CHANDRA
X-ray Center 60 Garden St., Cambridge Massachusetts 02138 USA

MEMORANDUM

Date: August 7, 2009
From: Joseph DePasquale
To: ACIS team
Subject: ACIS Realtime Software

Abstract

This memo is intended to consolidate the information related to the ACIS team's real-time telemetry processing software. Here you will find the directory locations and lists of all software used to create the ACIS realtime hardware engineering and software web pages as well as the ACIS fluence monitoring scripts. Most of the information presented in this memo is also available in a convenient graph located at: http://cxc.harvard.edu/acis/memos/webpage/doc/ and reproduced at the end of this memo.

1 Redundancies

The ACIS realtime software suite runs independently on 4 different machines on 2 networks. The "primary" software runs on the machine "RHODES" which is actually connected to both the "OPS" and "HEAD" LANs. Our first-string backup machine is "COLOSSUS" which runs solely on the "OPS" LAN and the second-string backup is "XCANUCK" which runs solely on the "HEAD" LAN. Our Back-up OCC machine, located at Cambridge Discovery Park on the "HEAD" LAN is luke (NOTE that luke is normally not receiving telemetry and the real-time processes are only turned on when the Back-up OCC is needed). Each machine receives Chandra telemetry data through a "User Datagram Protocol" or "UDP." The ports now in use are listed below along with the machine that provides the data feed to that port, and the processes that use data from that port.

Description	UDP #	FEED	Process
primary (rhodes)	11350	gamera	ACORN
backup1 (colossus)	11151	gamera	CATNRT
backup1 (colossus)	11350	forbin	ACORN
backup2 (xcanuck)	11113	crosby	CATNRT
backup2 (xcanuck)	11112	forbin	ACORN
backup3 (luke)	5978	cdpmon	CATNRT
backup3 (luke)	5979	cdpmon	ACORN

2 Mount Points

The subject of mount points becomes an issue whenever one of the machines serving those directories experiences a network/power interruption or disk failure. The table below shows which machines serve the directories where the realtime processes are run. For example, if the machine "remus" were to shut down or experience a disk failure, the /home/acisdude directory would become unavailable on both the primary(rhodes) and backup2(xcanuck) machines. The realtime processes on colossus would continue to function however, since /home/acisdude is mounted directly from colossus on that machine. During testing of the Back-up OCC, we arranged with syshelp to have the /home/acisdude mount point changed from "remus" to "numa" in the event of "remus" outage. It is now part of the syshelp standard protocol to arrange to have the /home/acisdude area moved during a network outage affecting "remus".

Description	Dir Root	Mount Point
primary (rhodes)	/export	rhodes
primary (rhodes)	/home/acisdude	remus
primary (rhodes)	/data/mta4	rom-48
backup1 (colossus)	/home/acisdude	colossus
backup1 (colossus)	/data/asc1	asc1
backup2 (xcanuck)	/proj/sot	rom-48
backup2 (xcanuck)	/proj/web-cxc	numa
backup2 (xcanuck)	/home/acisdude	remus
backup3 (luke)	/export/acisops	luke
backup3 (luke)	/proj/web-cxc	numa

3 Fluence Monitoring

The directories where fluence monitoring is maintained are indicated in the table below. Listed below the table are the fluence monitoring scripts that are run via cron on all three machines listed above, along with a brief description of the script. Note again that although the machine luke is configured to run FLU-MON, it is not normally running and is only turned on when needed for testing or an emergency.

primary (rhodes):	/export/acis-flight/FLU-MON
backup1 (colossus):	/home/acisdude/FLU-MON
backup2 (xcanuck):	/proj/sot/acis/FLU-MON
backup3 (luke):	/export/acisops/FLU-MON

ace-flux.pl - was originally written by Robert Cameron to integrate all of the ACE/EPAM channels irrespective of what instrument was in the Chandra focal plane and regardless of whether the gratings were deployed or not. Shanil Virani then modified this script. The purpose of this script is to produce the latest ACE flux measurement (5-min sample) for all the ACE/EPAM channels. This is achieved by retrieving the "ace_epam_5m.txt" file from the SEC FTP site via lynx. The script retrieves this file and determines the last valid ACE/EPAM flux measurement. It does this by requiring that all the ACE EPAM fluxes must be positive since "status flags" are not reliable. It also reads in a Chandra ephemeris file produced and maintained by Tom Aldcroft and located in /proj/rac/ops/ephem/gephem.dat. If the Chandra altitude is greater than 55000 km, it then writes the latest flux measurements for all the EPAM channels (both electrons and protons) to an output file ("ACEflux.dat") for use by either acis-fluence_P3pscaling.pl. If the altitude is less than 55000 km, it integrates zero flux for all of the ACE EPAM channels as Chandra is in the radiation zones. Therefore, two fluence measures are available for each orbit. One that folds in which instrument is in the focal plane as well as the gratings history (acis-fluence_P3pscaling.pl and acis-fluence_FP6pscaling.pl), and one that ignores what instrument and whether the gratings are deployed or not.

acis-fluence_P3pscaling.pl - is a perl script that reads in the output file produced by ace-flux.pl, ACE-flux.dat, as well as reading in the instrument focal plane and OTG history files (FPHIST-2001.txt and GRATHIST-2001.txt, respectively) produced by acis-backstop.pl. It multiplies the latest ACE/EPAM flux measurement found in ACE-flux.dat by the following "attenuation factors": 0 if the HRC is in the focal plane, 1 if ACIS is in the focal plane, 0.2 if the HETG are deployed, 0.5 if the LETG are deployed, and 1 if neither gratings are deployed. This script is run every 5 minutes via CRON so it, therefore, keeps a running sum of the "ACIS" fluence as determined by the ACE/EPAM fluxes for each orbit. Specifically, this script monitors the ACE/EPAM P3 channel. If the fluence (ie, the "running sum") in this channel exceeds 1E9, alerts are issued to various distribution lists depending upon which "version" is in violation. There are three versions currently in operation. The "primary" version, run out of /export/acis-flight/FLU-MON/ and is only visible on rhodes, sends alerts to "sot_red_alert" and "acisdude". The "back-up" versions, run out of /proj/sot/acis/FLU-MON/ and visible from any HEAD LAN machine and /home/acisdude/FLU-MON/ on colossus (only visible on OPS LAN), send their alerts to only "acisdude"; there is also a seldom used version of FLU-MON run on luke out of /export/acisops/FLU-MON which also only sends its messages to "acisdude." Thus, the latter are truly "back-up" processes to make sure we are not susceptible to a single-point failure in the event of a network going down, disk crashes, etc. Moreover, this script reads in the Chandra ephemeris file located in /proj/rac/ops/ephem/gephem.dat which is produced and maintained by Tom Aldcroft. It reads this file to determine the transition from "descending" to "ascending" (via the log file cxodirect.dat) – ie, to determine when Chandra has crossed perigee. When it has, the script resets all fluences to zero and writes the fluence for all EPAM channels to a fluence archive (ACIS-fluence.arc) file. The script also reads in the DSN schedule located in /proj/rac/ops/ephem/dsn_summary.dat which is also produced and maintained by TST. If an alert is issued, the text message includes the times for next 3 DSN passes. The script keeps track of alerts spawned in an orbit by keeping a log in falert.dat. This file also gets reset at perigee. Lastly, the ACIS fluences for all the ACE/EPAM channels are output to a text file – ACIS-FLUENCE.dat.

ace_fluence.pl - is a simple perl module that reads in the output file produced by acis-fluence_P3pscaling.pl, ACIS-FLUENCE.dat, and produces an output file that displays all the ACIS ACE/EPAM fluences in scientific notation – current.dat. It also reads in the ACIS ACE/EPAM-derived orbital fluences, ACIS-fluence.arc, and produces two output files, protons.dat and all.dat, which presents a running sum of the total ACE/EPAM-P3 derived ACIS fluence (in scientific notation) since we started monitoring ACE (May 1, 2000). The latter file, all.dat, is the almost the same as the former except that it includes all the ACE/EPAM channels.

3.1 ACIS Realtime Engineering Webpage

The directories where the realtime software is maintained are indicated in the table below. Listed below the table are the realtime webpage scripts that are run via cron on all three machines listed above, along with a brief description of the script.

Binary directories	
primary (rhodes):	/export/acis-flight/primary/acis/bin
backup1* (colossus):	/home/acisdude/real-time/back-up/acis/bin
backup2 (xcanuck):	/home/acisdude/real-time/back-up/acis/bin
backup3 (luke):	/export/acisops/real-time/back-up/acis/bin
Webpage directories	
primary (rhodes):	/data/mta4/www/RT/acis/www
backup1 (colossus):	/data/asc1/htdocs/acis/RT
backup2 (xcanuck):	/proj/web-cxc/htdocs/acis/RT
backup3 (luke):	/proj/web-cxc/htdocs/acis/RT

*NOTE: The COLOSSUS directory is physically seperate from XCANUCK.

RTwebpage.script is run every 5 minutes via cron and contains:

acis-acorn-check.pl - The purpose of this script is to ensure that the ACORN process is running on the CPU. If one is not found, a new one is launched. ACORN is a C++ program written by Peter Allen (CXC programmer) that monitors the real-time telemetry given a UDP port and a list of mnemonics (or MSIDs) to listen for. In the Chandra telemetry, the list of the ACIS mnemonics for which data are to be written out to log files by the ACORN process are listed in a text file, acis-acorn-msid-script which is an input to running ACORN. What are output by the ACORN process are "tracelog files" that contain the values for each MSID monitored in the Chandra telemetry. Currently, the list of ACIS MSIDS stored in acis-acorn-msid-script are output to 17 tracelog files, which are then fed to acis-read.pl. The table below lists those 17 files and their expected contents. Getting back to this script, it also reads in a file containing a list of all ACORN pids, acisdude-acis.pid, and gets the pid for the last known ACORN process. It then searches the CPU for this pid. If it does not find it, it launches a new one, appends the new PID to the PID logfile, and sends email to 'acisdude' informing the CXC ACIS Ops group that the ACORN process could not be found and a new one is relaunched. This script is run via CRON every 5 minutes by 'acisdude'. Under nominal operations, there are three versions running. The "primary" version that is run out of /export/acis-flight/primary/acis/bin/ on rhodes (via UDP port #1135), and the back-up versions run out of /home/acisdude/real-time/back-up/acis/bin (via UDP port #11112 on xcanuck) and /home/acisdude/real-time/back-up/acis/bin (via UDP port #11350 on colossus). When needed the CDP machine luke, which is not usually in use for telemetry monitoring, is run out of /export/acisops/real-time/back-up/acis/bin (via UDP port #5979). The maximum size of the tracelogs is a user-specified varible (via the "-e" switch). Once a tracelog exceeds this size, a new tracelog file is automatically started. acis-tl-cleanup.pl is a separate perl script that deletes obsolete tracelog files and is run independently from acis-acorn-check.pl via CRON.

ACIS Tracelog Files (note: the '*' indicates a timestamp in the filename)

acisda_*.tl	ACIS DA temps & voltages
acisdea_*.tl	ACIS DEA temps & voltages
acisdpa_*.tl	ACIS DPA temps & voltages
acisEPHIN_*.tl	EPHIN channel readings
acisFORMAT_*.tl	Spacecraft format, OBSID, Dither status
acisHWLEDs_*.tl	ACIS HW LEDs (1STAT7ST, 1STAT6ST, 1STAT5ST, 1STAT4ST)
acisISIM_*.tl	ISIM temperatures
acisLED1_*.tl	DPA power supply status / DA (B) heater status
acisLED2_*.tl	DEA power supply status / DA (A) heater & bakeout status
acismech_*.tl	ACIS Mech (door & vent valve status)
acisother_*.tl	ACIS Big Valve & support structure +/-Y temps
acisRADMON_*.tl	RADMON status
acisSIMFOCUS_*.tl	SIM and Focus positions
acisSIMMOTOR_*.tl	SIM Motor overcurrent & stall counters
acisSWLEDs_*.tl	ACIS SW LEDs (1STAT3ST, 1STAT2ST, 1STAT1ST, 1STAT0ST)
acistempa_*.tl	ACIS thermal control side A
acistempb_*.tl	ACIS thermal control side B

acis-read.pl - This perl script is to be executed on the ACIS tracelog files produced by ACORN (monitored via acis-acorn-check.pl). What this script does is to find the most current ACIS tracelogs in the current working directory (there should 17 tracelog files in use at any given time) and calculate the mean value and standard deviation for the last 10 samples for each mnemonic in every tracelog file. It will also determine the time span that corresponds to the range of samples in the mean value calculation. The calculated values and time stamps are then written to 14 log files to be displayed on an ACIS scoreboard web-page which is produced by acis-www.pl. This script is run via CRON every 5 minutes by 'acisdude'. Limitpager-primary.pl reads in these log files and compares each mean value against a set of red and yellow limits. If an excursion is found email and/or pager alerts are spawned. Under nominal operations, there are three versions running. The "primary" version that is run out of /export/acis-flight/primary/acis/bin/ on rhodes (via UDP port #1135), and the back-up versions that are run out of /home/acisdude/real-time/back-up/acis/bin (via UDP port #11112 on xcanuck - visible from anywhere on the HEAD LAN) and in /home/acisdude/real-time/back-up/acis/bin (via UDP port #11350 on colossus). When needed, the luke version is run from /export/acisops/real-time/back-up/acis/bin (via UDP port #5979).

acis-www.pl - This perl script takes the output generated by acis-read.pl, namely the 14 log files containing the mean value, standard deviation, and start/stop times for all the MSIDs monitored in the real-time telemetry and generates an output html file (acis-mean.html) to display these values. The html file is written such that the browser is instructed to reload the page every 2.5 minutes. This script itself is executed every 5 minutes via CRON by user 'acisdude' so that when in real-time contact with the observatory, the page quickly updates the values for the MSIDs monitored. The script also reads in the 5 history files originally produced by acis-backstop.pl, and concatenated to the "global" history files by ACE-update.ksh, so that ACIS operations team personnel may compare the real-time status of the observatory (in terms of obsid, focal plane instrument, gratings status, telemetry format, and dither status) against what is expected from the approved mission load. Moreover, the script also reads in the output ACIS fluence file, produced by acis-fluence_P3pscaling.pl and acis-fluence_FP6pscaling.pl, so that the current ACIS orbital fluence may also be read from this file. Since the EPHIN data are also monitored and displayed, this allows the ACIS operations team members to monitor the radiation environment at Chandra at all times anywhere in the world as long as they have access to a web browser! Under nominal operations, there are three versions running. The "primary" version that is run out of /export/acis-flight/primary/acis/bin/ on rhodes (via UDP port #1135), and the back-up versions that are run out of /home/acisdude/real-time/back-up/acis/bin (via UDP port #11112 on xcanuck and UDP port #11350 on colossus) and once again, when needed the luke version is run from /export/acisops/real-time/back-up/acis/bin through UDP port #5979. Lastly, limitpager-primary.pl reads in these log files and compares each mean value against a set of red and yellow limits. If an excursion is found email and/or pager alerts are spawned. Yellow limit violations are colored "yellow" and red limit violations are colored "red" on the html display page.

acis-www.plimitpager_[primary/backup].pl - This script evaluates the real-time data when in COMM with the spacecraft and generates alert messages if any values are found to be outside of their limit bounds for more than 3 consecutive readings. The limits used in this script have been decided upon by the SOT and ACIS teams in consultation with the MIT ACIS team. Those limits can be found at: http://asc.harvard.edu/mta/RT/acis/www/limits_web.htm. The script will generate email messages if a yellow limit has been exceeded and will generate email and pager messages if a red limit has been exceeded.

acis-www.pacis-tl-cleanup.pl - This perl script removes old ACIS tracelogs produced by ACORN (see acis-acorn-check.pl for more details). What this script does is to find all ACIS tracelogs (*.tl) in its working environment (/export/acis-flight/primary/acis/bin/ for the primary version on rhodes and /home/acisdude/real-time/back-up/acis/bin/ for the back-up versions on xcanuck and colossus, and /export/acisops/real-time/back-up/acis/bin on luke) and deletes all but the most recent one. Therefore, old and obsolete tracelog files are deleted so as to not fill up disk space.

3.2 ACIS Process Monitor (PMON)

The directories where the PMON software is maintained are indicated in the table below. Listed below the table are the PMON scripts that are run via cron on all three machines listed above, along with a brief description of the script.

Binary directories	
primary (rhodes):	MIT PMON is the primary PMON page
backup1 (colossus):	/home/acisdude/pmon2 (OPS LAN)
backup2 (xcanuck):	/home/acisdude/pmon (HEAD LAN)
backup3 (luke):	/export/acisops/pmon (HEAD LAN)
Webpage directories	
primary (rhodes):	MIT PMON is the primary PMON page
backup1 (colossus):	/data/acisdude/PMONweb & /data/asc1/htdocs/acis/RT
backup2 (xcanuck):	/data/acis/PMON & /proj/web-cxc/htdocs/acis/RT
backup3 (luke):	/export/acisops/PMON & /proj/web-cxc/htdocs/acis/RT

pmon-pid-check.pl - The purpose of this script is to ensure that the catnrt process is active on the machines "colossus" (OPS LAN) and "xcanuck" (HEAD LAN) and "luke" (HEAD LAN) - when it is operating. The catnrt process is the script that feeds the real-time data to the PMON script that produces the PMON web page. pmon-pid-check.pl is run via CRON every 5 minutes. At run-time, the script reads a file that contains a list of all catnrt processes (stored at /var/tmp/acisdude-colossus-pmon.pid on colossus, /var/tmp/acisdude-xcanuck-pmon.pid on xcanuck and /var/tmp/acisdude-luke-pmon.pid on luke) started on colossus/xcanuck/luke and grabs the last entry. It then queries the CPU (via '/usr/bin/ps -f -u acisdude') to isolate all the processes being run by 'acisdude'. From that list, it tries to find the catnrt process given the

PID from the list of all catnrt processes. If it cannot find it, an email is spawned to ‘acisdude’ indicating it could not find the catnrt process. At the same time, it launches a new catnrt process and appends the date and PID of the new process to the aforementioned process id (pid) log files.

catnrt is a shell script originally written by Peter Ford and modified by Shanil Virani to enable PMON to be run on the CXC side. It is a tool that makes use of Peter Ford’s ACIS TOOLS directory (/home/pgf on colossus, /home/acisdude/pgf on xcanuck, and /export/acisops/pgf on luke) to look for ACIS real-time telemetry either via the raw telemetry repository located at /dsops/GOT/baseline/RT_raw or through the UDP ports #11151 on colossus, #11113 on xcanuck, and #5978 on luke. It then feeds the ACIS data to the PMON executable (located at /home/acisdude/pmon2/pmon on colossus, /home/acisdude/pmon/pmon on xcanuck, and /export/acisops/pmon_luke on luke) to produce the PMON web page. pmon-pid-check.pl is a script run via CRON (at 5-minute intervals) to ensure there is always a ’catnrt’ process active on the CPU. If one cannot be found, a new one is automatically launched and email is sent to ’acisdude’.

PMON displays an ACIS serial telemetry stream. It reads an ACIS packet stream, e.g., as created by *getp* or *getnrt* and writes a formatted display to *stdout* via the *ncurses* interface. The stream is read from file, or if omitted, from the standard input stream, *stdin*. If *file* ends in ’.gz’ or ’.Z’, it will be decompressed by *gzcat* as it is read. When a suitable terminal is connected to stdout, PMON will use colored, boldfaced, and inverted text to highlight specific fields, and an audible alarm to warn of error conditions. An error, e.g., excessive bias-parity errors, t-plane latchup, etc., can be made to trigger an external script which may, for instance, send e-mail to a list of users. PMON can write an HTML file that displays ncurses window on a Web browser. A number of changes can be made while PMON is running by hitting various keyboard keys.

ACIS Realtime Software Directory Structure

function:	**Primary**	**Backup1**	**Backup2**	**Backup3**
	HEAD/OPS — LAN	OPS — LAN	HEAD — LAN	HEAD — LAN
machine:	**rhodes**	**colossus**	**xcanuck**	**luke+**
FLU−MON:	/export/acis−flight/FLU−MON	/home/acisdude/FLU−MON	/proj/sot/acis/FLU−MON	/export/acisops/FLU−MON
Eng BIN:	/export/acis−flight/primary/acis/bin	/home/acisdude/real−time/back−up/acis/bin	/home/acisdude/real−time/back−up/acis/bin	/export/acisops/real−time/back−up/acis/bin
Eng WEB:	/data/mta4/www/RT/acis/www	/data/asc1/htdocs/acis/RT	/proj/web−cxc/htdocs/acis/RT	/proj/web−cxc/htdocs/acis/RT
PMON BIN:	**N/A***	/home/acisdude/pmon2	/home/acisdude/pmon	/export/acisops/pmon
PMON WEB:	**N/A***	/home/acisdude/PMONweb & /data/asc1/htdocs/acis/RT	/data/acis/PMON & /proj/web−cxc/htdocs/acis/RT	/export/acis/PMON & /proj/web−cxc/htdocs/acis/RT
TLM FEED: UPD (machine) − process	11350 (gamera) − ACORN	11151 (gamera) − catnrt 11350 (forbin) − ACORN	11113 (crosby) − catnrt 11112 (forbin) − ACORN	5978 (cdpmon) − catnrt 5979 (cdpmon) − ACORN

*primary PMON page is run by MIT

Realtime Webpages:

rhodes engPage:	http://asc.harvard.edu/mta/RT/acis/www/acis−mean.html
rhodes PMON:	http://acis.mit.edu/asc/acisrt.html
colossus engPage:	http://asc1.cfa.harvard.edu/acis/RT/acis−mean.html
colossus PMON:	http://asc1.cfa.harvard.edu/acis/RT/acisrt−colossus.html
xcanuck engPage:	http://asc.harvard.edu/acis/RT/acis−mean.html
xcanuck PMON:	http://asc.harvard.edu/acis/RT/acisrt−xcanuck.html
luke engPage:	http://asc.harvard.edu/acis/RT/acis−mean−luke.html
luke PMON:	http://asc.harvard.edu/acis/RT/acisrt−luke.html

mount points:	/export − rhodes /home/acisdude − remus /data/mta4 − rom−48	/home/acisdude − colossus /data/asc1 − asc1	/proj/sot − rom−48 /proj/web−cxc − numa /home/acisdude − remus	/export/acisops − luke /home/acisdude − numa‡

+Note that the luke real−time processes are only used for testing or emergencies

‡ when remus is unavailable, syshelp will move the /home/acisdude mount point to numa

last updated: 08/04/09