WP3: Time-Tagging

Corbin Covault, Robert Sobin, Robert Halliday, CWRU

21 Jan 2015, slightly updated 29 Jan 2015

1 WP3 document and subdirectory List:

The following document files and directories are included:

```
wp3_cwru_overview_docs_for_cdr_20150122.pdf     (this file)

wp3_cwru_ttag_module_specs_20150121.pdf
wp3_cwru_ttag_module_status_update_20150129.pdf
wp3_cwru_ttag_sys_blockdiag_20150121.pdf
wp3_cwru_ttag_schematic_20150122.pdf
wp3_cwru_ttag_logic_analyzer_photos_20150122.pdf

wp3_cwru_ttag_gpsctrl_and_bringup_20150121.pdf
wp3_cwru_lab_images_captions_20150122.pdf

wp3_cwru_m12m_performance_studies_20150121.pdf

WP3_M12M_thermal_tests_timeseries_plots_Dec2014  (subdirectory)
WP3_SD_temperatures_vs_time_study_plots   (subdirectory)
WP3_Vendor_Documents   (subdirectory)
```

2 WP3: Specific Requirements

Here we briefly articulate the requirements of the WP3 task as they are understood by the members of the group at Case Western Reserve University (CWRU).

2.1 Hardware: Select GPS Receiver for time tagging

The GPS receiver engine (daughter board) that was used in the original Surface Detector Electronics (SDE) Unified Board (UB) is obsolete and no longer manufactured. We are tasked with finding a modern replacement and understanding which algorithms require adaption based on the new hardware.

We also take on the related task of characterizing and verifying the performance of the selected GPS boards in the lab. This requires the development of a time-tagging test bench.

2.2 Firmware: Develop and implement time-tagging firmware module on UUB

We are tasked with designing and developing a firmware module for the Time-Tagging function to be implemented for the Upgraded Unified Board (UUB).

To accomplish this, we are required to work within the development hardware and software framework (Xilinx Zynq-7000 FPGA and co-processor with Vivado development set). We are required to coordinate this work closely with other groups that will be developing firmware and software that will interact with the GPS in any way.

2.3 Firmware: Develop and implement serial I/O to GPS receiver on UUB

We are tasked with developing a UART-based serial I/O channel within the UUB software environment.

2.4 Software: Develop and implement driver for time-tagging

We are tasked with developing and document operating-system drivers that will provide software access to both the time-tagging firmware module and the GPS serial I/O channel.

2.5 Software: Develop and implement software for GPS initialization and control

We are tasked with developing onboard software for initializing and controlling the GPS and time-tagging systems on the UUB.

2.6 General: Develop local expertise with board development tools

We are tasked with developing a general expertise with the hardware, firmware and software environment, operating system, and development tools. We will work very closely with efforts coordinated by our partners in each of these areas.

3 WP3: Design Concept and Design Implementation

3.1 GPS Receiver Selection

We have selected the I-Lotus M12M Timing GPS Receiver manufactured by I-Lotus, LLC (Singapore). We are motivated for this selection based on the following rationale:

- The M12M Timing receiver is designed to be functionally compatible with the Motorola Oncore UT+ GPS receiver that is currently used within the Auger SDE Unified Board. Choosing a compatible unit means that fewer and simpler modifications to the basic time-tagging system design. Specifically, the M12M provides the same 1 PPS timing output with a "granularity correction" (so-called "negative sawtooth") of "2 nanosecond" (see I-Lotus M12M Timing brochure `m12mt_brochure.pdf`, included.) This accuracy is very good relative to the UUB specification to achieve better than 5.0 nanoseconds RMS accuracy (a number that ensures that GPS timing errors are small compared to time resolution imposed by 120 MHz FADC readout.)
- The CWRU group has already implemented a prototype test stand to calibrate and validate this particular GPS engine. We have already ported all software associated with operating the M12M Timing receivers on the lab bench and in the field.
- The availability and the cost of individual units is favorable, and not a major cost driver.

3.2 Time-tagging firmware module

The CWRU has developed a preliminary specification and design for the required Time Tagging firmware module. The fundamental architecture parallels the time-tagging design concept used in the original UB and planned for Auger North. Details are presented in the document `wp3_cwru_ttag_module_specs_20150121.pdf`. In particular:

- In consultation with groups at MTU and OSU, we have implemented an AXI slave peripheral interface to provide input and output ports, presenting data and providing control registers. These have been implemented with proper hand-shaking to ensure data integrity.
- The module presents a sent of thirteen data registers corresponding to required timing data.
- See `wp3_cwru_ttag_schematic_20150122.pdf` for a detailed schematic of the time-tagging modules as currently developed.
- See `wp3_cwru_ttag_sys_blockdiag_20150121.pdf` for a detailed block diagram showing how the time-tagging module connects to the larger system.

3.3 Serial I/O to GPS receiver

The CWRU has developed a UART-based serial I/O channel for data transfer and control to/from the M12M GPS receiver. See `wp3_cwru_ttag_gpsctrl_and_bringup_20150121.pdf` for details.

3.4 Driver for time-tagging

The CWRU group has responsibility for developing and implementing an OS-level Linux driver as a the sole mechanism for software access and control the time-tagging module and GPS hardware. These will be implemented in accordance with the driver specification document by P. Alison, OSU.

3.5 Software for GPS control: gpscntl

The CWRU group has responsibility for developing and implementing onboard software for initialization of the time-tagging modules, GPS hardware control, and timing data. Our design is to use the original software `gpscntl` as implemented on the original UB as a framework, forking changes and modifications as needed for the UUB. See
`wp3_cwru_ttag_gpsctrl_and_bringup_20150121.pdf` for details.

4 WP3: prototype test bench results

4.1 GPS stability vs. thermal variance

The CWRU group has a long-established program of testing and calibrating the timing performance of GPS receivers. The timing accuracy achievable from a given GPS receiver depends on the intrinsic accuracy of the 1 PPS output pulse and the rate and (to a lesser degree) stability of the dedicated oscillator used to drive the time-tagging counters.

During 2013, our group became concerned about evidence that the timing accuracy of the M12M might systematically degrade during large thermal shifts. Certain M12M units seemed subjects to swings as large as 8 to 10 nanosecond as a result of thermal ramping characteristic of standard thermal test cycling routines.

While we were investing this issue, however, I-Lotus announced a product line upgrade corresponding to equivalent functionality but with improved thermal timing stability. Figure 1 shows comparative plots of temperature and timing offsets (differences between a test receiver and a reference receiver) as a function of time. As can be seen, the timing stability of the new version of the M12M Timing receiver is much improved relative to the old version.

Figure 2 shows a histogram indicating the RMS timing accuracy as measured for twenty of the newer M12M Timing receivers during multi-hour thermal test chamber testing meant to mimic extreme temperate variations recorded on actual SD stations in the Auger SDE. As can be seen, all twenty of the test receivers demonstrate accuracy better than the 5.0 ns specification required.

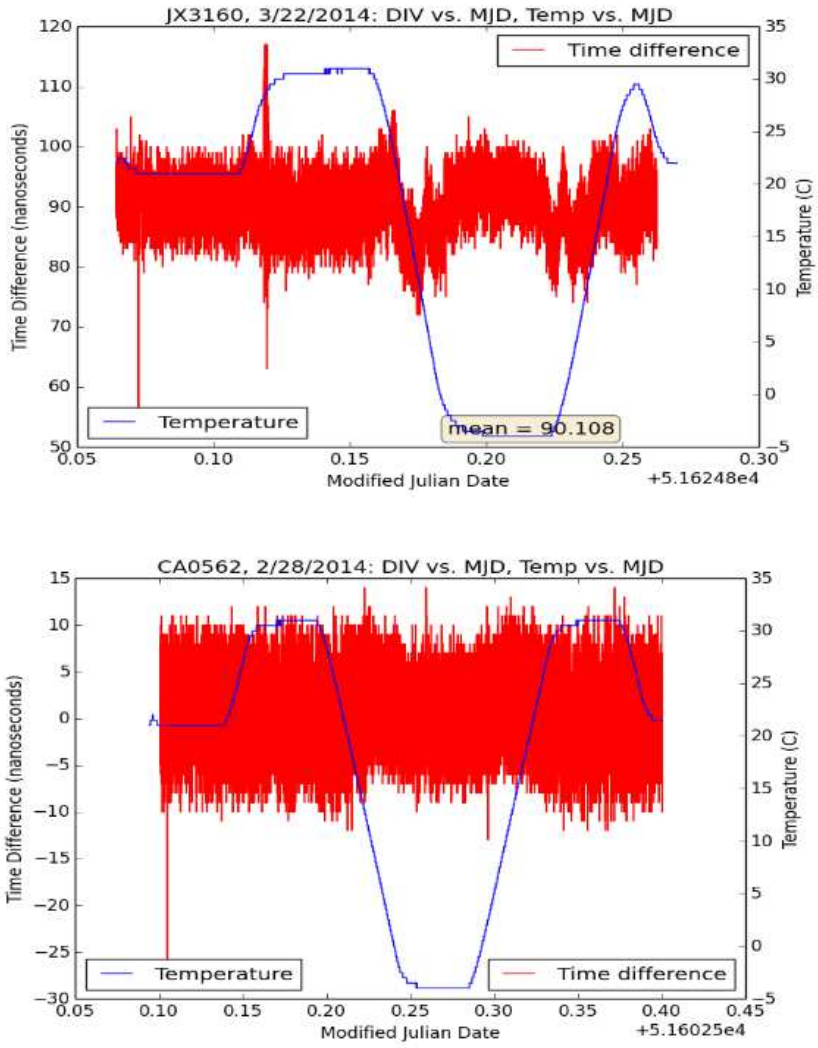

Figure 1: Comparative time stability of older version of the I-Louts M12M Timing vs. the newer version (after January, 2013).

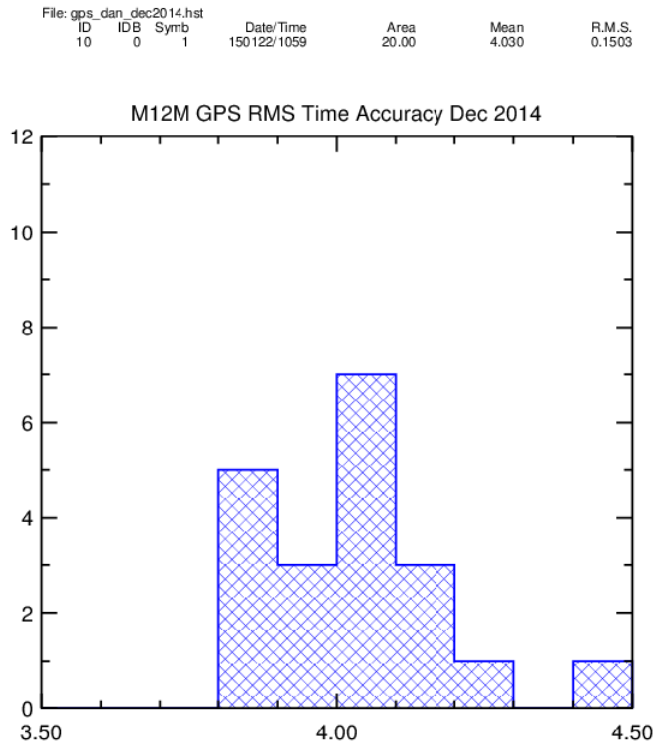

Figure 2: RMS timing differences (reference vs. test receivers) for twenty newer-version (after January 2013) M12M Timing GPS receiver tested in the thermal test chamber at CWRU with temperature cycling corresponding to "extreme Malargue" conditions. The required accuracy for all twenty receivers is significantly less than the 5.0 ns specification required. See document:
`wp3_cwru_m12m_performance_studies_20150121.pdf` for details.

4.2 Bench test results of time-tagging module

We have a working test bench for verifying the performance of the time-tagging module. See `wp3_cwru_ttag_logic_analyzer_photos_20150122.pdf` for details.

5 WP3: Design and development status

5.1 GPS test bench

We have one working test-bench that can be configured for either Motorola Oncore UT+ or I-Lotus M12M Timing GPS receivers based on Xilinx PLD.

5.2 GPS receiver procurement, testing, and calibration

We have twenty "newer version" M12M receivers which are now fully tested, characterized, and calibrated. These units are ready for deployment into the first prototype UUB boards.

5.3 Status of time-tagging Module

We have a preliminary design of the time-tagging module complete and working. See `d wp3_cwru_ttag_module_specs_20150121.pdf` for details. Performance of the module has been verified using virtual i/o and logic analyzer tools. See `wp3_cwru_ttag_logic_analyzer_photos_20150122.pdf` for examples.

5.4 Status of serial I/O to GPS module

We have a robust working UART based serial connection implemented on the Zedboard and we have used this to control and readout serial data to/from M12M GPS receivers. See `wp3_cwru_ttag_gpsctrl_and_bringup_20150121.pdf` for details.

5.5 Status of time-tagging driver

We have a definition of the the drivers and we are working toward an implementation.

5.6 Status of software for GPS initialization and control: gpscntl

We have made substantial progress in working systematically through the code for `gpsctrl` with preliminary modifications made to most sections of the code. See `wp3_cwru_ttag_gpsctrl_and_bringup_20150121.pdf` for details.

5.7 Status of work to extend local expertise with board development tools

5.7.1 Installing Petalinux

Work is progressing toward the installation of a Linus operating system on the Zedboard processor.

5.7.2 Developing Board Support Package

We are making our way through the example documentation. Our goal is to implement a "from scratch" board-bringup on the Zedboard under Linux as a demonstration project for doing the same within the UUB.