in mutually developing standard data handling techniques.

CDE *Cosmic Dust Experiment* One of the instruments comprising the AIM observatory, the CDE instrument helps scientists determine the amount of particles entering the atmosphere. This in turn helps them determine the impact of cosmic dust on PMC formation.

C&DH *command and data handling* Spacecraft subsystem that receives and processes commands, and produces telemetry. Typically the CDH also contains the flight computer.

CIPS *Cloud Imaging and Particle Size* One of the instruments comprising the AIM observatory, the CIPS instrument uses six CCDs to capture grayscale images of PMCs. These images will help scientists determine how these clouds are created, how they change over time, and what constitutes them.

CM *configuration management* Process of controlling access, identifying and archiving designated releases of software and documentation. CM is complementary to a version control system. See CVS.

command An instruction from a user (or script) to a spacecraft system to perform a specific action.

FSW *flight software* Embedded software that controls flight hardware.

glossary This is it.

H. GLOSSARY

high level command An instrument command that implements functionality of more than one low level command. High level commands are more efficient than low level commands for specifying complex instrument behavior.

HVPS *high voltage power supply* Device designed to provide unusually high voltages to electronics. Such devices are not only a safety hazard, but also can be hazardous to other hardware, should they arc, which is only a danger at low ambient pressures.

I/O *input output*

KB *kilobyte* $2^{10} = 1024$ bytes.

Kb *kilobit* $2^{10} = 1024$ bits.

LASP *Laboratory for Atmospheric and Space Physics* Institute at the [University of Colorado, Boulder](#), researching atmospheric, space and planetary physics.

low level command An instrument command that implements a single basic hardware or software function.

MB *megabyte* $2^{20} = 1,048,576$ bytes.

Mb *megabit* $2^{20} = 1,048,576$ bits.

ms *milliseconds*

MIL-STD-1553 Military standard for flight serial data bus.

octet CCSDS term for an 8-bit unit of storage. See byte.

PDF *portable document format* Binary document format with self contained fonts that can be viewed or printed from most any platform via [Adobe Acrobat](#) and other PDF viewers. Advanced PDF viewers can navigate hyperlinks to other locations within and external to the document.

POR *power on reset* Hardware reset due to power cycle (transition from off to on). Hardware and software return to their POR states after a POR.

SSI *Standard Serial Interface* Interface electronics used for loading, monitoring, and debugging FSW.

TC *telecommand* CCSDS term for command. See CCSDS.

telemetry Data produced by a spacecraft application.

UT69R See UT69R000.

UT69R000 A radiation-hardened 16-bit RISC microcontroller created by UTMC. The 69R000 is the microcontroller host for FSW on all AIM instruments.

End of glossary.

References

- [1] David Gathright. *AIM Mission Software Development Plan*. Laboratory for Atmospheric and Space Physics, Boulder, CO, 2003. LASP document no. AIM-T-0901. 25, 28, 29
- [2] David Gathright. *CIPS Flight Software Development Plan*. Laboratory for Atmospheric and Space Physics, Boulder, CO, 2003. Defines the flight software development process. LASP document no. AIM-T-0910. 25
- [3] David Gathright. *CIPS Flight Software Design Description*. Laboratory for Atmospheric and Space Physics, Boulder, CO, 2005. LASP document no. CIPS-T-80002. 27, 29, 34, 35, 39, 40
- [4] David Gathright. *AIM Spacecraft - Instruments Communications ICD*. Laboratory for Atmospheric and Space Physics, 2003. The communications ICD defines the data interface between the AIM spacecraft and instruments. Document no. AIM-T-0900. 29, 33, 38
- [5] Applied Physics Laboratory. *New Horizons to SDC ICD*, 2003. This ICD defines the interfaces between the New Horizons spacecraft and the SDC instrument. CDE's data interface will be identical to the SDC data interface. 29
- [6] Laboratory for Atmospheric and Space Physics, Boulder, CO. *CIPS Level 3 Specification*, 2003. Defines the external interface requirements for CIPS. LASP document no. AIM-T-0XXX. 29
- [7] David Gathright. *CIPS Flight Software Requirements Specification*. Laboratory for Atmospheric and Space Physics, Boulder, CO, 2004. Defining document for FSW requirements and interfaces. LASP document no. CIPS-T-80003. 29
- [8] Laboratory for Atmospheric and Space Physics, Boulder, CO. *OASIS-CC System Manager's Guide*, 1992. The OASIS-CC describes the necessary information for tailoring OASIS-CC to a particular application and provides instructions for inserting that information into the OASIS-CC database. 32, 38