# POCO-2ANT-200MHz TEST PROCEDURE.

Version 2 , Released Date : 11/01/2012. Written By : IMH/MVM.
Version 1 , Released Date : 01/11/2011. Written By : IMH/MVM/SCC/GSJ.

Note : This colour : used for Main titles. This colour : used for sub/sub-sub titles.
This colour : used for commands. This colour : used for comments.
This colour : used for procedure writeup/note.
This colour : used for changable information.

**Contents:**

## 1. Hardware and software requiremets.

a. ROACH unit with iADC cards. b. PC for controlling & data acquisition.

c. Clock , PPS and Input signals. d. Cables : Serial & ethernet.

| | | |
|---|---|---|
| [CTRL_PC] | : | 192.168.4.68 |
| [USER] | : | gmrt |
| [CTRL_PC_PASS] | : | gmrttifr |
| [PYSCRIPT_DIR] | : | /home/gmrt/POCO_2ANT_200MHz/PY_SCRIPTS/ |
| [BOF_DIR] | : | /home/gmrt/POCO_2ANT_200MHz/BOF_FILE/ |
| [TAX_DIR] | : | /home/gmrt/POCO_2ANT_200MHz/TAX/ |
| [CONFIG_PYSCRIPT_FILE] | : | wb_init_400mhz.py |
| [DELAY_UPDATE_PYSCRIPT_FILE] | : | wb_dly_rate_update_400mhz.py |
| [PLOT_PYSCRIPT_FILE] | : | poco_plot_400mhz.py |
| [DUMP_PYSCRIPT_FILE] | : | poco_acq_400mhz.py |
| [BOF_FILE] | : | wb_r_nfs_pps_cnt2_v15_2012_Apr_25_1600.bof |
| [TAX_PROG_FILE] | : | xtrgsb32 |

## 2. Introduction

The POcket COrrelator (POCO) is , single FPGA based correlator. This design accepts signals from 2 antenna's single polarization through iADC. The ADC samples at 400 Mhz clock giving 8bits of digitized data per pol. The digitized data goes through Delay correction followed by 1024 point FFT , FSTC , Fringe correction is done. Then data is integreated for 1 sync ( 1 sync = 1.34 seconds). The integreated data goes to control PC through 100Mbps link.

## 3. Setup

1. Connect the Serial port cable between the ROACH board's P2 connector and serial port of the CTRL_PC (on which minicom program exists). This needs to be done for debugging if needed.

2. Connect the Ethernet cable to J25 port of the ROACH board from the CTRL_PC's eth1 port.

3. Confirm the following entries :

With the present setup roach-030140 is connected. Which corresponds to the IP address of 192.168.100.40. Make sure the entries for this roach board in the following files. Ideally the entry for all the ROACH boards in the GMRT has been done in all the files.

a. ip address and corresponding roach board (host) name in the file /etc/hosts.

b. mac address and corresponding ip address in the file /etc/ethers.

c. eth1 entry in the file /etc/network/interfaces file.

4. Feed the following inputs to iADC0 card attached to the ROACH unit.

a. PPS signal to Sync input.

b. Clock signal of 400MHz@0dbm to -2dbm to Clk_i input.

c. Input signals may be antenna signals or from Noise source. This should be max. 200MHz@ -12dbm to -17dbm ( Total power over BW ) to I+ & Q+ inputs.

## 4. Getting ready by updating the information in header (*.hdr) files.

**This needs to be done only if antenna signals are used as input signals.**

The *.hdr and python scripts are located in the directory “[PYSCRIPT_DIR]” .

1. Edit the 2 header files for antenna and source information

a. Antenna used : Edit in sampler.hdr

eg. SMP000 = C09 USB-130

SMP001 = S02 USB-130

b. Source Information : The RA & DEC information required can be obtained from by following these steps ;

Login: observer@shivneri

Password: obs@gmrt

Enter the following at the command prompt

```
observer@shivneri:/home/observer 32> work
observer@shivneri:/home/observer/05aug2006/work 35>./user5.5
50                  # user id number
gts '3c48'          # If your source is '3c48'
```

Copy the 4th and 5th Appr. values from the "Process out" entry and paste as the I and II arguments in the line of '3c48' in source.hdr file. Remove the comment (#) for '3c48' and comment all other lines , if any.

```
>exit               # to exit.
```

## 5. Transferring the Design to the FPGA/ROACH unit.

### a. Transferring the Design to the FPGA.

1. Run "[CONFIG_PYSCRIPT_FILE]" from the directory [PYSCRIPT_DIR] to program the FPGA.

***[PYSCRIPT_DIR]/[CONFIG_PYSCRIPT_FILE]***

eg. cd /home/gmrt/POCO_2ANT_200MHz/PY_SCRIPTS/

```
./wb_init_400mhz.py 192.168.100.40
```

## 6. Updating the delay & fringe values and plotting the results to verify the Phase and Amplitude

### a. Loading the delay & fringe values after every sync interval.

1. Run "[DELAY_UPDATE_PYSCRIPT_FILE]" from the directory [PYSCRIPT_DIR] to update delays.

***[PYSCRIPT_DIR]/[DELAY_UPDATE_PYSCRIPT_FILE]***

eg. cd /home/gmrt/POCO_2ANT_200MHz/PY_SCRIPTS/

```
./wb_dly_rate_update_400mhz.py --dly_offset=0 0  --fract_offset=0 0  --frn_offset=-1.0 -1.0 -i -f -s -t -l 20000000
```

Adjust the dly_offset values till you get the correct Phase. Verify the Phase using plotting commands given below.

### b. Online plotting of results

1. Run "[PLOT_PYSCRIPT_FILE]" from the directory [PYSCRIPT_DIR] to plot the self, cross and the phase.

***[PYSCRIPT_DIR]/[PLOT_PYSCRIPT_FILE]***

eg. cd /home/gmrt/POCO_2ANT_200MHz/PY_SCRIPTS/

```
./poco_plot_400mhz.py 192.168.100.40 -l
```

## 7. Loading the desired delay & fringe values and Dumping the result on the "control PC : ctrlpoco (192.168.4.68)".

After statisfactory Amplitude and Phase plots use the -p and -q options to record the delay values in the log file. And then stop the online plot command and start the online dumping of results.

### a. Loading the desired delay & fringe values after every sync interval.

```
./wb_dly_rate_update_400mhz.py --dly_offset=0 54  --fract_offset=0 0  --frn_offset=-1.0 -1.0 -i -f -s -t -l 20000000 -p /home/gmrt/POCO_2ANT_200MHz/DATA/c9s2_3c48_080512.log -q /home/gmrt/POCO_2ANT_200MHz/DATA/c9s2_3c48_ 080512.log
```

### b. Online Dumping of results in a file for offline analysis using TAX program.

1. Run "[PLOT_PYSCRIPT_FILE]" from the directory [PYSCRIPT_DIR] to acquire the self, cross and the phase.

***[PYSCRIPT_DIR]/[PLOT_PYSCRIPT_FILE]***

eg. cd /home/gmrt/POCO_2ANT_200MHz/PY_SCRIPTS/

```
./poco_acq_400mhz.py 192.168.100.40 -f /home/gmrt/POCO_2ANT_200MHz/DATA/c9s2_3c48_080512.data
```

## 8. Data analysis using TAX program on the "control PC : ctrlpoco (192.168.4.68)".

1. Run "[TAX_PROG_FILE]" from the directory [TAX_DIR] to plot the self, cross and the phase.

***[TAX_DIR]/[TAX_PROG_FILE]***

eg. cd /home/gmrt/POCO_2ANT_200MHz/TAX/

```
./xtrbsb32                 # for usage details .....
```

*USAGE: ./xtrgsb32 -v visibility file -o outfile -r refant -b baselines -c chansel -t timesel -n 0/1*

*e.g. below*

***USAGE: ./xtrgsb32 -v raw.dat -o raw.out -r C00 -b C01,C02,W01,W05,S01,E05 -c 45,1100,1,5 -t 1,110,2,5 -n 1***

***: Chan selection format is start_chan,stop_chan,chan_incr,chan_integ***

***: Time selection format is start_time,stop_time,time_incr,time_integ***

**a. Plot self spectrum**

**i. for Ch 1–500 of time stamps 1-1000**

```
./xtrgsb32 -v /home/gmrt/POCO_2ANT_200MHz/DATA/c9s2_3c48_080512.data -c 1,500  -t 1,1000
```

**ii. for Ch 1–500 of time stamp 400.**

```
./xtrgsb32 -v /home/gmrt/POCO_2ANT_200MHz/DATA/c9s2_3c48_080512.data -c 1,500 -t 400
```

**iii. for Ch 10–500 of time stamps 660,690 in steps of 10**

```
./xtrgsb32 -v /home/gmrt/POCO_2ANT_200MHz/DATA/c9s2_3c48_080512.data -c 10,500  -t 660,690,10
```

**b. Plot cross spectrum**

**i. for Ch 10 – 500**

```
./xtrgsb32 -v /home/gmrt/POCO_2ANT_200MHz/DATA/c9s2_3c48_080512.data -c 10,500  -t 1,1000 -n 1 -r C00
```

**ii. for Ch 10 – 500 of time stamp 660.**

```
./xtrgsb32 -v /home/gmrt/POCO_2ANT_200MHz/DATA/c9s2_3c48_080512.data -c 10,500  -t 660 -n 1 -r C00
```

**iii. for Ch 10 – 500 of time stamp 660 to 690 in steps of 10.**

```
./xtrgsb32 -v /home/gmrt/POCO_2ANT_200MHz/DATA/c9s2_3c48_080512.data -c 10,500  -t 660,690,10 -n 1 -r C00
```

**c. Plot cross Ch – 310 over time**

```
./xtrgsb32 -v /home/gmrt/POCO_2ANT_200MHz/DATA/w4e6_3c48_040512.data -c 310  -t 1,1000 -n 1 -r C00
```

**d. Plot specific record(s)**

```
./xtrgsb32 -v /home/gmrt/POCO_2ANT_200MHz/DATA/c9s2_3c48_080512.data -c 10,500  -t 1,60,1,5 -n 1 -r C00
```

```
./xtrgsb32 -v /home/gmrt/POCO_2ANT_200MHz/DATA/c9s2_3c48_080512.data -c 10,500  -t 1,57,1,57 -n 1 -r C00
```