GLAST LAT Flight Software Demonstration

SLAC Campus
Building 84, Central Lab Annex

October 1, 2004
Overview Presentation 10:30 AM (Room B-259)
Demonstrations 10:45 AM

1 Demonstration Overview

1.1 Agenda for the Demonstrations

The demonstration will take place in the Central Laboratory Annex (Building 84), Room B-101.

Demo Agenda Item	Presenter(s)
1. Overview of the Demonstration (in Group C Conference Room, Room B-259)	Ed Bacho
2. Science Data Interface Demonstration	Steve Mazzoni
3. LAT Communication Board Driver Demonstration	Ed Bacho
4. Questions from Attendees	NA

Feel free to jot questions and comments down in the margins of this document or in the space provided on page 13.

1.2 Goals of the Demonstration

The September 2004 FSW Demonstration (actually held on October 1) covers two major areas of functionality:

The Science Data Interface. This portion of the demo showcases FSW's ability to deliver properly formatted and addressed data from an SIU through an Event Builder Module and out over the high-speed LVDS science data interface between the LAT and the Spacecraft. In flight, this data is bound for the Solid State Recorder, where it is stored until retrieved from the ground.

As in the August demonstration, the Spacecraft is represented by the Spacecraft Data Interface Simulator (SDIS), a simulator provided by Spectrum Astro, Inc. The LAT is represented by the Instrument to Spacecraft Interface Simulator (ISIS), which contains a single SIU along with primary and redundant GASU (with Command/Response Unit, CRU, and Event Builder Module, EBM) and primary and redundant PDU.

The LAT Communications Board (LCB) Driver. This portion of the demo shows the ability of the re-designed LCB driver to control LCB devices in separate CPU crates. It also demonstrates the test suite written to confirm correct operation and performance characteristics of this critical driver package. In the flight system, all LAT-internal communications are mediated by the LCB driver.

For today, 2 CPU crates, each containing an LCB board, will communicate as controlled by the LCB driver. This shows that the FSW development team has an infrastructure in place to begin the development and deployment of multi-CPU (SIU/EPU) systems.

1.3 Expected Outputs of the Demonstrations; Verification of Requirements

This section cites the expected outputs of each of the systems demonstrated today, and provides a space for the demonstration monitors to verify by signature that the demonstrations ran successfully.

Demo Agenda Item	Relevant FSW Requirements (SRS Version 3)	Expected Output of the Demonstration
Science Data Interface Demonstration	5.2.1.3.1, 5.2.1.3.3, 5.2.1.3.4, 5.2.1.4	For a full description of the outputs of the Science Data Interface Demo, see Section 1.3.1starting on page 3. For the demo's software and hardware context and the demonstration procedure, see Section 2.1.1 on page 9, and Section 2.2 on page 11.
		Monitor Signature: ______________________
LAT Communications Board Demonstration	5.3.17.1, 5.3.17.3	For a full description of the outputs of the LCB Driver Demo, see Section 1.3.2 starting on page 6. For the demo's software and hardware context and the demonstration procedure, see Section 2.1.2 on page 10, and Section 2.3 on page 11.
		Monitor Signature: ______________________

1.3.1 Output of the Science Data Interface Demo

The Science Data Interface Demo produces AstroRT display output.

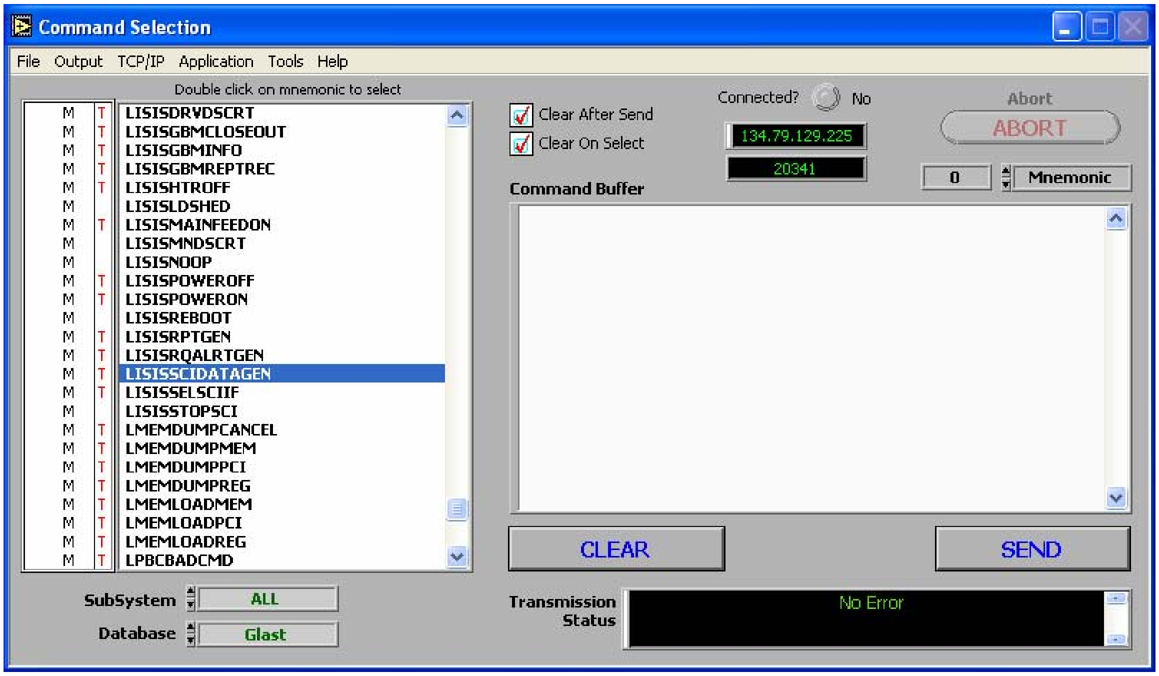
When 1553 communications between the SDIS and the ISIS have been established, the demonstrator uses AstroRT to display the status of the 1553 communications link and to display contents of telemetry and other data exchanged between the two systems. After issuing telecommands to define the type of science data sent out over the LVDS and to initialize the LVDS interface itself, science data packets are sent out. See Section 2.2 for the step-by-step procedure.

Five types of displays show how this sequence of steps is completed:

- AstroRT Command Selection window
- AstroRT Enter Parameter Values window
- AstroRT LVDS DAQ Main control window
- AstroRT LVDS Raw Frame display window. A separate Raw Frame window is opened each time a new science data pattern is selected for output.
- The demonstrator also refers to a VxWorks terminal display to display statistics on the science data traffic as it sent from the SIU to the EBM.

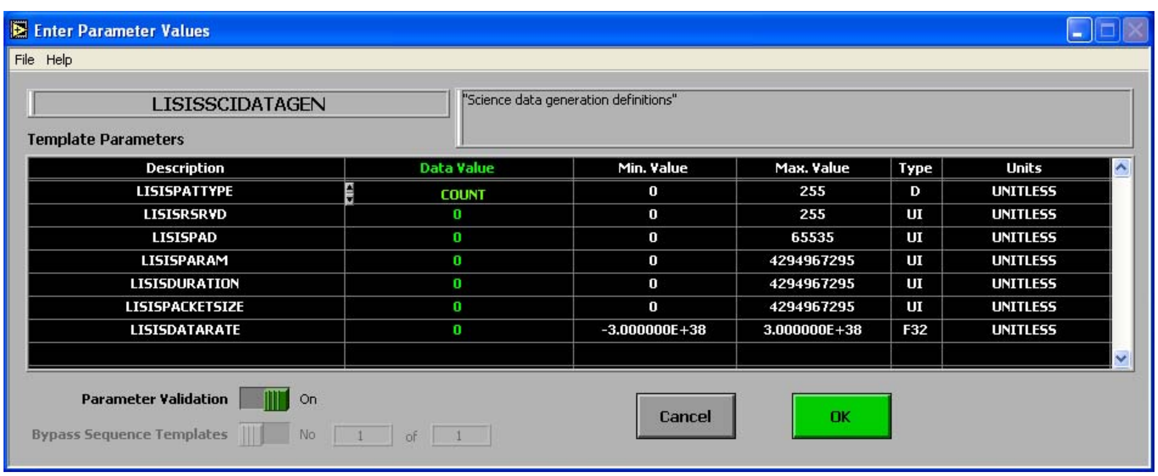
Samples of these 5 displays are shown in Figure 1 through Figure 5.

Figure 1: Command Selection Window (AstroRT)



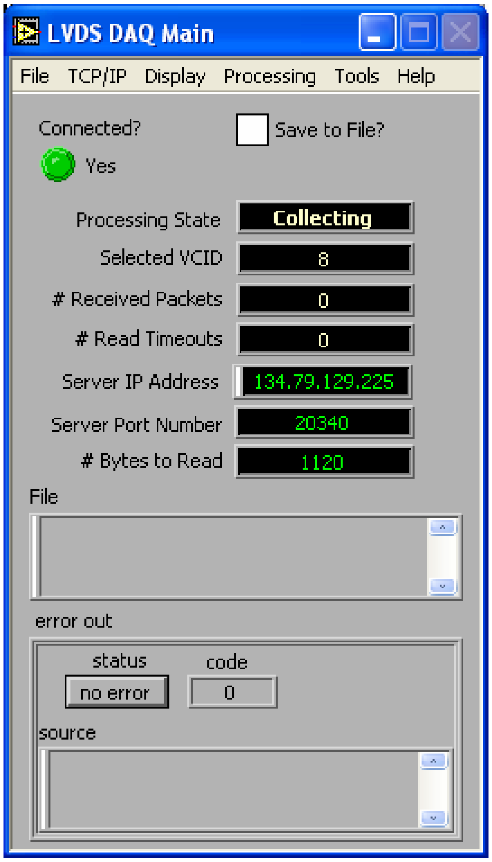
In Figure 1, note that the LISISSCIDATAGEN command is selected. The demonstrator will set parameters for this command in the AstroRT Enter Parameter Values window (next figure), then transmit the command using the SEND button.

Figure 2: Enter Parameter Values Window (AstroRT)



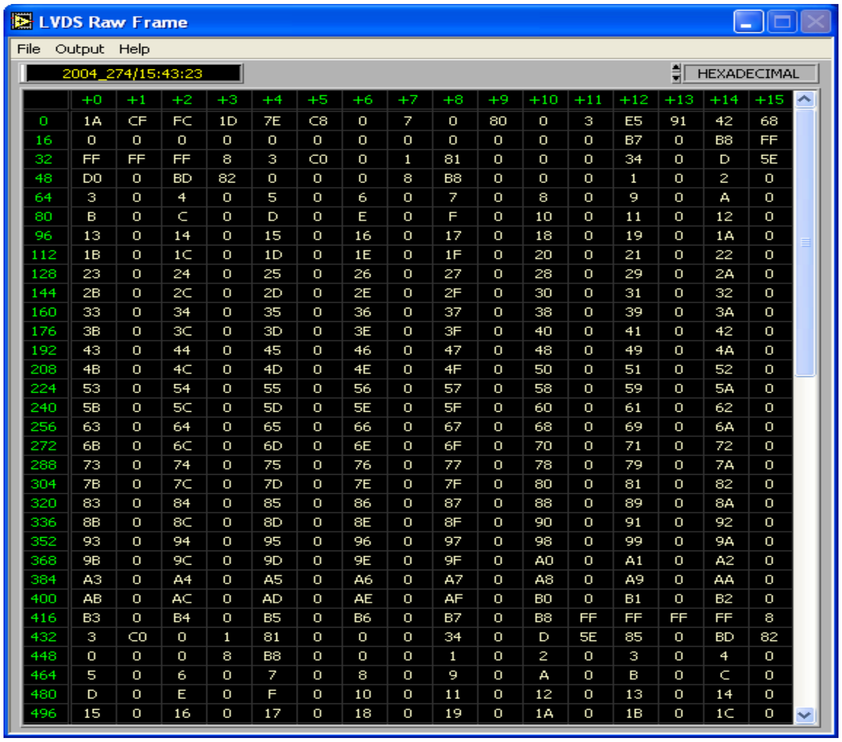
As shown in Figure 2, the demonstrator can set the pattern type (LISISPATTYPE), transmission duration (LISISDURATION), packet size (LISISPACKETSIZE), and other parameters of the LISISSCIDATAGEN command as desired.

Figure 3: LVDS DAQ Main Control Window (AstroRT)



From the Display menu in the LVDS DAQ Main window (above), the demonstrator selects the LVDS Raw Frame display window, which then opens (below).

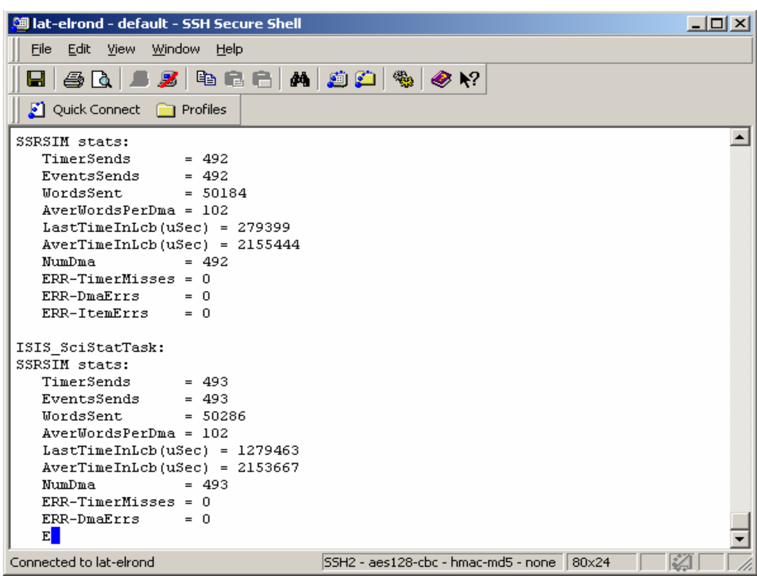
Figure 4: LVDS Raw Frame Display Window – Continuous Pattern (AstroRT)



	+0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11	+12	+13	+14	+15
0	1A	CF	FC	1D	7E	C8	0	7	0	80	0	3	E5	91	42	68
16	0	0	0	0	0	0	0	0	0	0	0	0	B7	0	B8	FF
32	FF	FF	FF	8	3	C0	0	1	81	0	0	0	34	0	D	5E
48	D0	0	BD	82	0	0	0	8	B8	0	0	0	1	0	2	0
64	3	0	4	0	5	0	6	0	7	0	8	0	9	0	A	0
80	B	0	C	0	D	0	E	0	F	0	10	0	11	0	12	0
96	13	0	14	0	15	0	16	0	17	0	18	0	19	0	1A	0
112	1B	0	1C	0	1D	0	1E	0	1F	0	20	0	21	0	22	0
128	23	0	24	0	25	0	26	0	27	0	28	0	29	0	2A	0
144	2B	0	2C	0	2D	0	2E	0	2F	0	30	0	31	0	32	0
160	33	0	34	0	35	0	36	0	37	0	38	0	39	0	3A	0
176	3B	0	3C	0	3D	0	3E	0	3F	0	40	0	41	0	42	0
192	43	0	44	0	45	0	46	0	47	0	48	0	49	0	4A	0
208	4B	0	4C	0	4D	0	4E	0	4F	0	50	0	51	0	52	0
224	53	0	54	0	55	0	56	0	57	0	58	0	59	0	5A	0
240	5B	0	5C	0	5D	0	5E	0	5F	0	60	0	61	0	62	0
256	63	0	64	0	65	0	66	0	67	0	68	0	69	0	6A	0
272	6B	0	6C	0	6D	0	6E	0	6F	0	70	0	71	0	72	0
288	73	0	74	0	75	0	76	0	77	0	78	0	79	0	7A	0
304	7B	0	7C	0	7D	0	7E	0	7F	0	80	0	81	0	82	0
320	83	0	84	0	85	0	86	0	87	0	88	0	89	0	8A	0
336	8B	0	8C	0	8D	0	8E	0	8F	0	90	0	91	0	92	0
352	93	0	94	0	95	0	96	0	97	0	98	0	99	0	9A	0
368	9B	0	9C	0	9D	0	9E	0	9F	0	A0	0	A1	0	A2	0
384	A3	0	A4	0	A5	0	A6	0	A7	0	A8	0	A9	0	AA	0
400	AB	0	AC	0	AD	0	AE	0	AF	0	B0	0	B1	0	B2	0
416	B3	0	B4	0	B5	0	B6	0	B7	0	B8	FF	FF	FF	FF	8
432	3	C0	0	1	81	0	0	0	34	0	D	5E	85	0	BD	82
448	0	0	0	8	B8	0	0	0	1	0	2	0	3	0	4	0
464	5	0	6	0	7	0	8	0	9	0	A	0	B	0	C	0
480	D	0	E	0	F	0	10	0	11	0	12	0	13	0	14	0
496	15	0	16	0	17	0	18	0	19	0	1A	0	1B	0	1C	0

The demonstrator may use the following VxWorks terminal window (Figure 5) to display statistics generated by the SciStatTask.

Figure 5: Terminal Output of ISIS_SciStatTask




```
SSRSIM stats:
   TimerSends       = 492
   EventsSends      = 492
   WordsSent        = 50184
   AverWordsPerDma = 102
   LastTimeInLcb(uSec) = 279399
   AverTimeInLcb(uSec) = 2155444
   NumDma           = 492
   ERR-TimerMisses = 0
   ERR-DmaErrs      = 0
   ERR-ItemErrs     = 0

ISIS_SciStatTask:
SSRSIM stats:
   TimerSends       = 493
   EventsSends      = 493
   WordsSent        = 50286
   AverWordsPerDma = 102
   LastTimeInLcb(uSec) = 1279463
   AverTimeInLcb(uSec) = 2153667
   NumDma           = 493
   ERR-TimerMisses = 0
   ERR-DmaErrs      = 0
   E
```

1.3.2 Output of the LCB Driver Demo

The LCB Driver Demo demonstration is conducted in two major pieces:

- Loop back testing, including stress testing
- CPU-to-CPU communication testing

Each portion of the agenda generates terminal window output, show in Figure 6 through Figure 8 in the following sections.

1.3.2.1 Output of LCB Driver Loop Back Tests

The loop back test suite executes several tests using a single LCB board-equipped crate, including tests of ability to read local PCI registers reads, read local command/response registers, read off-board command/response registers, and send/receive event data. The terminal window in Figure 6 shows the terminal output that corresponds to a stress test using 5 tasks to send out a high volume of random pattern event data and verify error-free transmission of events.

Figure 6: Selected Terminal Output from LCB Driver Loop Back Test

```
Multiple Task Test - 5 tasks doing Loopback Event - TDATA_DTEST_RANDOM
LCBT_sendEvent():cl=0x7eefc00  rl=0x7eeebf0
LCBT_sendEvent():cl=0x7eeda00  rl=0x7eec9f0
LCBT_sendEvent():cl=0x7eeb800  rl=0x7eea7f0
LCBT_sendEvent():cl=0x7ee9600  rl=0x7ee85f0
LCBT_sendEvent():cl=0x7ee7400  rl=0x7ee63f0
eventTest (5 task) -  dataTest=TDATA_DTEST_RANDOM  parm=0x9abc
  Events Sent:      num=500  words=176940  done=500
  Events Recv:      num=500  words=184000  truncated=0
  Time: 0.559 sec  SendBW=1266 KBs RecvBW=1316 KBs
  Events:  num=1800 rErr=0 wErr=0 cErr=0
```

1.3.2.2 Output of LCB Driver CPU-to-CPU Transmission Tests

The CPU-to-CPU test shows that event data transmitted by the LCB driver in one crate (the “sender” crate) via the GASU is successfully received by the LCB driver in the second crate (the “receiver” crate).

This test can be run in several configurations. The following two figures show the terminal output from the sender and receiver crates when a task running on the sender (Figure 7) generates and sends event data to the receiver crate (Figure 8).

Figure 7: Selected Terminal Output from Sender Crate During CPU-to-CPU Demonstration

```
-> LCBT_eventTest 10,0,0x24
LCBT_sendEvent():cl=0x7eefc00  rl=0x7eeebf0
eventTest (1 task) -  dataTest=TDATA_DTEST_INCREMENT  parm=0x0
  Events Sent:      num=10  words=7640  done=10
  Events Recv:      num=0  words=0  truncated=0
  Time: 0.015 sec  SendBW=2037 KBs RecvBW=   0 KBs
  Events:  num=300 rErr=0 wErr=0 cErr=0
```

Figure 8: Selected Terminal Output from Receiver Crate During CPU-to-CPU Demonstration

```
-> LCBT_setOption 2
value = 2 = 0x2
-> EvtRecvCb:num=0 len32=0x300 evtDesc=0x6000000 evtOff=0x80000000 free=0x0
EvtRecvCb:num=1 len32=0x300 evtDesc=0x6000308 evtOff=0x80000c20 free=0x308
EvtRecvCb:num=2 len32=0x300 evtDesc=0x6000610 evtOff=0x80001840 free=0x610
EvtRecvCb:num=3 len32=0x300 evtDesc=0x6000918 evtOff=0x80002460 free=0x918
EvtRecvCb:num=4 len32=0x300 evtDesc=0x6000c20 evtOff=0x80003080 free=0xc20
EvtRecvCb:num=5 len32=0x300 evtDesc=0x6000f28 evtOff=0x80003ca0 free=0xf28
EvtRecvCb:num=6 len32=0x300 evtDesc=0x6001230 evtOff=0x800048c0 free=0x1230
EvtRecvCb:num=7 len32=0x300 evtDesc=0x6001538 evtOff=0x800054e0 free=0x1538
EvtRecvCb:num=8 len32=0x300 evtDesc=0x6001840 evtOff=0x80006100 free=0x1840
EvtRecvCb:num=9 len32=0x300 evtDesc=0x6001b48 evtOff=0x80006d20 free=0x1b48
```

2 Demonstration Procedure

2.1 Context of the Demonstrations

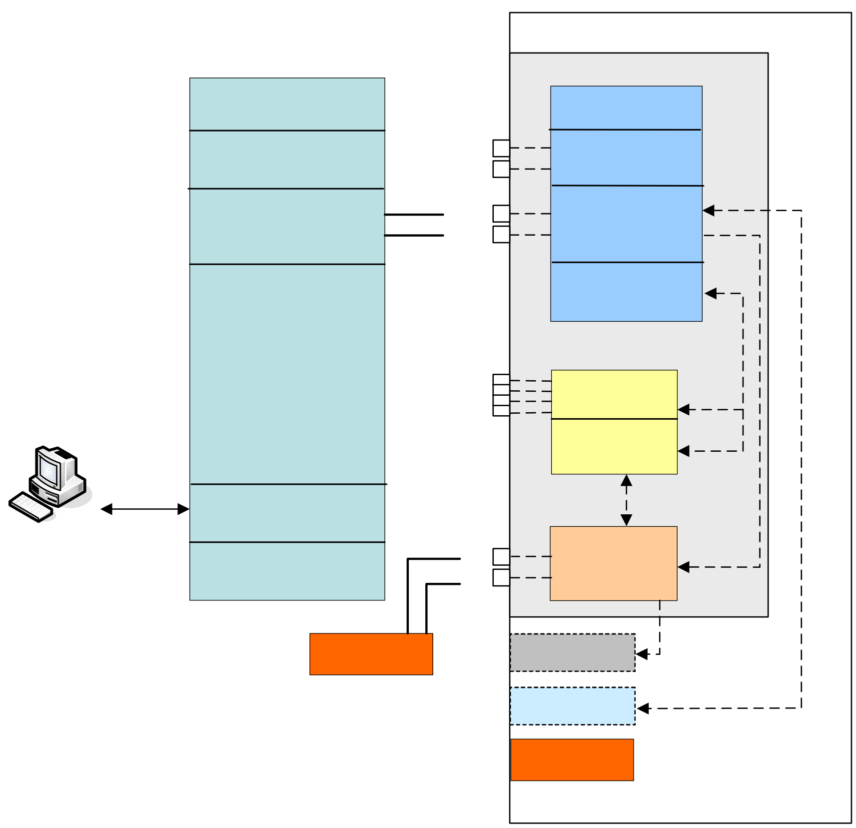
2.1.1 Context of the Science Data Interface Demo

The Science Data Interface Demo involves two major collections of hardware: the ISIS and the SDIS. These are set up in the Dataflow laboratory (Room B-101).

2.1.1.1 The Instrument to Spacecraft Interface Simulator (ISIS)

The ISIS is made up of engineering models of flight hardware. The distinctions between the ISIS SIU, PDU, and GASU hardware and flight-qualified parts are very few. With the exception of the sole SIU, the remaining pieces of the core LAT hardware are fully redundant. The structure of the ISIS is shown in Figure 9.

Figure 9: Science Data Interface Demonstration: ISIS and SDIS Hardware



2.1.1.2 The Spacecraft Data Interface Simulator (SDIS)

The SDIS represents significant portions of the interface between the LAT and the Spacecraft. The SDIS comprises one side of the SC C&DH system with a 1553 bus, LVDS Science Interface, and Discrete monitors and controls.

A PC, connected to the SDIS via Ethernet, is used to control the Simulator. The PC runs Spectrum Astro's proprietary Spacecraft command and control package, AstroRT.

2.1.1.3 FSW Context of the Science Data Interface Demo

The demonstrator uses the **lato.vxw** script to load the following FSW libraries (all production versions as of September 29, 2004)::

RAD750/ librad750_reboot.o, CMX/libcmx_asBuilt.o, CMX/libcmx_asBuiltSpy.o, PBS/ libpbs.o, MSG/libmsg_mt.o, MSG/msg_print/libmsg_print.o, ZLIB/libzlib_inflate.o, FILE/libfile_hdr.o, CCSDS/libccsds_pkt.o, CCSDS/libccsds_dump.o, CCSDS/libccsds_swap.o, SIB/libsib.o, CTDB/libsumt_rt_sib.o, ITC/libitc_dump.o, ITC/libitc.o, CTS/libctx_lcp_sumt.o, CTS/libcts_lcp.o, LCB/liblcbd.o, LCB/libliox.o, LCB/libt_adr.o, DEM/libtem_reg.o, DEM/libaem_reg.o, DEM/libltem_to.o, DEM/liblaem_to.o, DEM/libltem.o, DEM/liblaem.o, DEM/libsaem.o, DEM/libstem.o, DAB/libdab_to.o, DAB/libdab_reg.o, DAB/libdab.o, DAB/libsgem.o, DAB/libspdu.o, DAB/libscru.o, DAB/libsebm.o, DAB/liblcru.o, DAB/liblgem.o, DAB/liblpdu.o, LHK/liblhk.o, PIG/libpig_power.o, PIG/libpig_bacon.o, /LCBT/libssrsimOld.o, LCBT/liblcbtOld.o, LSM/liblsm_dump.o, LSM/liblsm.o, LSM/liblsm_scp.o, isisLAT/libisisLAT.o

The new LCB driver package is NOT used in the science data interface demo. The libssrsimOld.o and libisisLAT.o libraries are ISIS-specific.

2.1.2 Context of the LAT Communications Board Demo

2.1.2.1 Hardware Context

The LCD Driver demo is performed at one of the corner test benches in the Dataflow laboratory (Room B-101)

The test bench is configured as follows for the demonstration:

- A GASU containing a Command/Response Unit and Event Builder Module.
- Two VME crates, each with an mv2304 processor. Each crate contains a PMC LAT Communications Board and an Ethernet board. The crates are connected through their LAT Communications Boards via the GASU.
- A PC is connected via Ethernet to both crates. This PC is used to run the Tornado windSh, from which commands to load FSW and run the LCB demonstrations are issued.

2.1.2.2 FSW Context of the LCB Driver Demo

The demonstrator loads FSW libraries for the demo using the CMX tool over an Ethernet connection. The following FSW libraries (production versions as of September 29, 2004) are loaded for each portion of the demo:

PBS/libpbs.o, MSG/libmsg_mt.o, MSG/libmsg_print.o, CCSDS/libccsds_pkt.o, LCBD/liblcbd.o, LCBT/liblcbt.o

2.2 Science Data Interface Demonstration Procedure

This section lists the steps taken during the science data interface demonstration. For the windows accessed and the outputs displayed in those windows, see Section 1.3.1.

Before the science data interface demonstration session begins, the demonstrator performs the following steps:

- Boots the PC controlling the SDIS and launches LabView and AstroRT software.
- Opens up any AstroRT windows needed to support the demonstration.
- Turns power on to the SDIS and ISIS.
- Takes the SIU onboard the ISIS through the primary and secondary boot process (the FSW team has shown the boot process in earlier demonstrations; consult earlier demonstration guidebooks for details)
- Using VxWorks Tornado, creates a target server to access a WindSh terminal window to the SIU on the ISIS.
- From the WindSh terminal, runs the **lato.vxw** script to load FSW libraries needed for the demo (see above for library list).

2.2.1 Science Data Demo Procedure

With the hardware and software configured and initialized as discussed above, the demonstration proceeds as follows:

- At the WindSh prompt, the demonstrator issues the **LCP_init** command to initialize communications between the ISIS and the SDIS, then start up the PBS and MSG services.
- In the AstroRT Command Selection Window, issues the ISISMAINFEEDON CCSDS command to power up the ISIS GASU and PDU and launch the Housekeeping system. Only the primary GASU and PDU are powered for the demonstration.
- In the AstroRT Command Selection Window, issues LISISSCIDATAGEN CCSDS command. The Enter Parameter Values window appears, in which the demonstrator configures which science data pattern is to be sent (constant, counting, or random), the packet size (100), and the rate (1 Hz).
    - After the command is issued, the SciStatTask dumps statistics on event data transmission from the SIU to the WindSh terminal.
- In the AstroRT Command Selection Window, the demonstrator issues three CCSDS commands to set up the SDIS side of the LVDS interface: SDILVDSRESET, SDILVDSFLUSH, and SDILVDSENABLE.
- From the Display menu in the LVDS DAQ Main window, the demonstrator opens an LVDS Raw Frame window to display the contents of each different type of science data stream.

2.3 LAT Communications Board Driver Demonstration Procedure

This section lists the steps taken during the LCD Driver Demonstration. For the outputs of this demo, see Section 1.3.2.

The LCB Driver Demo demonstration is conducted in two major pieces:

- Loop back testing, including stress testing
- CPU-to-CPU communication testing

Before either of the two parts of the demo begin, the demonstrator performs the following steps:

- Powers up the PC, GASU, and crates.
- Loads FSW libraries to each crate (see above for library list).
- Initializes the PBS and MSG systems on each crate.
- Sets separate LATp addresses for the “sender” and “receiver” crates using a call to **LCBT_init** command. The addresses are “0x22” for the sender and “0x24” for the receiver.
- Sets the LCB driver on each crate to use byte-wide communications using a call to **LCBT_writeLatpCsr**.
- Initializes interaction with the GASU with a call to **LCBT_determineHw**.

2.3.1 LCB Driver Demo Loop Back Test Procedure

With the hardware and software configured and initialized as discussed above, the demonstration proceeds as follows:

- On the sender crate, the demonstrator calls **LCBT_testAll** to run the full test suite and reviews results of the various tests performed.

2.3.2 LCB Driver Demo CPU-to-CPU Test Procedure

This part of the demonstration proceeds as follows:

- On the receiver crate (address “0x24”), the demonstrator calls **LCBT_eventRcv** to prepare the LCB driver to receive data.
- On the sender crate, the demonstrator calls **LCBT_eventTest 10,0,0x24** to send 10 events to the receiver LCB.
- On the sender crate, the demonstrator calls **LCBT_eventTest 10000,0,0x24** to send 10000 events to the receiver LCB.
- On the sender crate, the demonstrator calls **LCBT_eventTest5 10000,0,0x24** to start 5 tasks on the sender, each task sending 10000 events to the receiver LCB.

3 Demo Wrapup and Summary

The FSW Team thanks you for attending the demonstration and welcomes any questions about the software and hardware systems showcased. Comments or questions can be written in the following space.

4 Glossary

1553 – MIL-STD-1553B. Serial data bus specification; in particular, the serial data bus and data protocol implemented for the GLAST mission.

APID. CCSDS packet application identifier. A numerical code indicating the general type of data in a CCSDS packet.

Crate. Fond, generic term for development versions of Spacecraft Interface Units (SIUs) or Event-Processor Units (EPUs): custom-built, standalone on-board FSW processors and communications hardware units that control the LAT and communicate with the spacecraft (SIU), and process/filter instrument events (EPU). Crates are used for development purposes and will be replaced in the flight unit with single board computers with the same functionality.

GASU (Global-Trigger/ACD-EM/Signal-Distribution Unit). Portion of the FSW hardware suite that serves as the major hardware interface between data acquisition electronics on the LAT and other hardware and electronics that make up the FSW hardware package. The GASU contains the GEM, EBM, AEM, and CRU.

HKP. Real-time housekeeping telemetry data; telemetry data which relates to the health and safety of the LAT instrument.

LCB – The LAT Communications Board. A cPCI board that allows the internal components of the LAT to communicate with one another.

PID – Programmable Discrete. The RAD750 CPU board contains 32 channels of digital I/O. The primary boot code uses two channels configured as outputs.

PDU (Power Distribution Unit). Portion of the FSW hardware suite that manages power distribution from the spacecraft and monitors the health of other FSW hardware.

RTOS – Real Time Operating System. In particular the VxWorks 5.4 operating system used by the LAT.

SC – The GLAST Spacecraft. As built by Spectrum Astro. Refer to the GLAST LAT Instrument – Spacecraft Interface Control Document for the formal specifications of the SC as seen by the LAT.

SDIS – Spacecraft Data Interface Simulator. The SDIS represents significant portions of the interface between the LAT and the Spacecraft. The SDIS comprises one side of the SC C&DH system with a 1553 bus, LVDS Science Interface, and Discrete monitors and controls.

SDRAM. The RAD750 CPU board 128 MB of synchronous DRAM; the SDRAM serves as the RAD750 main memory.

SIB – Spacecraft Interface Board. The board in the SIU crates that contains the LAT 1553 remote terminal hardware.

SIU – Spacecraft Interface Unit. A type of single board computer (SBC) in the FSW hardware suite that acts as an interface between the spacecraft and the LAT.

SUROM – Startup ROM. 256 KB of EEPROM memory on the RAD750 CPU boards that holds the primary boot code; the SUROM is only programmable on the bench through the PPCI JTAG interface.

T&DF (Trigger and Dataflow System). Large LAT subsystem that provides gamma-ray identification, readout of the detector measurements, assembly of gamma-ray source location and energy measurements, and the streaming of data to the spacecraft. The T&DF subsystem contains the TKR, CAL, and ACD front-end electronics, Tower Electronics Modules (TEMs), ACD Electronics Modules (AEMs), Event Builder Module (EBM), Global Trigger (GLT), Global Trigger Electronics Module (GEM), and CPUs used for instrument configuration and data processing.

VxWorks. Computer operating system used on board the RAD750 processor.