

CLOCK SYNCHRONIZATION OF A BROADBAND SEISMOMETER THROUGH IEEE-1588 PROTOCOL

Oriol Pallarés, Joaquin Del Río, Shahram Shariat-Panahi

SARTI Research Group. Electronics Dept. Universitat Politècnica de Catalunya (UPC), Rambla Exposició 24, 08800 Vilanova i la Geltrú, Spain. Phone: +(34) 938967200, Fax: +(34) 938967201

Abstract- In seismology, the time of the signal acquisition is highly important in order to know the magnitude and location of the earthquake. This paper presents the tests carried out to synchronize the seismometer clock through the IEEE-1588 protocol.

Introduction

Clock synchronization of seismometers is a key point to find the exact location of the earthquake. When seismometers are installed at seafloor observatories, data is collected through a marine cable and by using the Ethernet protocol. In IEEE-1588 protocol [1], instruments are synchronized through an Ethernet connection. They exchange timing data with a master clock through Ethernet in order to synchronize the clocks with an error below a microsecond. Therefore, instruments MUST be able to receive timing data from an external master clock. We have used the Trillium 120P/PA Broadband seismometer [2] together with Taurus data logger [3] to implement the clock synchronization Through IEEE-1588.

In order to equip Taurus with IEEE-1588 time synchronization capability, Taurus needs to receive timing data (TSIP: Trimble Standard Interface Protocol) and trigger (PPS) from an external clock. In a normal Taurus, GPS receiver [4] inside Taurus provides this information. In Figure 2, a description of how Taurus can be equipped with IEEE-1588 time synchronization capability is found.

We have been using Stellaris Luminary LM3S9B96 embedded system, Figure 1 [5], which supports hardware IEEE-1588 Precision Time Protocol (PTP). This module is a small embedded system. In our OBSEA seafloor observatory we plan to attach LM3S9B96 to Taurus externally, prior to its deployment.



Figure1: LM3S9B96 Board

It would be in charge of implementing the IEEE-1588 protocol with the master clock located at the shore station (PC), and also would provide Taurus with timing data (TSIP) and trigger (PPS) for time synchronization. Therefore, this module would substitute the GPS receiver inside Taurus. The block diagram of the overall system can be seen in the following figure(Figure 2):

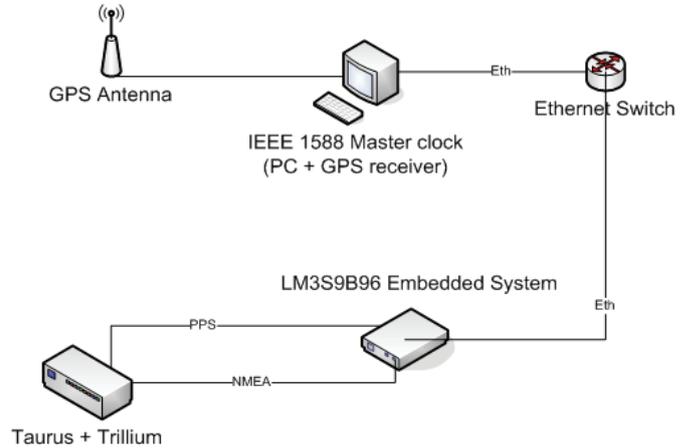


Figure2: Schematic functional diagram

In order to implement the system above, it is necessary to know the following information:

- 1- The data received by Taurus from the GPS receiver.
- 2- How to enable Taurus to receive this data from an external module (eg. LM3S9B96).

System implementation and tests

GPS receiver transmits to Taurus, using the TSIP protocol. Five frames every certain time interval. First of all, for initialization, it sends frames 0x45, 0x46, 0x4B, 0x4A and 0x41 that contains information about software version, receiver health, machine code/status, position output and GPS time. Then in order to notify Taurus that position fixes are available, it sends every second for four seconds, the frame 0x4A. And finally, repeatedly, sends every five seconds the packets 0x4A, 0x41, 0x46, 0x4B and 0x5C in this order. Last packet is sent to respond to the 0x3C received from the Taurus which contains information about satellite tracking status. Because the internal GPS receiver was removed from the Taurus, these frames are received via the serial port. Luminary board will perform this operation as shown in figure 2. It sends to Taurus a TTL signal through the serial port with all the frames mentioned above. It also sends a PPS signal in order to trigger Taurus synchronization.

To test the correct operation of the system, first we have to be sure that the frames sent to the Taurus are correct. To do this we implemented a program in LabVIEW which sends all the packets, and the PPS via Luminary board with enet_ptpd program. Every subVI implements a packet to be sent to the Taurus.

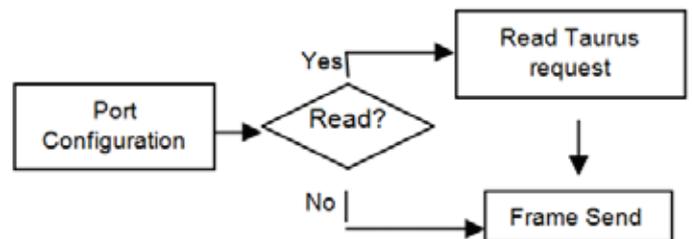


Figure 3: Packets sent for Taurus initialization

First packet is sent only for initialization, which is the first part of the sequence structure as we see in Figure 3, and the second part is the packet sent every time interval:

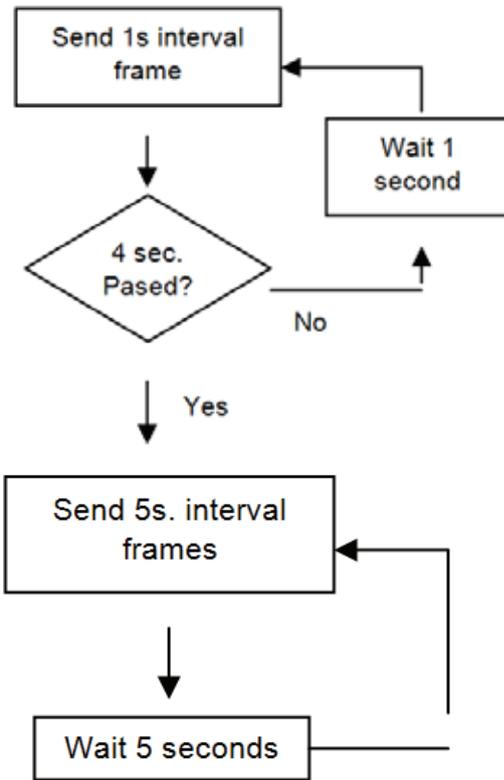


Figure 4: Packets sent every 5 seconds time interval

At same time, we can enable a read function which allows us to see if Taurus is sending a request, Figure 4. In the front panel of this VI we only have to configure the last Sunday-date in order to get correct timing frames.

Results

At present, we have managed to configure many parameters of the Taurus via serial port and we are now testing time synchronization of the Taurus. When the Taurus is working, we are able to find the timing information of Taurus, although timing precision is not much accurate. This can be due to incorrect time synchronization with shore PC. The Information of GPS satellites, and all the indicators from the status window of TAURUS, show that the system is operating correctly and the data is received in real time.

References

- [1] IEEE-1588 Precision clock synchronization protocol for networked measurement and control systems.
- [2] "Trillium 120P/PA Seismometer", Nanometrics Inc.
- [3] "Taurus Portable Seismograph User Guide". 2009. Nanometrics Inc.
- [4] Trimble "Lassen iQ GPS Receiver". 2005
- [5] Texas Instruments, "Stellaris LM3S9B96 Microcontroller". 2010.

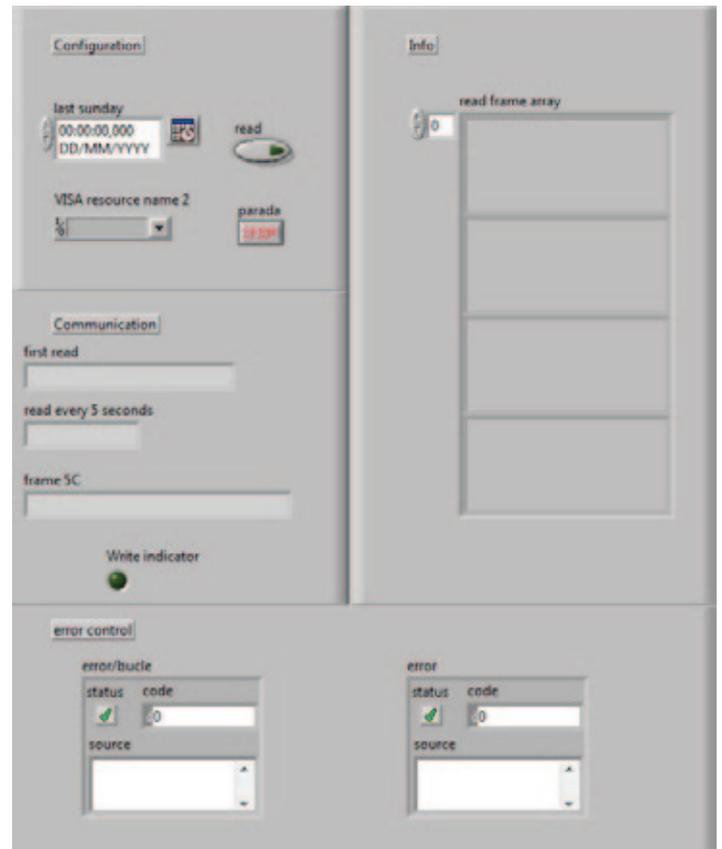


Figure 5: Front panel of the sending program

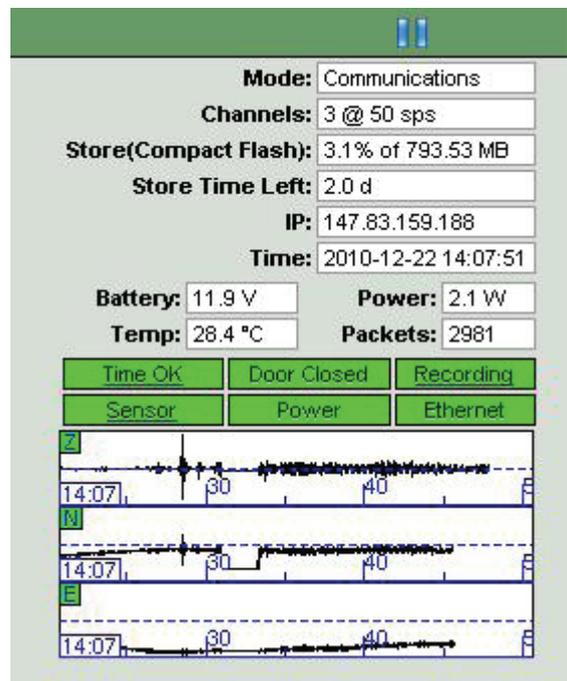


Figure 6 Taurus data logger Waveform window.