NATIONAL
OPTICAL
ASTRONOMY
OBSERVATORY

SYSTEM INSTRUMENTATION GROUP
950 N. Cherry Ave.
P. O. Box 26732
Tucson, Arizona 85726-6732
(520) 318-8000 FAX: (520) 318-8303

`

TORRENT

Program Design & Description

collector

NOAO Document TRNT-AD-08-0004

Revision: 1.1

Authored by:
Nick C. Buchholz

Please send comments:
nbuchholz@noao.edu

Revision History

Version	Date Approved	Sections Affected	Remarks
0.1	7/9/2009	All	Initial draft - ncb
1.0	20091211	All	Added Appendices, Introduction
1.1	20120726	All	Corrections of typos etc. Removal of Beowulf cluster requirement for archive – pnd.

Table of Contents

List of Figures

1.0 Introduction

This document describes the design, usage and internals of the Torrent ***collector*** program. This program pulls together the ".cfg" files, the data contained in the DHE EEPROMs and transition module EEPROM and produces the ".csv", ".ini" and ".mod" files used by the PAN processes to control and configure the DHE and detector.

2.0 Torrent Auto-configuration

The start of the auto configuration process is the ***assimilate*** program. This program reads the FPGA project build file for a firmware version and extracts the names and locations of all files used to build the FPGA firmware. The program then reads all of these files and extracts comment lines of a particular format. The program then interprets/parses the comments and creates a set of ".cfg" files, one for each module in the FPGA code as described in the comment lines. The program then reads two ".vhd" files containing vhdl comment lines but no vhdl code. These files, Torrent_SFTW.vhd and Torrent_TSM.vhd, are parsed as well. The result of this activity is a set of ".cfg" files containing a description of the properties and location if the DHE EEPROMS of every attribute defined in the FPGA code that is available for reading and or writing by the PAN processes.

These ".cfg" files are then used to create a template "Tmplt.csv" file. The files are unique to each major revision/version of the firmware running in a Torrent system, and describe the memory map of the DHE attributes and EEPROMs.

At runtime the ***collector*** program reads these ".cfg" and uses them to obtain information from the EEPROMs about the attributes to be described. This information is then combined with the "Tmplt.csv" to create a sysName_config.csv that is read by the pan processes to configure and control the detector.

The user GUI, mborg, borg, nocs or mop, executes a script that has the same name as the focal plane or instrument to be run. Thus to start the Kosmos E2V instrument the GUI would run the ***"kosmos_e2v"*** script. This script clears the SysTran fiber link and starts the ***collector*** program. On successful completion the system script calls the runTRNT or runPANPcie scripts to start the PAN processes.

The Torrent_TSM.cfg file is modified during the system configuration and detector optimization process. This file is used by ***collector*** to read the default, maximum and minimum values for every hardware function connected to the Dewar from the TSM module's EEPROM and create a defaultSetup.mod and a setVoltages.mod file to be used by the initialization routines at system startup to set all attributes to optimum levels. See Section 3.0 and the sysConfig User Manual for a detailed description of this process

2.1. Connecting Registers to the Detector Functions (TSM .cfg File and EEPROMs)

The TSM EEPROM contains the default, maximum and minimum values allowed for each of the detector voltages, clocks and other attributes. Each detector voltage/clock and so forth is described in the TSM configuration file and is associated with a hardware register described in one of the FPGA module ".cfg" files described above. The integrating engineer will use the ***sysConfig*** tool to create a Torrent_TSMV###.cfg file that connects a hardware register function to a detector requirement.

The Torrent_TSMV###.cfg file will be used and modified by sysConfig. It will have the format described in Appendix II.

2.2. Torrent Configuration and calibration

When a system or detector engineer is assigned to create a new focal plane to be used in an instrument they must first describe the system configuration using the ***sysConfig*** suite of description tools. See the sysConfig manual (TRNT-AD-08-0006R1.3-SYSCFG.doc) for a description of these tools.

Each Torrent board can be calibrated using a python tool to do the calibration. The result of this calibration is written into the board EEPROM for later use by the ***collector*** program. In practice only the AFE boards need to be calibrated in this way as the default values derived for the other board attributes from the FPGA code comments are sufficiently accurate for use in real detector systems

2.3. Collector usage

The collector program as run by the startup scripts has a GUI display that allows the user to follow the steps performed. In fact there is usually no reason for the user to interact with the collector at all under normal conditions. However, several unusual or error conditions can be encountered by collector that may require the user to take some action. The following paragraphs outline these conditions and the user's response and the consequences of those responses.

2.3.1.Pan processes Running.

Message: "**The Collector has detected the pan processes are already running.**
These need to be shut down before Collector can run."

User Response: Ensure that no other user in using the system and shut down the PAN processes if the system is Idle.

Result: The system will be shutdown while the user kills the PAN processes and the user will have to re-issue the command to start the system

2.3.2.*collector* fails to connect to PAN.

Message: "**Error: The Collector has failed to connect to the PAN**
Cause: *cause of the failure is listed here*"

or: "**Error: connect failed: DHE Connected to *panName:*** *some Linux system error message.*

User Response: If possible the user should correct the problem listed

Result: The system will be shutdown while the user corrects the problem or calls electronic maintenance to correct the problem. The user will have to re-issue the command to start the system

2.3.3.Communication to PAN failed

Message: **"Communication to PAN failed:**
Cause: *cause of the failure is listed here*"

User Response: If possible the user should correct the problem listed or determine why the problem occurred.

Result: The system will be shutdown while the user corrects the problem or calls electronic maintenance to correct the problem. The user will have to re-issue the command to start the system

2.3.4.Pan comes up in simulation mode

Message: **"The pan software has started in simulation mode.**

User Response: The user will be asked if they want to continue in simulation mode.

Result: If the user answers "YES", the system will continue to startup with the software in simulation mode. No valid data can be taken in this mode. If the answer is "NO", the system will shutdown until the user determines why the software came up in simulation mode or calls electronic maintenance to correct the problem. The user will have to re-issue the command to start the system

2.3.5.Not a Torrent system

Message: "**Error: the DHE Connected to** *panName* **is not a Torrent system.**
Collector will shut down and try to start the Orange software"

User Response: The user will be asked if they want to attempt to start the MONSOON Orange version of the software with this system.

Result: If the users answers “YES” ,the software will continue to start the system as a MONSOON Orange system. If the answer is “NO”, the system will shutdown until the user determines why collector detected a non-Torrent DHE. The user will have to re-issue the command to start the system

2.3.6.Cfg database setup failed.

Message: **"Collector failed during DB Setup:**
Cause: *cause for setup failure listed here.*
Should we try to start using existing Files?'

User Response: The user will be asked if they want to attempt to start the system using the previous version of the startup and setup files.

Result: If the users answers “YES”, the software will ask the user for the continuation password and if correct, continue to start the system using the existing “.csv”, “.ini”, “.arr” and “.mod” files. This is generally safe if no hardware changes have been made since the last time the system was started. If the answer is “NO”, the system will shutdown until the user determines why the database setup failed and corrects the problem or calls electronic maintenance to correct the problem. The user will have to re-issue the command to start the system

2.3.7.collector fails reading the EEPROMs or EEPROM Mirror files.

Message: **“Error (C431): Collector failed reading EEPROMs Mirror files:**
Cause: *cause of the failure is listed here*"
or: **"Error (C437): Collector failed reading EEPROMs:**
Cause: *cause of the failure listed here*
or: **"Error (C453): Collector failed in rdEeproms:**
Cause: *cause of the failure listed here*

User Response: The user will be asked if they want to attempt to start the system using the previous version of the startup and setup files.

Result: If the users answers “YES”, the software will ask the user for the continuation password and if correct, continue to start the system using the existing “.csv”, “.ini”, “arr” and “.mod” files. This is generally safe if no hardware changes have been made since the last time the system was started. If the answer is “NO”, the system will shutdown until the user determines why the database setup failed and corrects the problem or calls electronic maintenance to correct the problem

2.3.8.sysName_Config.csv file write failed.

Message: **“Error (C467): Collector failed doing sysName_Config.csvm file write:**
Cause: *cause of the failure is listed here*"
or: **"Error (C473): Collector failed doing sysName_Config.csvm file write**
Cause: *cause of the failure listed here*

User Response: The user will be shown the error message and will asked to confirm an orderly shutdown of the system

Result: The system will shutdown until the user determines why the file write failed and corrects the problem or calls electronic maintenance to correct the problem

2.3.9.sysName.ini file write failed.

Message: **“"Error (C505): Collector failed doing sysName.ini file write:**
Cause: *cause of the failure is listed here*"
or: **"Error (C511): Collector failed doing sysName.inim file write**
Cause: *cause of the failure listed here*

User Response: The user will be shown the error message and will asked to confirm an orderly shutdown of the system

Result: The system will shutdown until the user determines why the file write failed and corrects the problem or calls electronic maintenance to correct the problem

2.3.10. sysName_DefaultSetup.csv file write failed

Message: **“Error (C524): Collector failed doing sysName_DefaultSetup.mod file write:**
Cause: *cause of the failure is listed here*"
or: **"Error (C530): Collector failed doing sysName_DefaultSetup.mod file write**
Cause: *cause of the failure listed here*

User Response: The user will be shown the error message and will asked to confirm an orderly shutdown of the system

Result: The system will shutdown until the user determines why the file write failed and corrects the problem or calls electronic maintenance to correct the problem

2.3.11. sysName_SetVoltages.mod file write failed

Message: **“Error (C542): Collector failed doing sysName_SetVoltages.mod file write:**
Cause: *cause of the failure is listed here*"
or: **"Error (C548): Collector failed doing sysName_SetVoltages.mod file write**
Cause: *cause of the failure listed here*

User Response: The user will be shown the error message and will asked to confirm an orderly shutdown of the system

Result: The system will shutdown until the user determines why the file write failed and corrects the problem or calls electronic maintenance to correct the problem

2.3.12. collector failed renaming setup files

Message: **“Error (C568): Collector failed moving existing setup files to .old:**
Cause: *cause of the failure is listed here*"
or: **" Error (C579): Collector failed moving .m files to final name:"**
Cause: *cause of the failure listed here*

User Response: The user will be shown the error message and will asked to confirm an orderly shutdown of the system

Result: The system will shutdown until the user determines why the file write failed and corrects the problem or calls electronic maintenance to correct the problem

2.4. FPGA code Version Numbers

Each FPGA firmware version has a three digit version number, i.e. 2.10 or 2.22. These numbers are stored as an attribute in the FPGA and in each DHE EEPROM. The first digit of the version number describes the overall version of the system. The second number designates the memory map being used. The third digit is the version of a memory map. The minor (third) version number denotes a version that corrects errors or changes functionality in a way that does not effect the overall operation of a system.

When the collector starts it verifies that these version numbers match in the first two digits. Versions 2.10, 2.11 and 2.16 are all considered equivalent for the purposes of getting startup information from the EEPROMs and building the startup and configuration files

If a mismatch occurs; FPGA code version 2.22 with EEPROM 2.12 or EEPROM version 2.22 with EEPROM mirror files version 2.12, the collector will announce an error and halt until the problem is resolved.

The engineer writing the FPGA code and the software engineer have agreed that all versions 2.1x will use the same memory map and the only thing that will change from version 2.10 to 2.11 or 2.13 will be the internal functionality of the FPGA. This means that *assimilate* only needs to be run when changing from 2.1x to 2.2x or

3.1x

Appendix I Control of Torrent Configuration Data, files and EEPROM storage

A key feature of the Torrent detector controller system is the ability to automatically determine the hardware configuration, required voltage settings and other information needed to safely and effectively run the attached detector to produce astronomical data. One goal of the Torrent system is to allow rapid response in the event of a failure to reduce lost observing time to a minimum. Four identifiable pieces of hardware can fail and allow mountain maintenance staff to swap hardware in the event of a failure will reduce the duration of any down time.

The four pieces of hardware are the PAN computer, the power supply brick, the DHE controller box, and the dewar/transition module. The first three will be swappable with backup hardware in a matter of minutes.

Swapping the power supply brick can be done in less than a minute unplugging two plugs and plugging in the new brick.

The DHE controller box should be similarly swappable requiring disconnecting communications cables and a few screws removing the old DHE and re-installing the replacement. Since the Torrent DHE's were designed without hardwired configuration jumpers of any kind. Two DHE of the same type (IR or CCD) with the same number of AFE (Analog Front End) boards can be swapped with the assurance that the new DHE will run the detector in the same manor as the old one.

Swapping the PAN may take longer if the spare is powered down or does not have an installed Systran communications card it could take up to an hour to install the card and boot up the spare machine. Additional time may be needed to reconfigure the network description and IP addresses. If a spare PAN with the Systran communications board installed is available the swap can be reduced to swapping the communications fibers assuming the spare is kept online with its own name and IP address.

A failure in the dewar/transition module is more problematic as it will probably require warming the dewar and debugging procedures while protecting the detectors from damage. This could require hours or even days.

The features of the Torrent system that make this rapid turn-around possible are partially a set of tools to insure the information required by the system is consistent, up-to-date and correct, and partially a set of procedures to insure backup and verification of that information.

Appendix II Required Torrent Information

The Torrent systems, tools and procedures are designed to keep track of several kinds of information:

- Hardware module configuration files – these files are produced by the ***assimilate*** tool that reads the VHDL source code files used to build the FPGA code that implements the various functions available in the DHE. One "**.cfg**" file is produced for each of the firmware modules in the system. The structure of these files is explained in NOAO document TRNT-AD-08-0003 – "TORRENT Program Design & Description – ***assimilate***". One set of "**.cfg**" files is produced for each version of the FPGA code.
- Firmware module EEPROM data and EEPROM mirror files – Each firmware module stores calibration and conversion data in one of the module EEPROMS in the Torrent DHE. The PSM (power supply module), LCB (local control board) and AFE1 and AFE2 (analog front end boards) each have an EEPROM consisting of 128 pages of 16 - 32 bit words. All of the user accessible DHE functions in all modules have an entry in one of these EEPROMS. The data in the EEPROMS match the layouts described by the ".cfg" files created by assimilate**.** This layout is constant for any major revision level of the FPGA code. As newer firmware versions are produced the ***assimilate*** program will create a new layout as needed. The version of the layout is stored in the first two pages of the EEPROM so the Torrent tools can determine the structure of the data in the EEPROM
- Transition Module configuration files – The configuration file that describes the layout of the TSM or Transition Module EEPROM is created by the detector engineer doing the integration of the Torrent system with the detector/dewar using the ***sysConfig*** tool. This tool allows the engineer to assign detector/dewar functions to DHE functions. The tool keeps track of connections and allows the assignment of meaningful names to the DHE functions. The tool also allows the engineer to specify the allowed operation range, normal operation levels, and alarm and warning levels for each function. The results of this effort are: a) a configuration file including a wire list that can be used to connect the Dewar connectors and DHE transition or utility boards, b) a "**.cfg**" file describing the EEPROM layout and, c) an EEPROM mirror file containing the data that should be stored in the TSM EEPROM.
- operational "**.csv**", "**.arr**", "**.ini**" and "**.mod**" files – at startup the PAN processes use a per system "**.csv**" file to determine what attribute-value pairs are accessible and used by the current system and their conversion factors and calibration.. With earlier MONSOON systems this "**.csv**" file was a static file created by the detector integration engineer and shipped with the PAN computer. Changes to these files were fraught with complication and editing was difficult and exacting. The Torrent systems use the same PAN process software as the earlier MONSOON systems, however the "**.csv**" files used by the Torrent systems will be automatically generated by the ***collector*** program from the information stored in the firmware module "**.cfg**" files, EEPROMS and the TSM module "**.cfg**" and EEPROM. This "**.csv**" file should not be hand edited.

Appendix III Torrent Start-up Procedure

The Torrent systems will use the same PAN process software as the other MONSOON systems. Since the Torrent software tools are able to determine the capabilities of the hardware using the information stored in the ".cfg" files and EEPROMS the startup procedure is slightly different from the procedure used by a MONSOON system.

The system startup procedure can be broken into two parts; the Torrent startup, and the MONSOON startup. The Torrent startup scripts will perform the tasks needed to start a torrent system using the MONSOON software. These tasks will be performed by the ***collector*** program:

1. ***collector*** will read the module "**.cfg**" files for the current firmware version available on the PAN. It will then attempt to connect to the MONSOON server and verify that these "**.cfg**" files are up to date and consistent with the archived files.
2. ***collector*** will read the TSM module "**.cfg**" file available locally and attempt to check it against the archived file. If the archived file is unavailable or the files do not have the same date, ***collector*** will announce an error and request direction as to how to proceed.
3. The ***collector*** program now reads the EEPROMS and confirms the data integrity of the data using the checksum and internal ID fields.
4. Using the information in the "**.cfg**" files and EEPROMS ***collector*** now creates a MONSOON "*sysname*_Config.csv" file and a "*sysname*_DefaultSetup.mod" file in the "/Monsoon/cfg/_*sysname*" directory on the PAN and starts the pan processes using the standard MONSOON startup scripts.

If the "**.cfg**" files are unavailable and/or the EEPROMs fail the validity tests and there exists a valid "**.csv**" file in the "/Monsoon/cfg/_*sysname*" directory the collector program will request permission to proceed with the startup using the existing files.

Appendix IV Data Integrity, Consistency and Validity Procedure

The validity and consistency of the four data inputs to the collector program will be maintained automatically by the programs that produce the data.

- ***assimilate*** will create the "**.cfg**" files from the VHDL files for the current firmware version. These files will be stored on each PAN in the "/Monsoon/cfg/_common" directory and in the Torrent archive location. The files will contain a digital signature created by ***assimilate*** and will contain verification information in the file header. ***assimilate*** will also create a template ".csv" file in the "/Monsoon/cfg/_common" directory using the hardware design max and min values and binary to binary slope and intercept values.
- ***sysConfig*** reads the "**.cfg**" files created by ***assimilate*** and will create the "**.cfg**" file for the transition module in accordance with the desires of the detector integration engineer. A copy of the "**.cfg**" file and a mirror of the TSM module EEPROM will be stored by **sysConfig** in the "/Monsoon/cfg/_*sysname*" directory on the PAN and in the Torrent archive location. These files will contain a digital signature created by ***sysConfig*** and will contain verification information in the file header.
- ***collector*** will read the "**.cfg**" files created by ***assimilate*** and ***sysConfig*** from the local PAN and will verify the validity of those files using the digital signatures and verification information in the file header. If possible ***collector*** will contact the Torrent archive location to verify that the archive contains the same data as the local PAN. ***collector*** will then read the data from the EEPROMS and create a digitally signed copy of the generated "*sysname*_Config.csv" and a "*sysname*_DefaultSetup.mod" files.
- ***eepStor*** should only be used in extreme circumstances. ***eepStor*** will read the "**.cfg**" files created by ***assimilate*** and ***sysConfig*** from the local PAN and will verify the validity of those files using the digital signatures and verification information in the file header. If possible ***eepStor*** will contact the Torrent archive location to verify that the archive contains the same data as the local PAN. ***eepStor*** will, on command write changes made to the EEPROM images by the engineer to the EEPROMs, to the "/Monsoon/cfg/_*sysname*" directory as EEPROM mirror files and to the Torrent Archive location. The files created by ***eepStor*** will be digitally signed and will contain verification information in the file header.
- the torrent test procedure will read the "**.cfg**" files created by ***assimilate*** and conduct the board test using the information contained in those files. When the test is complete and the board checks as "OK". The procedure will write the test result report and create an EEPROM mirror file in the Torrent archive area.
- The torrent AFE calibration procedure will calculate the calibration constants for all of the attributes it handles and write the calibration results to the board EEPROM in the correct format. The test report and mirror file will be digitally signed and will contain verification information in the file header.

sysConfig, collector and eepStor will be restricted to run only on a PAN with the desired DHE/dewar system connected and powered up. A system ID will be stored in the TSM EEPROM as part of the ID data on the first two pages. These programs will be configured to allow them to run with a system containing blank or corrupt EEPROM data as a special case.

Appendix V Torrent Archive Location

A 2 GB partition on big-boy will be needed as the Torrent archive with a mirror of the partition on a machine in the south. Syncing of the two archives should take place on a regular basis.