



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

## Software System Description

NOAO Document TRNT-AD-08-0001  
Revision: 0

Authored by:  
Nick Buchholz

Please send comments:  
[nbuchholz@noao.edu](mailto:nbuchholz@noao.edu)

## Revision History

Version	Date Approved	Sections Affected	Remarks
0	8/5/2009	All	Initial draft release - aro

# Table of Contents

<b>Revision History .....</b>	<b>2</b>
<b>Table of Contents.....</b>	<b>3</b>
<b>1.0 Introduction.....</b>	<b>4</b>
1.1 Torrent System Hardware/Software Goals .....	4
1.2 Torrent Hardware .....	4
1.3 System Software to Meet the Torrent Goals .....	5
<b>2.0 Torrent Processes and Programs .....</b>	<b>7</b>
2.1 MONSOON PAN Processes and Programs .....	7
2.1.1 panDaemon .....	7
2.1.2 panCapture .....	7
2.1.3 panProcAlg.....	8
2.1.4 panSaver .....	8
2.1.5 MONSOON mec .....	8
2.1.6 csvEdit.....	8
2.2 Torrent-specific Process Descriptions .....	8
2.2.1 <i>assimilate</i> .....	8
2.2.2 System Configuration Tools .....	8
2.2.3 <i>mborg</i> Test and Optimization.....	9
2.2.4 <i>collector</i> .....	9
<b>3.0 Torrent Auto-configuration.....</b>	<b>9</b>
3.1 sysConfig .....	10
3.2 mborg and Detector and Focal Plane Testing/Optimization.....	10
3.3 Connecting Registers to the Detector Functions. (TSM .cfg File and EEPROM) .	10
3.4 Torrent Configuration and calibration .....	10
3.4.1 sysConfig.....	10
3.5 Torrent Runtime Routines.....	11
3.5.1 borg .....	11
3.5.2 borgMec .....	11
<b>4.0 Version Numbers .....</b>	<b>11</b>

## 1.0 Introduction

This document describes the design of the Torrent software system from the lowest level to the user interface. At each level we have tried to make certain that a Torrent system will operate in a manner compatible with safe operation of the focal plane and easy maintenance of the hardware, while ensuring minimum down time in the event of a failure.

In addition, to the extent possible, the design goals are obtained by automating as much of the configuration and operation of the systems as possible.

### 1.1 Torrent System Hardware/Software Goals

The Torrent hardware and associated software are designed to meet several goals. They are:

- The Torrent system should provide detector-limited performance for the focal plane being controlled.
- The Torrent system should use the same software as a base MONSOON system. All existing user clients, (MEC, MOP, NOCS and CTIO's ArcView for MONSOON) must work with Torrent hardware systems without modification.
- The correspondence between the Torrent hardware attributes/registers and the attribute-value pairs in the configuration files used by the base MONSOON software should be automated.
- The startup and initialization of the focal plane control should automatically choose the proper configuration and voltage levels for the focal plane being controlled.
- The focal plane calibration values, default, maximum, minimum and warning and alarm out of range values should be carried with the focal plane Dewar, not on external media.
- Tools should be provided to ease the configuration, testing and calibration of Torrent focal plane control systems and to prevent costly mistakes due to a mis-configuration.

Other goals related purely to the Torrent hardware can be found in Torrent DHE Hardware Description, TRNT-AD-11-0001.

### 1.2 Torrent Hardware

A complete description of the Torrent hardware is provided in Torrent DHE Hardware Description, TRNT-AD-11-0001 Revision 0. Briefly, the Torrent hardware consists of three components, a power supply brick, a controller box consisting of a power supply module and a DHE module, and a transition module that is permanently attached to the focal plane Dewar containing the detector(s) being controlled.

The DHE module contains a Local Control Board (LCB) board that handles communications (an FPDP compatible SYSTRAN link, a GigE data transfer link and custom synchronization links used for connecting multiple DHE's together), voltage control and control of the Analog Front End (AFE) board(s); one or two AFE boards that generate the clocks and voltages required by the focal plane, controls the conversion of the analog video to a digital signal and generates the video offset voltages to condition the incoming video signal.

The transition module provides pre-amplification of the video inputs, voltage protection circuitry, heater control and temperature read back capabilities. In addition it provides protection for the detector signals when the DHE is not connected to the system. This module also contains an EEPROM that will be programmed with the correct values for all bias and clock voltages as well as a description of the focal plane. This information will be used when the system is started to create the correct ".csv" file for the focal plane and to do the initialization of the detector voltages to the correct operating levels.

### 1.3 System Software to Meet the Torrent Goals

Meeting the Torrent System software goals starts with the design of the hardware and firmware. The AFE board circuitry is designed to deliver detector-limited performance. The software does not impose any limitation that would compromise this goal. The combined hardware/firmware/software system should then be able to achieve the first goal.

The Torrent DHE is controlled at the lowest level by the firmware code located in the FPGA on the LCB board. The LCB provides the same FPDP communications interface as the MONSOON "Orange" systems. Both the hardware link and the software protocol are completely compatible with the "Orange" systems. This means current MONSOON client software can control a Torrent system without modification as long as there is no need to access any of the unique features of the Torrent hardware. This satisfies the second goal.

The Torrent firmware is built using a series of ".vhd" files defining the code to be loaded into the FPGA. These files are annotated with comments, (for each user-accessible attribute, register or function), that describe information about that attribute. These same files are read by a Torrent auxiliary program called *assimilate* that uses the information in the ".vhd" files to create a description of the attributes, assign attribute information to locations in the EEPROM for each hardware module and from this information create a ".cfg" file description for each firmware module and a template ".csv" file that can be read by the MONSOON PAN processes to define how the DHE is controlled. This template ".csv" file uses best guess conversion factors based on our knowledge of the hardware design to do conversions from engineering values to bit patterns. The board testing procedure and software will refine these values to allow the registers to be set with engineering values like volts and automatically set the register to the correct bit pattern in the DAC to achieve the desired voltage. This satisfies the third goal.

An additional program will be run at the start of all future MONSOON/Torrent systems. This program (*collector*) will connect to the DHE through a minimal panDaemon and determine if the DHE is a MONSOON "Orange" or a Torrent system. If it is an "Orange" system the collector will hand off execution to a standard MONSOON startup script and shut down.

If the DHE is a Torrent system the **collector** will obtain the “.cfg” description files for the firmware modules and combine the “.cfg” information with the data stored in the EEPROM(s) on the DHE and create a *sysName\_Config.csv* file for the system being started. This “.csv” file will be customized with the correct calibration constants for all the voltage and telemetry attributes and with the maximum and minimum voltages that can be achieved by the hardware. The **collector** program will then read the TSM “.cfg” file and read the TUB (Transition Utility Board) EEPROM that is permanently attached to each Torrent-configured Dewar. This allows the **collector** program to obtain the Focal Plane ID , default, maximum, minimum and warning and alarm values for this Dewar and focal plane. Using this information, it will create a *sysName\_DefaultSetup.mod* and a *sysName\_SetVoltages.mod* file that will be used by the MONSOON client to initialize the detector and an alarm database file that the client software can use to monitor the health of the system and warn the user or shutdown the detector as required. The details of this procedure can be found in Section 3.0. This will satisfy both the fourth and fifth goals. Note that the use of the alarm database will require changes to Torrent client systems.

Finally, a number of programs are provided to safely configure and characterize a Torrent system:

**arrayDesc** – will allow a detector or instrument engineer or scientist to describe a detector in a consistent way that is recognized by the other configuration tools.

**fclPlnDesc** – will allow a detector or instrument engineer or scientist to describe a focal plane in a consistent way that is recognized by the other configuration tools.

**dewarDesc** – will allow a detector or instrument engineer or scientist to describe a Dewar in a consistent way that is recognized by the other configuration tools.

**dheDesc** - will allow a detector or instrument engineer or scientist to describe a DHE in a consistent way that is recognized by the other configuration tools.

**sysConfig**, - will allow a detector or instrument engineer or scientist to use the results from the previous configuration/description tools to configure a Torrent focal plane system to match the requirements of the detector(s) being used. In addition, this program will provide documentation of the decisions made for the system record.

**mborg** – will provide for the testing of Torrent boards and systems and for the optimization of focal plane once a system has been assembled.

**eepStor**, - allows a detector engineer or instrument scientist to edit the values stored in the EEPROM(s) if required. This program will normally be used as part of the configuration, characterization and optimization cycle for a focal plane.

**csvEdit**, - added for MONSOON “Orange” users, this program allows the user to safely edit a “.csv” file by keeping track of the correct format and allowed values for fields

The Torrent version of the Monsoon PAN software is a slight expansion of the standard MONSOON software capabilities. Two new routines are provided to read the EEPROMS in the system, one EEPROM page (16 – 32 bit words) at a time. Any current MONSOON user client, MEC, NOCS or the MOP, should be able to run a Torrent system or a MONSOON “Orange” system with this enhanced version.

The Torrent system's ability to read configuration and calibration data from the transition board EEPROM protects the focal plane by not allowing dangerous voltages to be applied to the detector. In addition, in the unlikely event that a Torrent DHE fails, a replacement controller box can be connected with the assurance that the new box will run the detector with voltages that are compatible with the focal plane since they are determined by the values stored in the TUB EEPROM. CRC and other error checking is done to ensure the data stored in the TUB EEPROM is valid. In the event that the EEPROM becomes corrupt, the system will ask the user how to proceed and if a valid set of .csv and .mod files exist, will allow the detector to be powered up with the existing files. The system will also inform the responsible engineer of the failure and request the failure be repaired.

## 2.0 Torrent Processes and Programs

### 2.1 MONSOON PAN Processes and Programs

The base MONSOON processes will be used to control Torrent systems. Minor modifications have been made to accommodate the advanced configuration features of the Torrent hardware.

#### 2.1.1 panDaemon

Two new commands have been added to the ppx/gpx command set for the Torrent systems. Both enable the Torrent *collector* program to read or write EEPROM pages as blocks of data instead of having to read individual EEPROM values.

*ppxRdPrmPg* – this routine reads values from an EEPROM page. It takes two arguments, the first being the module address of the hardware that contains the EEPROM. The second is the page to be read. The routine returns a success or failure indication and a string containing the page number and the sixteen values stored on that page if successful. A failure indication is accompanied by a reason for the failure.

*ppxWrtPrmPg* – this routine writes values into an EEPROM page. It takes eighteen arguments. The first is the module address of the hardware that contains the EEPROM. The second is the page to be written. Arguments 3 through 18 are the 16-32 bit values to be written in the page, The routine returns a success or failure indication.

The current torrent design contains an 8kB EEPROM organized as 128 pages of 16–32 bit words per page. A similar pin-compatible EEPROM with 256 pages could be substituted if required in the future.

#### 2.1.2 panCapture

The panCapture routine is unchanged from the current MONSOON software. The lower level detector library routines, detCapture, detExpParams and detDescramble, can be written exactly as they were done for the “Orange” systems or they could be written to take advantage of the Torrent's planned on-board descrambling and GigE data transfer link. This will be left to the focal plane development team.

### 2.1.3 panProcAlg

The panProcAlg routine is unchanged from the current MONSOON software.

### 2.1.4 panSaver

The panSaver routine is unchanged from the current MONSOON software. In addition to the localFITS, netw and imgSrvr DHS libraries, a labDHS library and server program to allow transferring data from the PAN to a data reduction/ observer machine automatically has been added.

### 2.1.5 MONSOON mec

The MONSOON mec, while still usable as an engineering interface, is being replaced by the “*borg*” and “*mborg*” programs. These programs, written in Python and using the “Tkinter” GUI routines, will be easier to maintain and will allow easy expansion to control focal planes at both the engineering and observer level. See Section 3.5, Torrent Runtime Routines.

### 2.1.6 csvEdit

The csvEdit program has been developed mostly for the benefit of MONSOON “Orange” users. This program allows the user to edit a “.csv” config file without the dangers inherent to using a text editor to make changes. The program keeps track of the attribute lines and ensures that the proper format is maintained.

## 2.2 Torrent-specific Process Descriptions

Torrent systems use a number of new tools to support the design and implementation of detector image acquisition systems. These tools, to the extent possible, automate the process of designing, configuring and optimizing a new Torrent system.

### 2.2.1 *assimilate*

*assimilate* is a program that will be run on the firmware source code to extract the address and configuration information for the various registers and functions. The program will create a set of .cfg files that will be used to describe the hardware to the configuration tool and *collector* software. These files will be common to all systems with the same firmware version.

The software will also read a torrent\_SFTW.vhd file to create a torrent\_SFTW.cfg containing the appropriate information to be added to the .csv file to handle the software attributes.

A complete explanation of how to use *assimilate* is included in the NOAO document TRNT-AD-08-0003 Torrent Software User Manual *assimilate*.

### 2.2.2 System Configuration Tools

The sysConfig suite of programs is designed to be an engineer’s assistant to be used when an engineer is designing and configuring a new Torrent system. The suite consists of five programs; *arrayDesc*, *fclPlnDesc*, *dewarDesc*, *dheDesc* and *sysConfig*. A complete explanation of their use is included in NOAO document TRNT-AD-08-0006, Torrent Software User Manual *arrayDesc*, *fclPlnDesc*, *dewarDesc*, *dheDesc* and *sysConfig*.

### 2.2.3 *mborg* Test and Optimization

Provide software to automate the testing of Torrent Boards. (May be done with LabView??)

lcbTest – verifies the functionality of an LCB board.

afeTest – verifies functionality of an AFE board.

mezTest - verifies functionality of a mezzanine board.

preAmpTest – verifies functionality of a transition module pre-amp board.

utilTest - verifies functionality of a transition module utility board.

psmTest - ?? May be done with LabView.

- pteTest – a test to determine the serial and parallel photon transfer efficiency for a CCD.

- noiseTest - a test to determine the system noise for a CCD or IR system.

- features - a test to determine the location and extent of “hot” and “bad” pixels, rows and columns.

-optimize – a series of sequences to ease the task of optimizing a focal plane or detector TBD.

-eepEdit –this tool allows the engineer to write the optimization voltage values to the TSM EEPROM and/or edit the EEPROM values by hand.

- Done after engineer is satisfied with optimization results.
- Hand editing is “possible but not recommended.”

### 2.2.4 *collector*

## 3.0 Torrent Auto-configuration

Torrent systems and software are designed to auto-configure themselves. This is done in several stages. First, the *assimilate* program will read the FPGA project build file for the current firmware version and extract comment lines of a particular format. The program will interpret the lines and create a set of .cfg files, one for each module described in the comment lines. These .cf” files and a software .cfg are then used to create a template Tmplt.csv file. These .cfg and Tmplt.csv are unique to each version of the firmware running in a Torrent system.

The second stage of the process is carried out by the sysConfig suite of programs. These programs act as an engineer’s assistant to help define, record and document the detector controller design. See Section 3.1 for an explanation of the sysConfig suite.

The third stage of the process is the testing and optimization of the focal plane using *mborg* . Section 3.2 explains how this process creates the data to be stored in the EEPROMS.

At runtime a fourth program called *collector* will read the .cfg files and use them to obtain information from the EEPROMs associated with each module. This information will then be combined to create a *sysName\_config.csv* that is read by the PAN processes to configure and control the detector.

In addition, separate `torrent_TSM.cfg` and `sysName_SFTW.cfg` files are created during the system configuration and detector optimization process. These files are used by *collector* to read the default, maximum and minimum values for every hardware function connected to the Dewar from the TUB EEPROM and create a `sysName_DefaultSetup.mod` and a `sysName_SetVoltages.mod` file to be used by the initialization routines at system startup to set all attributes to optimum levels. See Section 3.3 for a detailed description of this process.

### 3.1 `sysConfig`

### 3.2 `mborg` and Detector and Focal Plane Testing/Optimization.

### 3.3 Connecting Registers to the Detector Functions. (TSM .cfg File and EEPROM)

The TUB 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 .cfg files described above. The integrating engineer will use the *sysConfig* tool to create a file that connects a hardware register function to a detector requirement.

The TSM.cfg files will be constructed by *sysConfig* and have the format described in an appendix in the document TRNT-AD-08-0006, Software User Manual for `arrayDesc`, `fclPlnDesc`, `dewarDesc`, `dheDesc`, `sysConfig`.

### 3.4 Torrent Configuration and Calibration

#### 3.4.1 `sysConfig`

During the testing and calibration of the boards the EEPROM tool *pStor* will create and load the EEPROM images for the boards/modules to be used in a system.

Next, the integrating engineer will use `sysConfig` to create a `torrent_TSM.cfg` file containing the correspondence between user or detector signal names and DHE register names and will include appropriate help. This file will be used at the end of detector optimization to load the TUB EEPROM with the correct values.

At the system startup the PAN, LCB, PSM and CCD# or IRFE# and TSM .cfg files, and the LCB, PSM, CCD# or IRFE# and TUB EEPROMS will be read and a .csv file containing the required attributes will be created. The TSM.cfg file and TUB EEPROM will be used to create a `sysName_DefaultSetup.mod` file to be loaded to bring the detectors to their operating mode or reboot of the system.

## 3.5 Torrent Runtime Routines

### 3.5.1 borg

### 3.5.2 borgMec

## 4.0 Version Numbers

Each FPGA source file is named and has inside a version number as an example the file /MNSN..../AFE\_Control\_Ver103/AFE\_AcqControl.vhd has a comment line

```
-- VERSION      : 1.03
```

near the beginning. Likewise the EEPROM and FPGA have version numbers embedded in them as well. I'd like to suggest that these be meaningful to the system so that at startup the software can deal with a possible mismatch in version number.

As far Dave and I can tell, regardless of the actual version, the only time a mismatch will be important is if the memory map of the firmware has changed. We thought that one part of the version number should be used to describe the memory map being used. If we use the first number then any version 1.xx system could be safely run with any version 1.xx boards, files and EEPROMS. We could also add a section to the version number to designate the memory map as in 1.03.xx where the 03 is the memory map version and subversions. Bug fixes and so forth would be designated by the xx.