GIANT METREWAVE RADIO TELESCOPE

PPS Generation on Serial Port using Python and Processing It

PROJECT REPORT

Submitted To

Shri Sandeep Chaudhari

By

Dhiraj Tambade

STP 2012

GMRT Project, P. B. No. 6, Narayangoan,Tal - Junnar, Dist - Pune. State - Maharashtra

Index

Abstract

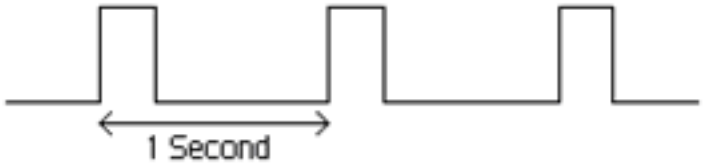
A Pulse per Second (PPS) signal is required as triggering input to packetized design. As per existing setup in GMRT, a long differential cable approx. length of 45m is required to be laid from GPS receiver to Digital Backend Lab. This is unrealistic in terms of cost and noise immunity.

To solve this problem, a PPS signal can be generated on serial port of computer which is locked to NTP server for precise timing. The PPS waveform is generated using Python programming and modules in it called Pyserial and Time. Using small MAX 485 circuitry, we will convert output of Serial Port in 0V-5V level i.e. Digital Logic.

What is PPS Signal?

A pulse per second (PPS) is an electrical signal that has a width of less than one second and a sharply rising or abruptly falling edge that accurately repeats once per second.

PPS signals are output by radio beacons, frequency standards, other types of precision oscillators and some GPS receivers.



Accuracy - 12 picoseconds to a few microseconds per second
2 nanoseconds to a few milliseconds per day

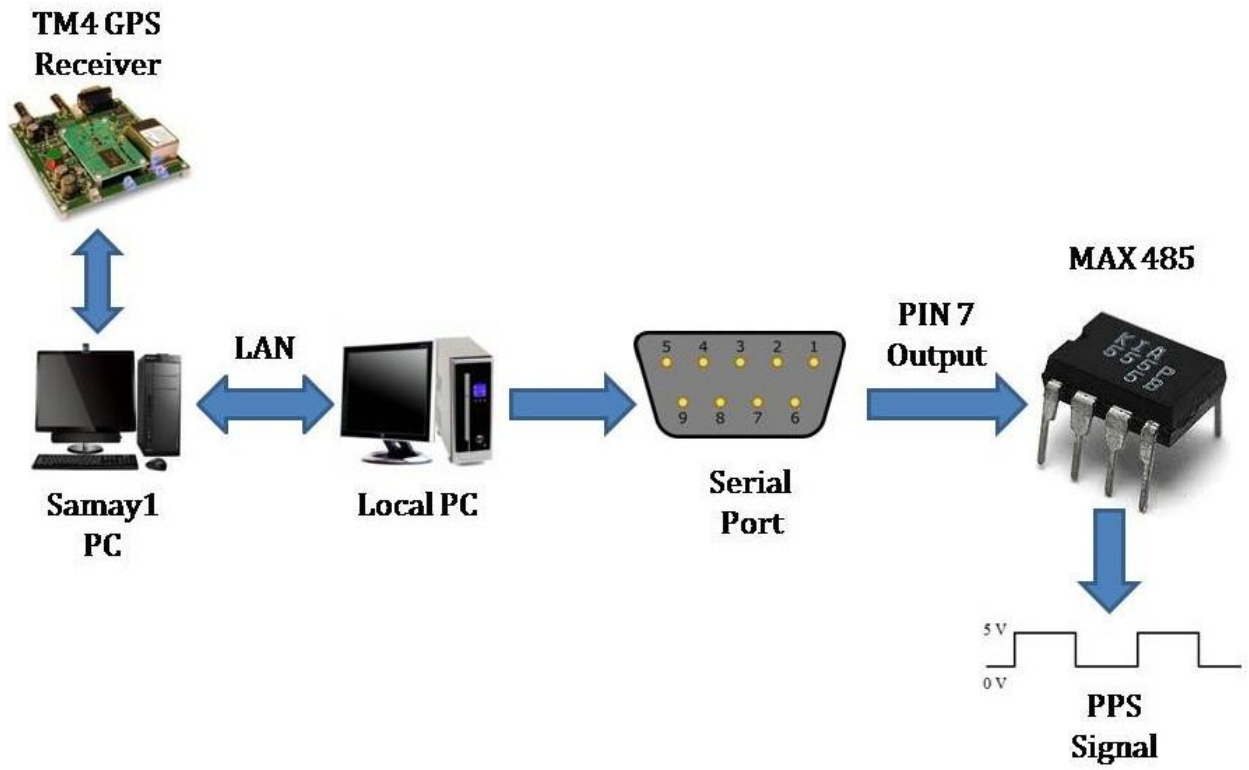
Use – Triggering input to Packetized Design
Precise timekeeping and time measurement

Common use for the PPS signal is to connect it to a PC using a low-latency, low-jitter wire connection and allow a program to synchronize to it.
This makes the PC a stratum-1 time source.

Note that because the PPS signal does not specify the time, but merely the start of a second.

One must combine the PPS functionality with another time source that provides the full date and time in order to ascertain the time both accurately and precisely.

Graphical Overview of Project



Hardware Used

- ✓ PC connected to samay1 and Ubuntu OS installed on it.
- ✓ Serial Cable
- ✓ MAX 485
- ✓ Resistors - 6.8k, 4.7k
- ✓ Connecting wires
- ✓ 5V Power Supply
- ✓ Tektronics DPO2024

Network Timing Protocol (NTP)

The Network Time Protocol (NTP) is a protocol designed for accurately synchronizing local time clocks with networked time servers.
NTP is distributed, hierarchical system.
Primary Servers are machines that are synchronized to external time sources
Secondary Servers allow thousands of machines/organizations to synchronize without overloading primary servers.

- ✓ GPS is considered a stratum-0 source.
- ✓ Samay1 PC is primary server connected to TM4 GPS receiver.
- ✓ Local PC is connected to primary server and can also be used as secondary server.

How to Maintain an Accurate System Clock with ntpd

1. Install the NTP daemon:

   sudo apt-get install ntp

2. Configure the daemon properly:

   The configuration file for ntpd is located at /etc/ntp.conf
   First section you may want to modify is the list of servers to synchronize with.

   # You do need to talk to an NTP server or two (or three).
   server samay1.gmrt.ncra.tifr.res.in

3. Make sure the configuration works

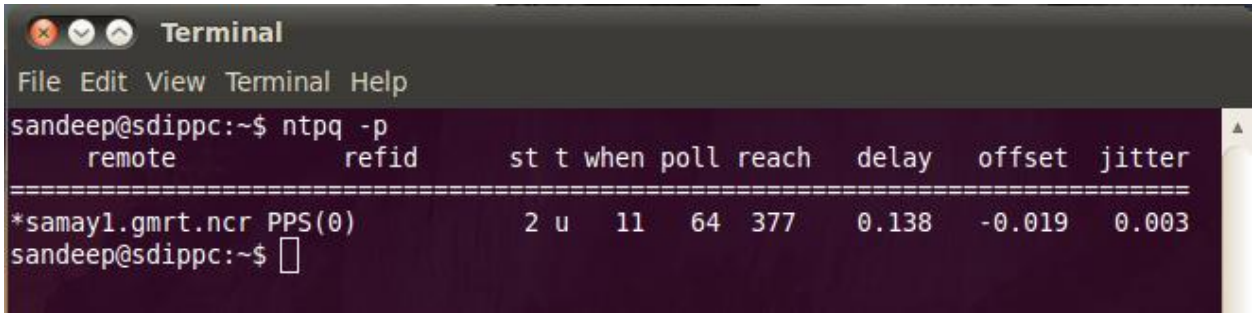
   Now that you have a proper server list in your /etc/ntp.conf file, it is time to run the daemon and see if you synchronize properly! Make sure you have an active Internet connection, and then run:

   sudo /etc/init.d/ntp restart

4. Check status of ntp:

ntpq –p

It will give result like this



- ✓ Outputs a two-line header and then one line per server, with 11 fields
- ✓ 1st char: '*' for the host we sync to ; '+' for OK ; '–' and 'x' for bad ; '' for unreachable; '0' for PPS sync source
- ✓ Hostname
- ✓ Where does it get time from?
- ✓ Its stratum
- ✓ Type of peer ('u' is for 'unicast')
- ✓ Seconds ago we heard from it
- ✓ How often do we currently poll, in seconds
- ✓ Octal reach ability mask, 377 means A-OK
- ✓ Delay (round-trip) to the server, in milliseconds
- ✓ Offset in milliseconds
- ✓ Jitter in milliseconds

Python

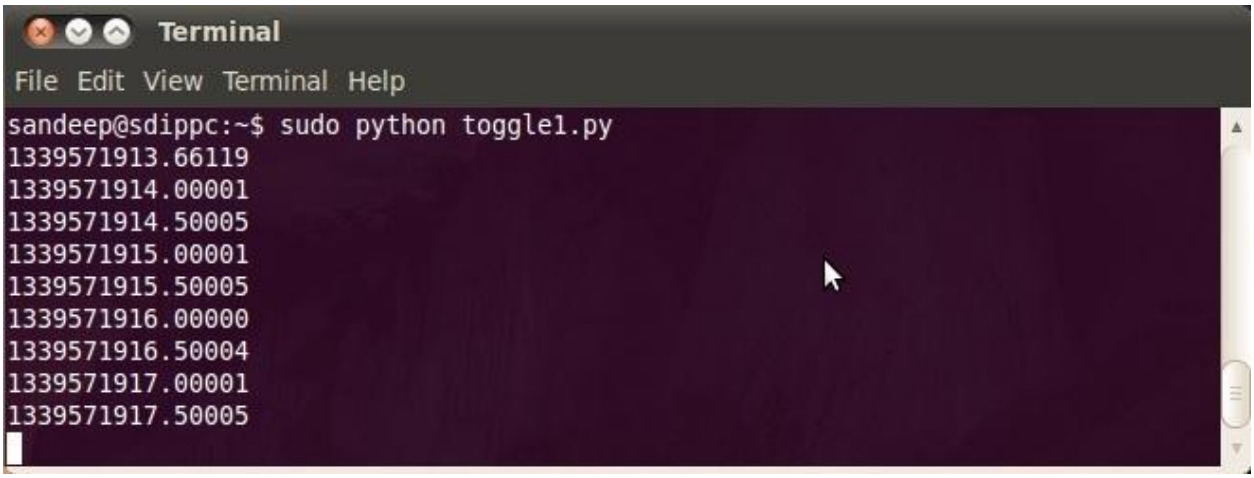
1. Install Python:
   sudo apt-get install ipython

2. Create file:
   sudo vi <filename>.py

3. Write the script given on next page and save the file.

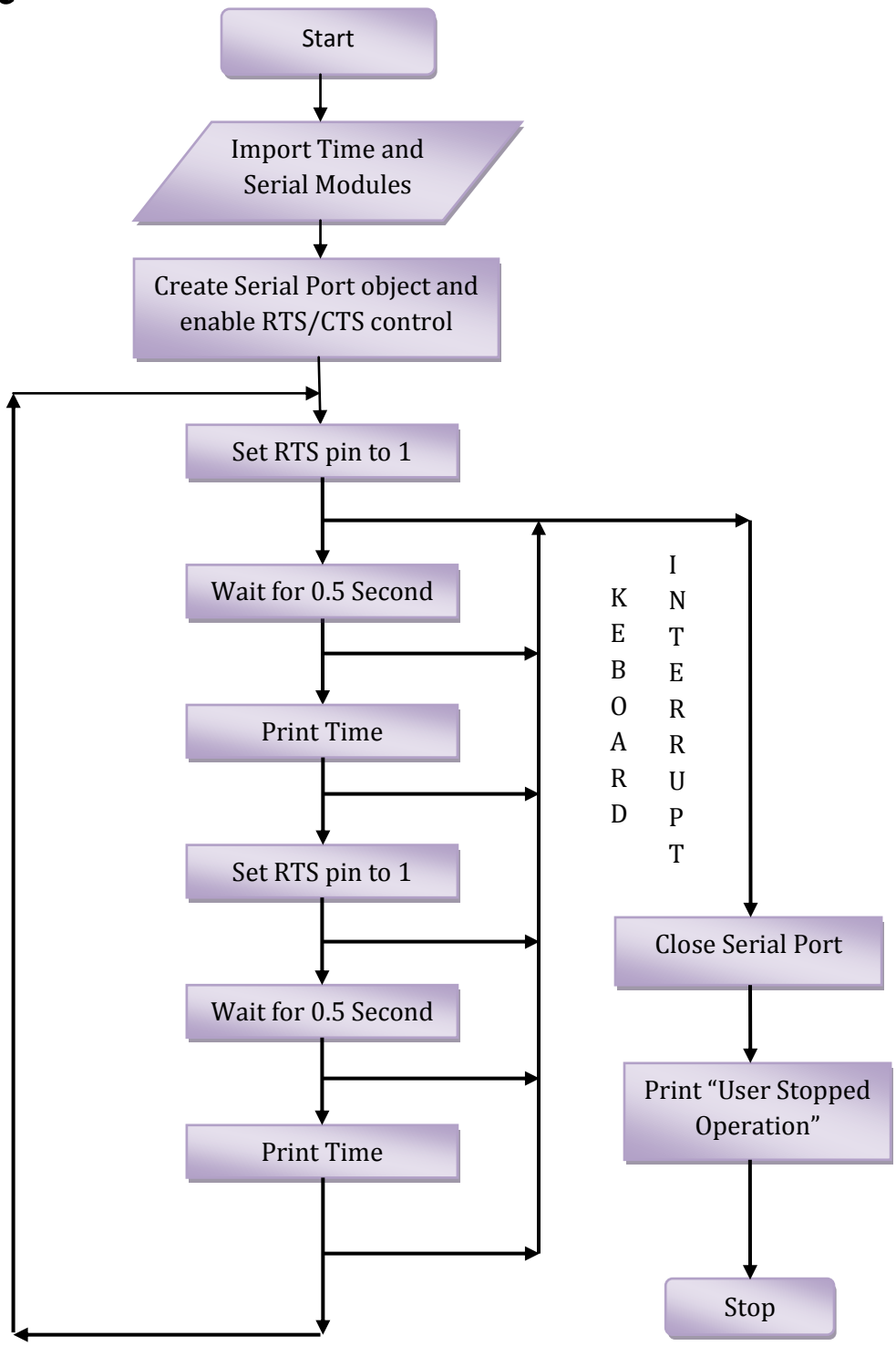
4. Run the program
   sudo python <filename>.py

5. Output will look like this



6. Stop operation only by pressing **ctrl+c**

Flow Chart


Start
Import Time and Serial Modules
Create Serial Port object and enable RTS/CTS control
Set RTS pin to 1
Wait for 0.5 Second
Print Time
Set RTS pin to 1
Wait for 0.5 Second
Print Time
KEBOARD INTERRUPT
Close Serial Port
Print "User Stopped Operation"
Stop

Python Script to generate PPS signal

```
#!/usr/bin/python
import serial
import time
ser=serial.Serial('/dev/ttyS0',rtscts=True)

try:
        while(1):

                x=time.time()-int(time.time())

                if (x+.5)>1:
                        x=(x%1)-0.5

                ser.setRTS(1)

                while( (x+0.5)>(time.time()-int(time.time()))):
                        pass
                print "%4.5f"%time.time()

                ser.setRTS(0)

                while ( (x+0.5)<(time.time()-int(time.time()))):
                        pass
                print "%4.5f"%time.time()

except:KeyboardInterrupt
ser.close()
print("\n User stopped operation")
```

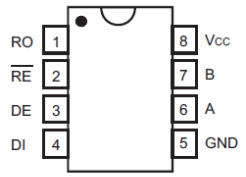
MAX 485 Circuitry

Need –

Output of serial port varies between ±15V

We need PPS Signal varying from 0V to 5V.

PIN CONFIGURATION



D OR P PACKAGE

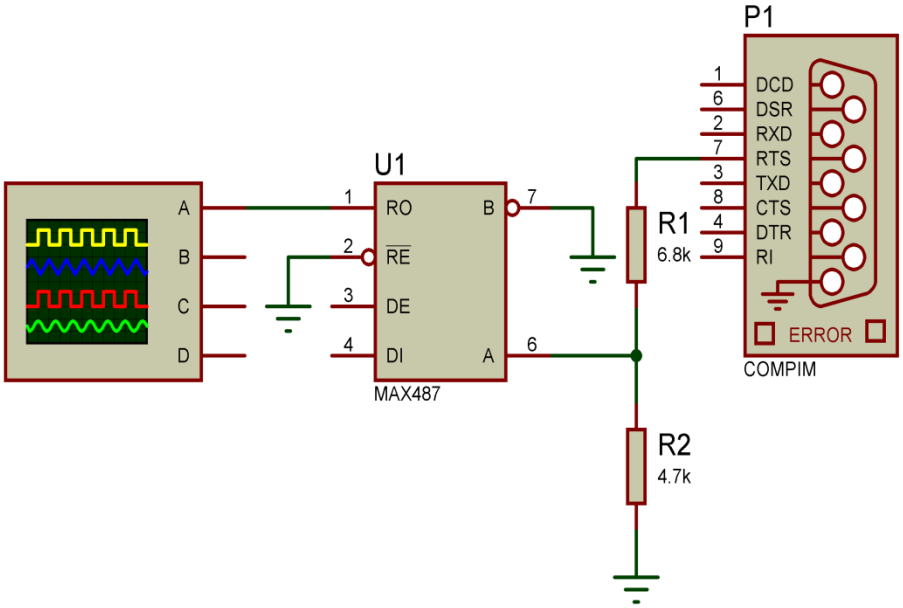
(Top View)

Features

- ✓ -7V to +12V Input Voltage Range
- ✓ 30ns Propagation Delays
- ✓ Operate from a Single 5V Supply

Circuit Description –

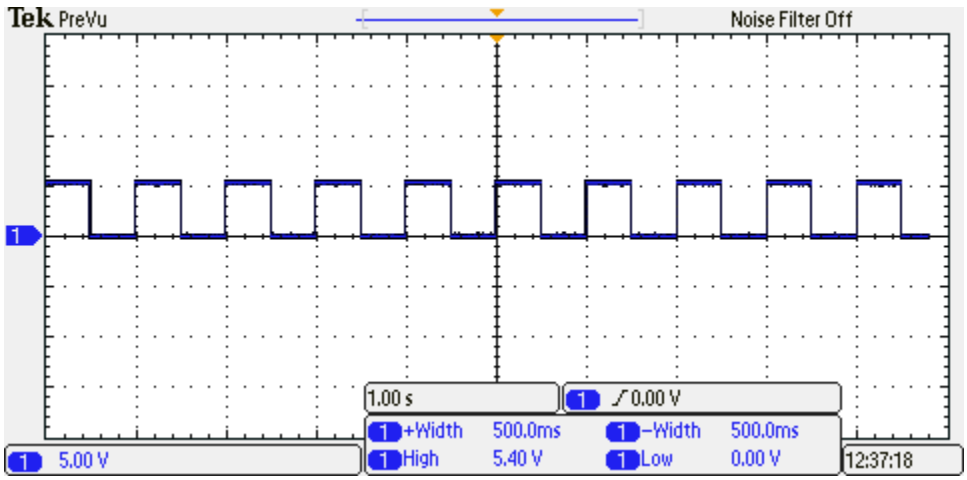
- ✓ Resistor Divider gives 40% of Serial Port output to A which becomes compatible for MAX 485.
- ✓ Receive output – If A > B by 200mV, RO will be high;
  If A < B by 200mV, RO will be low.
- ✓ Here we connected B to ground so that RO toggles with respect to A.



Results

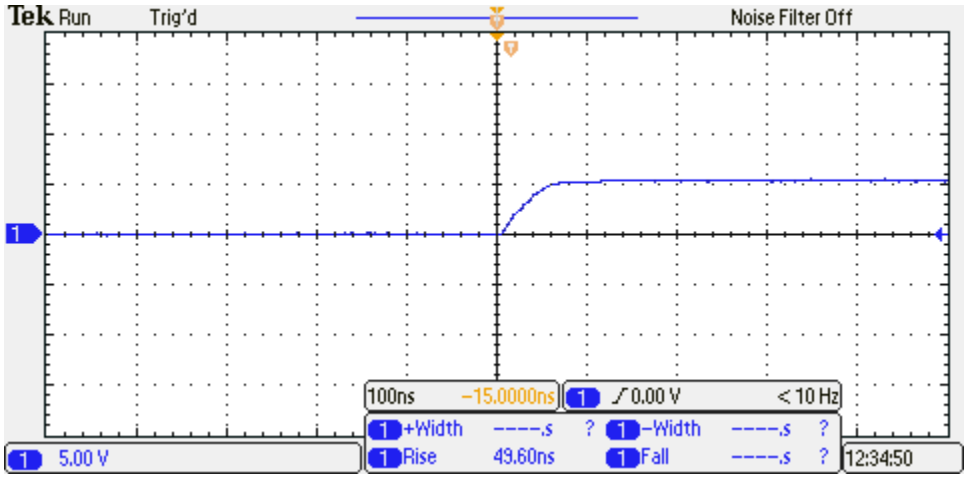
Following results were observed on Tektronics DPO2024

1. PPS Signal

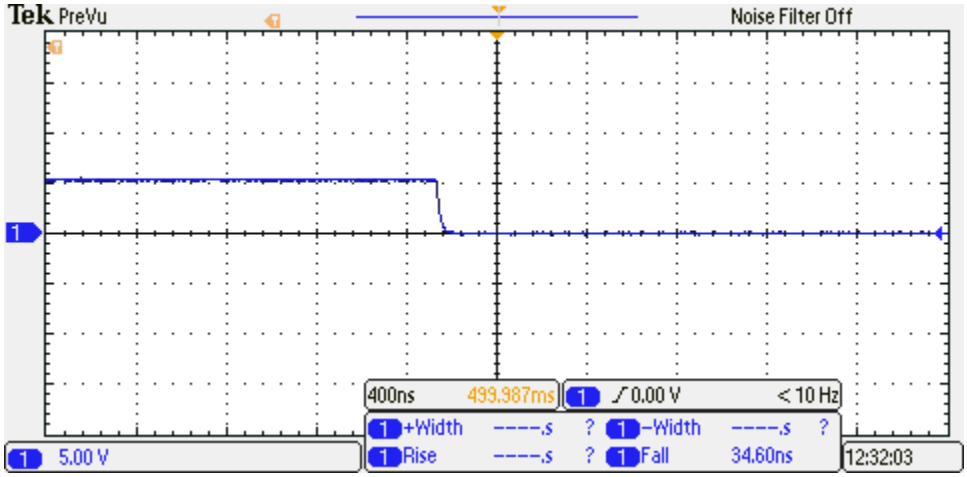


Period – 1 second
Amplitude – High: 5.4 V and Low: 0 V
Width – Positive: 500 ms and Negative: 500ms

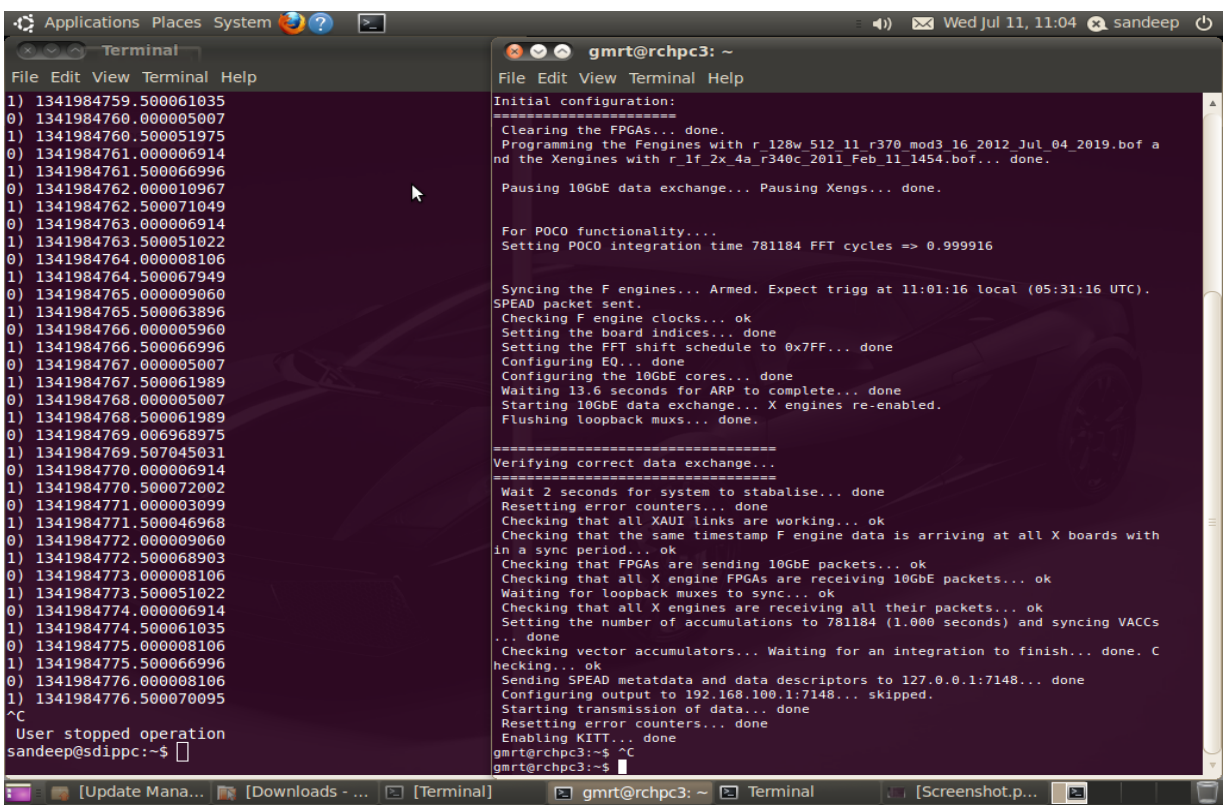
2. Rise Time: 49.60 ns



3. Fall Time: 34.60 ns



4. Testing Result – Successful Triggering Of FPGA



References

1. *www.en.wikipedia.org*
2. *www.ubuntuforums.org*
3. *www.ntp.org/*
4. *www.docs.python.org/py3k/tutorial/index.html*
5. *MAX 485 Datasheet*
6. *Network Time Protocol (NTP):Overview and Configuration*
   Stanislav Shalunov hshalunov@internet2.edui
   Performance Workshop, Atlanta, 2005-04-22