Version 3.3 ***Date : 21st October 2014.***

OPERATING PROCEDURE
FOR
RUNNING GWB - II

Authors :

Nilesh Raskar

Dharam Vir Lal

Sanjay Kudale

Harshavardhan Reddy

Table of Contents

Illustration Index

Abstract

This document will provide you the information and standard operating procedure (or SOP) in a way to configure and run the new broadband **GPU** based **GMRT Wide-band Back-end** (**GWB**) and its related applications.

Chapter 1: GWB PARAMETERS

1.1 Available gwb_config parameter selections and resultant values

GWB Parameter	GUI Selection	Resultants in hdr file
MODE	REALTIME	0
LTA	32	0.671088 * 32 = 21.474816 sec
	16	0.671088 * 16 = 10.737408 sec
	8	0.671088 * 8 = 5.368704 sec
	4	0.671088 * 4 = 2.684352 sec
	2	0.671088 * 2 = 1.342176 sec
	1	0.671088 sec
ACQ BW	400 MHz	400.0000 (Not released)
	200 MHz	200.0000
	150 MHz	150.0000 (Not released)
	100 MHz	100.0000 (Not released)
	50 MHz	50.0000 (Not released)
CHANNELS	512	512
	1024	1024
	2048	2048
	4096*	4096(Not released)
	8192*	8192(Not released)
	16384*	16384(Not released)
STOKES	Single-Pol-USB 130	1
	2 STOKES	2
	4 STOKES	4
CONTROL	ONLINE	1
TPA SELECTION	Online (tpa)	1
	Manual (GWB)	0
SIDEBAND FLAG	Flipped (LSB)	1
	Normal (USB)	-1
GAB LO FREQUENCY	LO 130 & LO 175	LO SET at GAB taken as RF for GWB.
GAIN	ON/OFF	1/0 respectively.
FSTOP	ON/OFF	1/0 respectively.
Beam – 1 / Beam - 2	OFF/IA/PA	0/1/2 respectively.
Beam Stokes	1 Stokes/ 4 Stokes.	1/4 respectively.
Beam Integration	128	No. of FFT's = 128 (sampling period 1.3ms)

Chapter 2: Running Interferometry Observations

Before starting the gwb correlator chain (from astro8). It needs to set up some Important parameters at the beginning viz., gpu-config, sampler and hosts settings etc. gwb-config generates a gpu.hdr file, which contains all necessary parameters with available options to set up the gpu correlator in different modes. And sampler and hosts settings window generates the sampler.hdr (for antenna connections to the gpu samplers) and hosts.dat (host machines entries with their socket ID's) files.

2.1 GWB Config Parameters

Below gives the parameter lists, which are saved/changed from the GWB Config window.

```
MODE            :       Sets the GWB operating mode (default is REALTIME, and selection is disabled
                        from GUI).
LTA             :       Minimum data acquisition rate for GWB, can be set to 1, 2, 4, 8, 16 and 32.
ACQ BW          :       selectable from the available bandwidth modes, viz. 50, 100, 150, 200, 400
                              MHz.
CHANNELS        :       User selectable range of channels available for the given GWB ACQ BW.
STOKES          :       STOKES parameter selections (Single POL USB-130, Total Intensity, Full
                        Stokes).
CONTROL         :       GWB control mode, either ONLINE or LOCAL/Manual. (default :ONLINE, selection
                        is disabled from GUI).
TPA Frequency   :
                    1)  Online : This will take TPA parameters from online machine, and disables the
                        Sideband Flag and  GAB LO entries at GUI.
                    2)  Local : This enables user to choose sideband, and  GAB LO Entries.
                        1.  Flipped (LSB- decreasing frequency over channels, RF < LO).
                        2.  Normal (USB – increasing frequency over channels, RF > LO).
GAIN            :       Default is ON.
FSTOP           :       Default is ON.
Beam1/Beam2     :       It can be set as IA(0) or PA(1) with beam stokes as 1 or 4.
Beam Integ      :       It can be set as 128 or 64 FFTs which corresponds to 1.3 ms or 0.65 ms
```

Now we present all three modes available with the GWB-II; for each mode we present a screen-shot and also provide the sample **gpu.hdr** file. Note that GWB (GPU) RF frequency is the same as GAB LO frequency.

2.1.1 Single POL CHAN 1

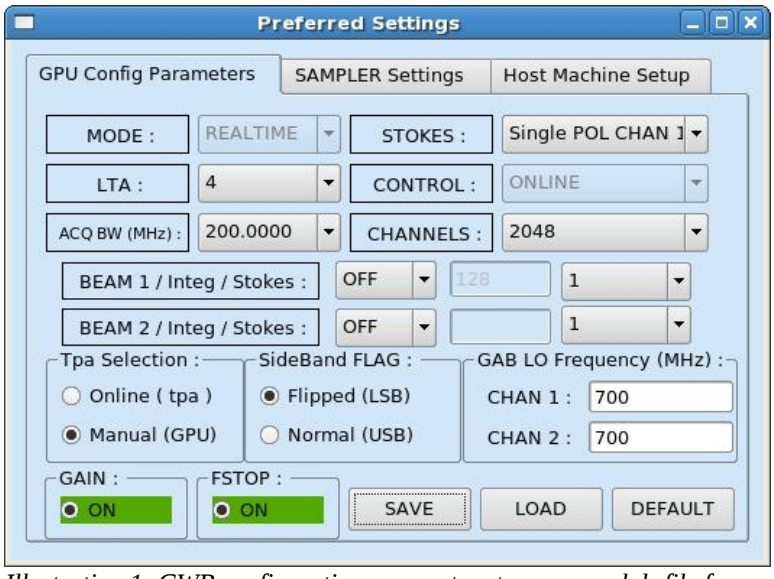


Illustration 1: GWB configuration parameters to save gpu.hdr file for Single channel USB 130 data acquisition

Below we also show corresponding `**gpu.hdr**' file, which is for GMRT observing assistants and not for general user.

gpu.hdr (sample file)

```
{ Corrsel.def
GPU_MODE    = 0              /* 0 - Realtime, 1 - RawDump          */
GPU_LTA     = 4              /* Value can be : 1,2,4,8,16,32 etc.  */
GPU_ACQ_BW  = 200.0000       /* 50, 100, 150, 200, 400 MHz. etc     */
GPU_FINAL_BW= 0              /* 0,4,8,16,32,64,128 As frac of Nyq, Val = OFF */
GPU_EDGE_FRQ= 0.000000       /* Freq Entry in steps of Nyq, Val = 0 */
GPU_CHAN_MAX= 2048           /* 512, 1024, and 2048 are possible.  */
GPU_CHAN_NUM= 0:2047:1       /* any range i:j:1;for i,j<chan_max   */
GPU_STOKES  = 1             /* 1 SinglePolUSB-130; 2 Total_Intensity; 4 Full_Stokes   */
GPU_RF      = 700  700      /* 150 235 325 610 610D 1060 1170 1280 1390 1420  */
GPU_LO-1    = 540  540      /* are in respective values of GWB RF  */
GPU_SIDEBAND= 1             /* 1 : LSB (LO > RF) and -1 : USB (LO < RF)  */
GPU_CNTRL   = 1              /* 0 -LOCAL, 1 -ONLINE, 2 -MANUAL     */
GPU_FSTOP   = 1              /* 1 - ON, 0 - OFF                    */
GPU_BEAM_1  = 0:1          /* 0-OFF,1-IA,2-PA */
GPU_BEAM_2  = 0:1          /* 0-OFF,1-PA,2-PA */
GPU_BM_INT  = 128              /* BEAM INTEGRATION (No. OF FFT CYCLES) i.e. 128
GPU_GAINEQ  = 1              /* 1 - ON, 0 - OFF                    */
GPU_TPA     = 1              /* 0 - From online tpa, 1 - From GWB GUI. */
}Corrsel
*
END_OF_HEADER                   /*  VERSION RELEASED */
```

2.1.2 Total Intensity

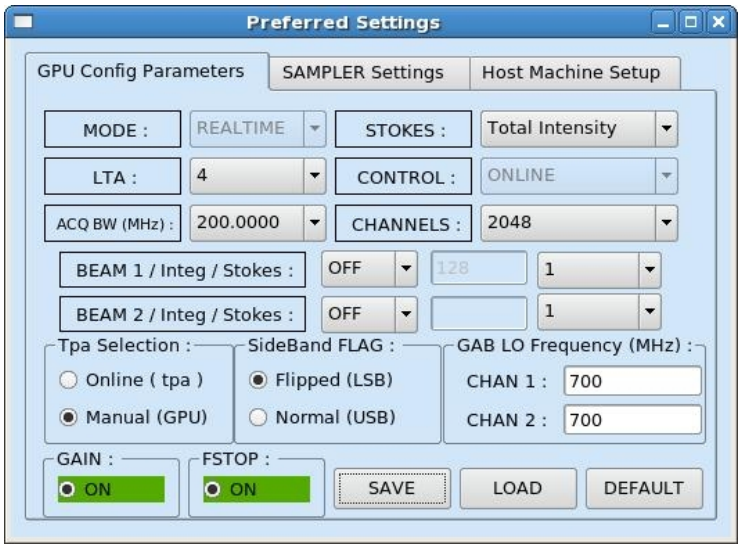


Illustration 2: GWB configuration parameters to save gpu.hdr file for Total Intensity data acquisition

Below we also show corresponding `**gpu.hdr**' file, which is for GMRT observing assistants and not for general user.

gpu.hdr (sample file)

```
{ Corrsel.def
GPU_MODE    = 0              /* 0 - Realtime, 1 - RawDump          */
GPU_LTA     = 4              /* Value can be : 1,2,4,8,16,32 etc.  */
GPU_ACQ_BW  = 200.0000       /* 50, 100, 150, 200, 400 MHz. etc     */
GPU_FINAL_BW= 0              /* 0,4,8,16,32,64,128 As frac of Nyq, Val = OFF */
GPU_EDGE_FRQ= 0.000000       /* Freq Entry in steps of Nyq, Val = 0 */
GPU_CHAN_MAX= 2048           /* 512, 1024, and 2048 are possible.  */
GPU_CHAN_NUM= 0:2047:1       /* any range i:j:1;for i,j<chan_max   */
GPU_STOKES  = 2             /* 1 SinglePolUSB-130; 2 Total_Intensity; 4 Full_Stokes   */
GPU_RF      = 700  700      /* 150 235 325 610 610D 1060 1170 1280 1390 1420  */
GPU_LO-1    = 540  540      /* are in respective values of GWB RF  */
```

```
GPU_SIDEBAND= 1              /* 1 : LSB (LO > RF) and -1 : USB (LO < RF)  */
GPU_CNTRL    = 1              /* 0 -LOCAL, 1 -ONLINE, 2 -MANUAL     */
GPU_FSTOP    = 1              /* 1 - ON, 0 - OFF                    */
GPU_BEAM_1   = 0:1           /* 0-OFF,1-IA,2-PA */
GPU_BEAM_2   = 0:1           /* 0-OFF,1-PA,2-PA */
GPU_BM_INT   = 128               /* BEAM INTEGRATION (No. OF FFT CYCLES) i.e. 128
GPU_GAINEQ   = 1              /* 1 - ON, 0 - OFF                    */
GPU_TPA      = 1              /* 0 - From online tpa, 1 - From GWB GUI. */
}Corrsel
*
END_OF_HEADER                    /*  VERSION RELEASED */
```

2.1.3 Full Stokes

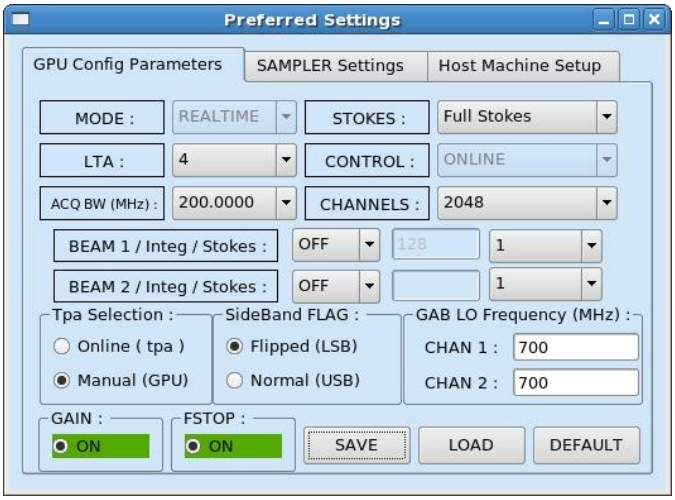


Illustration 3: GWB configuration parameters to save gpu.hdr file Full Stokes data acquisition

Below we also show corresponding `**gpu.hdr**' file, which is for GMRT observing assistants and not for general user.

gpu.hdr (sample file)

```
{ Corrsel.def
GPU_MODE     = 0              /* 0 - Realtime, 1 - RawDump          */
GPU_LTA      = 4              /* Value can be : 1,2,4,8,16,32 etc.  */
GPU_ACQ_BW   = 200.0000       /* 50, 100, 150, 200, 400 MHz. etc     */
GPU_FINAL_BW= 0               /* 0,4,8,16,32,64,128 As frac of Nyq, Val = OFF */
GPU_EDGE_FRQ= 0.000000        /* Freq Entry in steps of Nyq, Val = 0 */
GPU_CHAN_MAX= 2048            /* 512, 1024, and 2048 are possible.  */
GPU_CHAN_NUM= 0:2047:1        /* any range i:j:1;for i,j<chan_max   */
GPU_STOKES   = 4             /* 1 SinglePolUSB-130; 2 Total_Intensity; 4 Full_Stokes   */
GPU_RF       = 700  700      /* 150 235 325 610 610D 1060 1170 1280 1390 1420  */
GPU_LO-1     = 540  540      /* are in respective values of GWB RF  */
GPU_SIDEBAND= 1              /* 1 : LSB (LO > RF) and -1 : USB (LO < RF)  */
GPU_CNTRL    = 1              /* 0 -LOCAL, 1 -ONLINE, 2 -MANUAL     */
GPU_FSTOP    = 1              /* 1 - ON, 0 - OFF                    */
GPU_BEAM_1   = 0:1           /* 0-OFF,1-IA,2-PA */
GPU_BEAM_2   = 0:4           /* 0-OFF,1-PA,2-PA */
GPU_BM_INT   = 128               /* BEAM INTEGRATION (No. OF FFT CYCLES) i.e. 128
GPU_GAINEQ   = 1              /* 1 - ON, 0 - OFF                    */
GPU_TPA      = 1              /* 0 - From online tpa, 1 - From GWB GUI. */
}Corrsel
*
END_OF_HEADER                    /*  VERSION RELEASED */
```

2.2 Sampler Settings

Here we present sampler settings for all three modes available with the GWB-II discussed above; for each mode we present a screen-shot and also provide the **sample.hdr** file. User must select polarization (in case of single channel observations) and the corresponding antennas that are connected.

Default cluster size (defines the no. of ROACH boards) is set to 4, which can be reduced to 2 (for minimum configurable antenna connections with the sampler).

2.2.1 Single POL CHAN 1

Following are the sampler file and sampler file for single pol chan 1 (130 polarization) mode (Illustration 6). If user wants to acquire data from chan 2 (175 polarization) then “Channel 2” must be selected at the polarization selection box, and also configure the antenna connections list as per user requirement.

sampler.hdr (sample file)

Below we also show corresponding `**sampler.hdr**' file, which is for GMRT observing assistants and not for general user. Following is the sampler file for Single POL chan 1 (Illustration 4).

```
{ Sampler.def
*  All the even numbered are lower pipelines of the samplers.
*  All the odd  numbered are upper pipelines of the samplers.
* SamplerId  = Ant Band
*
* Top bin ( Cable starts from 0 )
* SampId = Ant BandId FftId
SMP000  = C01 USB-130 000
SMP001  = C02 USB-130 001
SMP002  = C05 USB-130 002
SMP003  = C08 USB-130 003
SMP004  = C11 USB-130 004
SMP005  = C14 USB-130 005
SMP006  = E02 USB-130 006
SMP007  = W06 USB-130 007
} Sampler
END_OF_HEADER
```

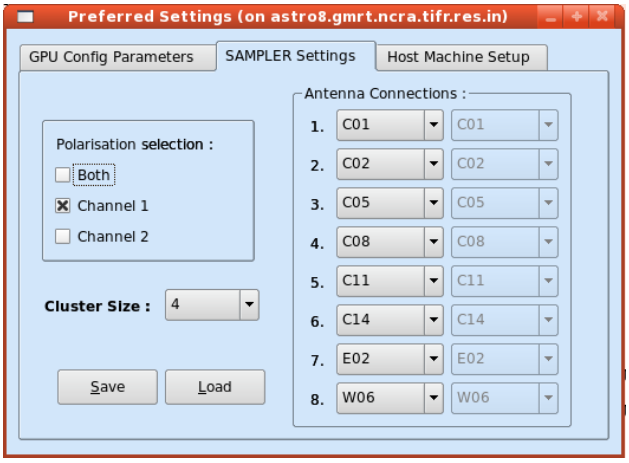


Illustration 4: antenna connections to correlator sampler for single polarization 130 mode sampler.hdr

2.2.2 Total Intensity

In this mode only co-polar data will be recorded, uses 4 antennas in dual polarization (only records of Chan1-Chan1 and Chan2-Chan2 polarizations are made). Note that this is a sub-set of Full Stokes mode discussed in Section 3.2.3 below.

sampler.hdr (sample file)

Below we also show corresponding `**sampler.hdr**' file, which is for GMRT observing assistants and not for general user. Following is the sampler file for Dual polarization Total Intensity (Illustration 5).

```
{ Sampler.def
*  All the even numbered are lower pipelines of the samplers.
*  All the odd  numbered are upper pipelines of the samplers.
* SamplerId  = Ant Band
*
* Top bin ( Cable starts from 0 )
* SampId = Ant BandId FftId
```

```
SMP000   = S04 USB-130 000
SMP001   = S04 USB-175 001
SMP002   = S06 USB-130 002
SMP003   = S06 USB-175 003
SMP004   = W01 USB-130 004
SMP005   = W01 USB-175 005
SMP006   = W06 USB-130 006
SMP007   = W06 USB-175 007
} Sampler
END_OF_HEADER
```

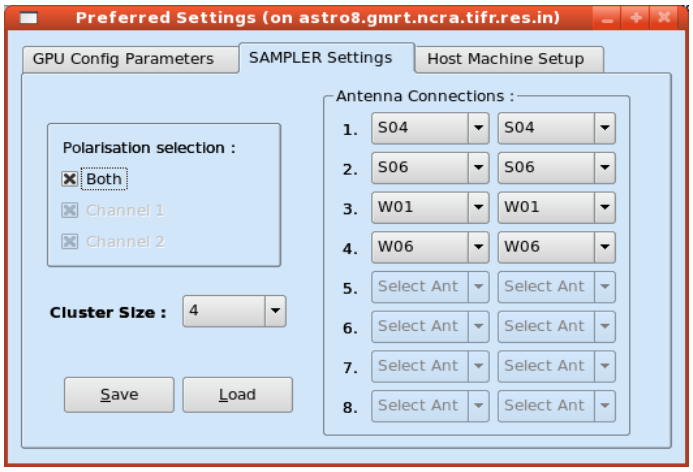


Illustration 5: Antenna connections to correlator sampler for dual polarization sampler.hdr

2.2.3 Full Stokes

In this mode the correlator records both the co-polar and the cross-polar data. 4 antenna dual polarization (Full Stokes mode records all products of polarizations).

sampler.hdr (sample file)

Below we also show corresponding `**sampler.hdr**' file, which is for GMRT observing assistants and not for general user. Following is the sampler file for Dual polarization with Full Stokes mode (Illustration 6).

```
{ Sampler.def
*  All the even numbered are lower pipelines of the samplers.
*  All the odd  numbered are upper pipelines of the samplers.
* SamplerId  = Ant Band
*
* Top bin ( Cable starts from 0 )
* SampId = Ant BandId FftId
SMP000   = S04 USB-130 000
SMP001   = S04 USB-175 001
SMP002   = S06 USB-130 002
SMP003   = S06 USB-175 003
SMP004   = W01 USB-130 004
SMP005   = W01 USB-175 005
SMP006   = W06 USB-130 006
SMP007   = W06 USB-175 007
} Sampler
END_OF_HEADER
```

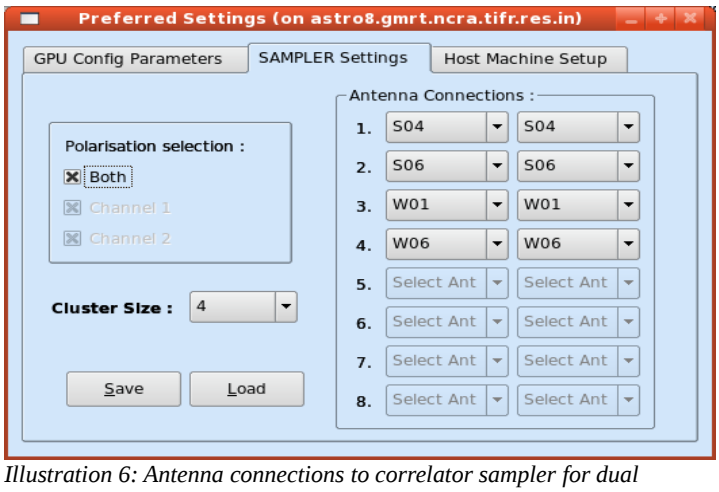


Illustration 6: Antenna connections to correlator sampler for dual polarization sampler.hdr

2.3 Hosts Machine Setup

Below we also show corresponding `**hosts.dat**' file, which is for GMRT observing assistants and not for general user. Host machines for online, correlator (both interferometry and pulsar's) are set.

hosts.dat (sample file)

```
lenyadri 192.168.1.13    6001  ComSock4
node52    192.168.11.16   6002  LogSock4
node54    192.168.11.18   6002  LogSock4
node53    192.168.11.17   6002  LogSock4
```

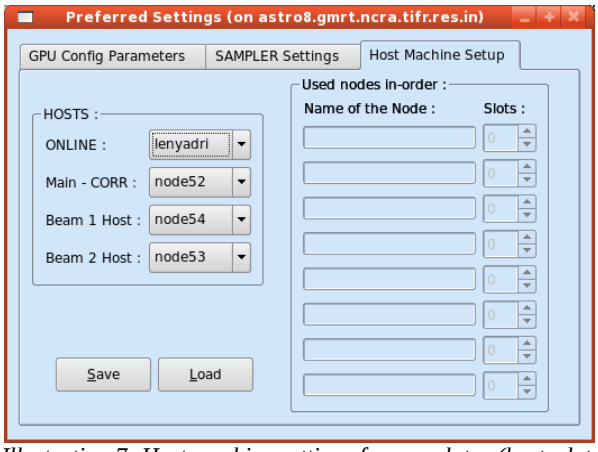


Illustration 7: Host machine settings for correlator (hosts.dat & host.list)

NOTE : host.list file modification through gui (used nodes in order) is disabled.

2.4 Steps to Run GWB Correlator and Applications

1. logon to **observer@astro8** and enter commands as:
   cd ~/bin/gpucorr/ver3/
   ./gpucorr-node52 or **./gwbcorr** (for node52).
   This will open a qt interface for gwb correlator (gwb-dasconsole).
   NOTE : files *gpu.hdr, sampler.hdr, sampler_1pol.hdr, sampler_dual.hdr, default.hdr, hosts.dat, host.list,* must be available in the *'~/bin/gpucorr/ver3'* directory.

2. On Menubar go to "**Start -> Open All Windows"** or "**Ctrl+O"** or go to "**Start -> GWB - Windows -> Sockcmd "** to open all client processes to run gwb with sockcmd and online dassrv processes, or "**Start -> GWB - Windows -> Getcmd "** to run gwb without sockcmd and dassrv processes.

   This will popup the client workspaces for each command with following order:
   - "**node52::gwb_corr_released.sh**":
     It can also be termed as acquisition client. This starts and broadcasts the acquisition processes to the gpunode6, gpunode5, gpunode4, gpunode3.
   - "**node52::sockcmd.sh**": (can be started only when GWB running with sockcmd mode)
     This sets up the communication between online and correlator and gives acknowledgments to the commands from the correlator to the online and vice versa.
   - "**node52::collect.sh**":
     This dumps the Astronomical data into the buffer and keeps it there for a while and removes it as per the FIFO logic.
   - "**node52::record**":
     one can write the acquired data into specified lta format file as per requirement.

3. On Menubar go to "**Edit -> Preferences"** (or use accelerator "**CTRL+P"**).
   This opens the tab-widget to set the parameters for gpu.hdr, hosts.dat, sampler_1pol.hdr and sampler_dual.hdr files. On the tab-widget there are three tabs named
   1. Configuring GWB Parameters:
      Using this user can set various available GWB modes, and related parameters, this is depend on the user's choice or requirement, what kind of data he needs. (e.g. Interferometry-continuum, Interferometry-line, and Interferometry-Pulsar etc).

   2. configuration of SAMPLER settings:

This tab Will be re-loaded automatically as one saves the gpu.hdr file from first tab (i.e. After configuring required GWB parameters), and/or previously saved parameters from either sampler_1pol.dat or sampler_dual.hdr files. One can change this as per requirement and connections made at the GWB ROACH-Boards inputs. **Currently, the cluster size made default to 4 irrespective of the GUI selection**. i.e. GUI will always save the sampler file for 4node cluster size. So that, user can record **8 antennas single pol arization or 4 antenna dual polarization (either Total Intensity which records only RR and LL polarizations and Full Stokes which records RR RL LR and LL polarizations).**

3. Host Machine Setup:
   In this User will be asked to set the Host machines used for the GWB (GPU) correlator. Please, do not bother about the disabled GUI part in this tab.

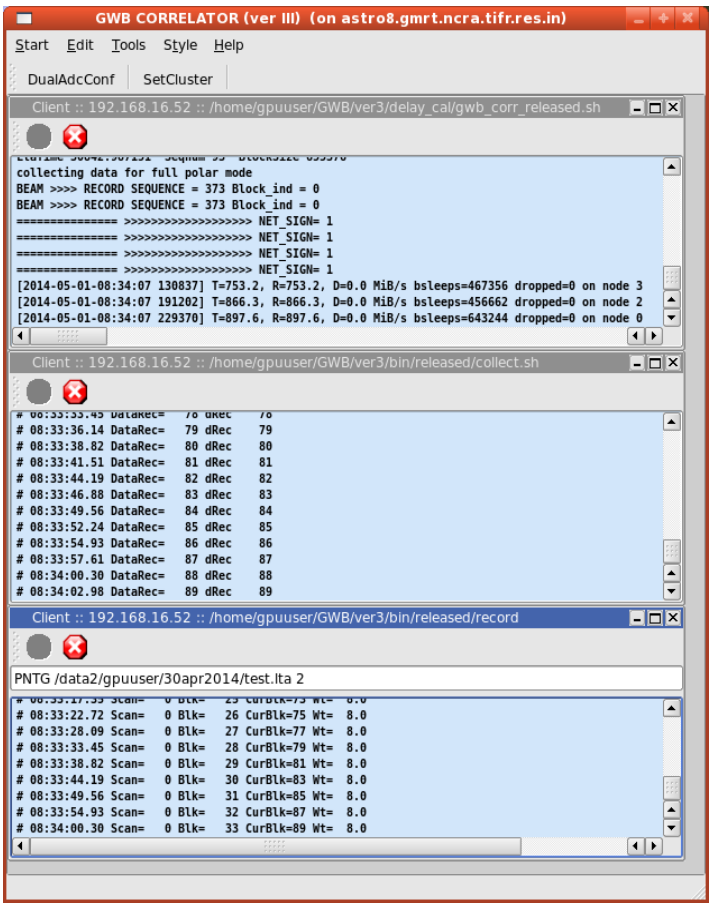
4. Configure **Dual ADC config**:
   From MenuBar select **"Start ->DualAdcConfig**" or With a single click on **DualAdcConfig** (visible on Toolbar) button one can configure this. This sets the GWB Roach boards in programmed mode, It takes about 10 seconds to configure. During That time GUI will be frozen intentionally for user interactions.
5. 

Illustration 8: GWB Data acquisition Console

6. Now, click the **start button** (blue icon button) of first client window named "gwb_corr_released.sh" and wait till it says:
   gmrt_correlator : Waiting For Initialization Cmd ..

7. Start sockcmd.sh, collect, and dass-srv (from online machine) for gwb.
   ssh -X observer@shivneri
   cd /odisk/online1/gsbe/dassrv-gpu/
   ./dassrv_released
   In Case, if you are running GWB, in **"getcmd"** mode, then there is no need to start both

sockcmd.sh and dassrv-gpu processes, user just have to start the process “collect” only.

8. Enter initndas command from online user0.

```
allant;cmode 1; tpa(11)=15; initndas'/temp2/data/gsb.hdr' **(GWB ONLY).
allant;cmode 9; tpa(11)=15; initndas'/temp2/data/gsb.hdr' **(GSB + GWB).
```

For getcmd mode :

On the online machine terminal, enter command

```
gwbcmd initndas
```

9. Setup the cluster acquisition :
On Menubar go to “**Start**-> **SetCluster”** or With a single click on **setCluster** (visible on Toolbar) one can setup this. This takes about 2 to 3 seconds to set.

10. Start project from online machine...
For Sockcmd mode :

```
allant;subar 4; prjobs'';prjtit'';initprj(1,'GWBTST')  **(GWB ONLY).
allant;subar 4; prjobs'';prjtit'';initprj(15,'GWBTST') **(GSB + GWB).
```

For getcmd mode :

On online machine terminal, enter command

```
gwbcmd initprj                          ;for subar 4.
gwbcmd initprj <subar number>           ;for multi-sub-array observation.
```

11. Start and stop scan as per requirement and one can start record for the same.
To record the data in record window type in the format as :
GWBTST /data2/gpuuser/gwbtst_26jan2014.lta
GWBTST /data2/gpuuser/gwbtst_26jan2014.lta 4

For GWB with Sockcmd mode default commands strtndas and stpndas from online user / subar controller (command file) can be used.

```
Recording (start and stop scans)  using getcmd :
            gwbcmd strtndas                     ;for subar 4.
            gwbcmd stpndas
            gwbcmd strtndas <subar number>    ;formulti-sub-array observation.
            gwbcmd stpndas <subar number>
```

12. Starting **DASMON** :
Dasmon is released, and can be started with following command :
login to node52 : **ssh -X gpuuser@node52**
enter commands as : **/home/gpuuser/GWB/ver3/bin/released/dasmon.pl**

Also, DasMon Can be Started from the main DasConsole GUI from
“**MenuBar->Tools->Interferometry->GWB DasMon” or *CTRL* + *M*** as an accelerator.

13. Starting Power Equalisation Program :
GWB Power Equalise GUI Can be Started from the main DasConsole GUI from
“**MenuBar->Tools->Interferometry->GWB_PowerEq” or CTRL + E** as an accelerator. This can also be done as explained in later section 'GAB - GWB Power Equalise'.

14. Starting IA beam Observations :
GWB Incoherent Array Pulsar Dasconsole programs can be started as :
“**MenuBar->Tools->Pulsar Tools->Pulsar DasConsole” or ALT + B** as an accelerator. For more information on this please refer section 'Starting Incoherent Pulsar Observation' which is explained here later.

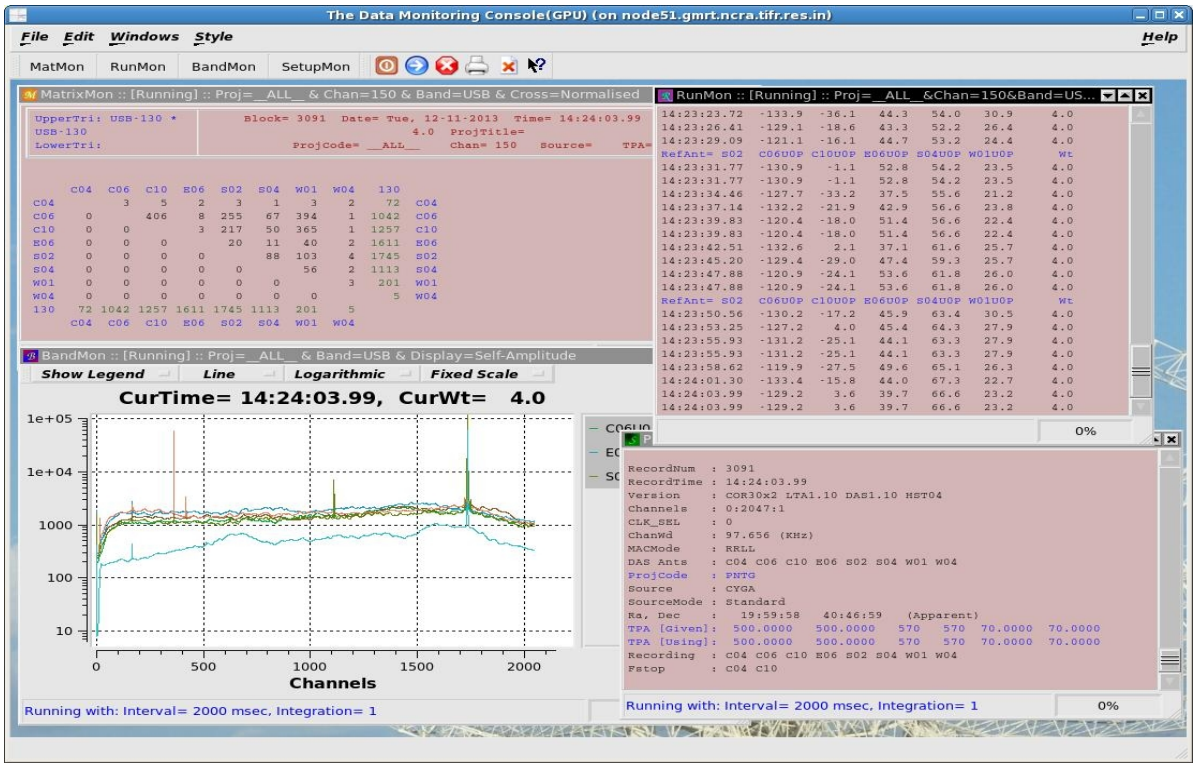


Illustration 9: Dasmon Showing matmon, bandmon, parmon and setupmon

15. Running **<u>Offline Data analysis programs</u>** :
    1. ltahdr , listscan, gvfits, dasmon tools are released for further analysis.
    2. tax, xtract, rantsol are not yet released for this in order to use these tool please copy the file to astro0 or any other NIS machines where it works.
       For some cases user has to set fmt as :***fmt = ist%10.5f;base{chan{a%31.4f;p%8.1f}};\n***

2.5 GAB Power Equalise

Power Equalise program is released for GWB, which uses the output self visibility data from GWB and equalizes the power levels at GAB (GMRT Analog Backend) system.

Steps to follow (with GUI)

GWB Power Equalise GUI Can be Started from the main DasConsole GUI from
"**MenuBar->Tools->GWB_PowerEq" or CTRL + E** as an accelerator (as explained earlier in GWB-CORRELATOR).
Also, The same can be started manually as follows:

1. Log on to gpuuser@192.168.16.52 (node52 = 192.168.16.52).
2. Enter command as : **/home/gpuuser/GWB/ver3/bin/released/gwbpeq**
3. Select the antennas connected to the GAB and GWB.
4. Initially make all GAB attenuation's same for both the polarization's, as set in the cdsetX file, or change Attenuation's to MAXIMUM, by clicking on button 'MAX Attn', apply it to GAB accordingly.
5. Set the Optimum level, Begining channel, End channel, Upper level, Lower level and Integrations as per requirement.
6. Click on the button save to generate text files as per selected gui options.
7. Click the button 'EQUALISE' to start first iteration.

8. Run the process 'run gwblev' from userX window from online(2-3 times).
9. Repeat steps 5 and 6 till optimum level is attained.

Steps to Power Equalisation (without GUI)

1. **ssh -X gpuuser@node52**
2. Edit the file ***/home/gpuuser/GWB/PowerEq/gab_attn.cur*** set the attenuation values for both polarisation to values which are already set from cdsetX file for GAB.
3. Run the following command
   power_eq -o 100 -c 1 -u 100000 -l 0 -b 100 -e 20000 -i 2 -m shivneri
   Or
   power_eq -o 100 -c 1 -u 100000 -l 0 -b 100 -e 20000 -i 2 -m lenyadri
   *Note that -**h** option for above program will print help regarding the options used by it.*
   This will generate the new attenuation values for the selected or required antennas and modifies the above **gab_attn.cur** file (Content of the file will be as given below).
4. After this, From Online userX please run "run gwblev" 2-3 times.
5. Repeat steps 3 and 4 till the optimum power levels at GWB output are achieved.

Content of *gab_attn.cur*

```
#ANT SEL ATTN130 ATTN175
C00 0 8   8
C01 0 8   8
C02 0 8   8
C03 0 8   8
C04 1 14   14
C05 0 8   8
C06 1 14   14
C08 1 14   14
C09 0 8   8
C10 0 8   8
C11 0 8   8
C12 0 8   8
C13 0 8   8
C14 0 8   8
S01 0 8   8
S02 1 14   14
S03 0 8   8
S04 1 14   14
S06 0 8   8
E02 0 8   8
E03 0 8   8
E04 0 8   8
E05 0 8   8
E06 1 14   14
W01 0 8   8
W02 0 8   8
W03 0 8   8
W04 1 14   14
W05 0 8   8
W06 1 14   14
```

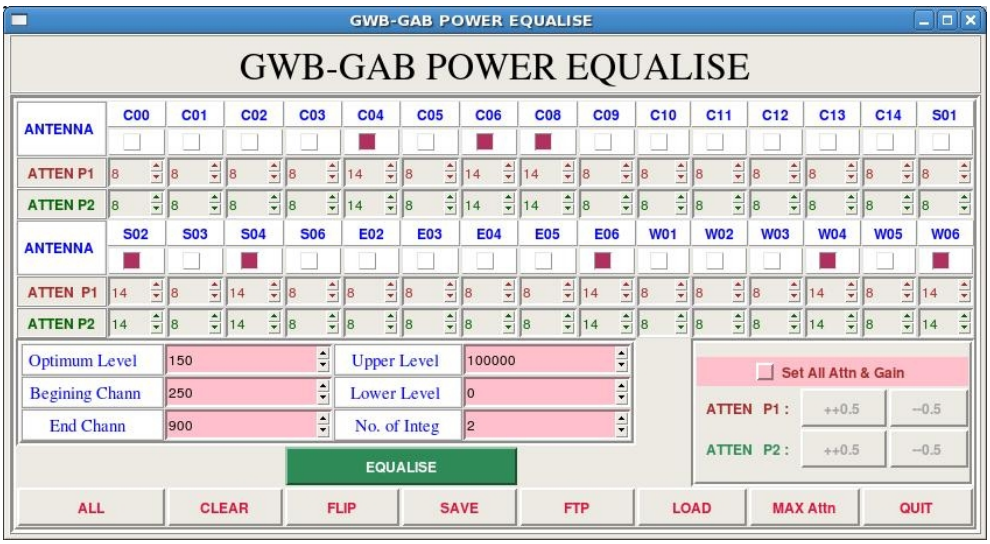


Illustration 10: GWB-GAB power Equalise Window

Chapter 3: Running Pulsar Observations

Below we first list the GWB configuration parameters and sampler settings for all three modes of GWB-II, and hosts machine setup in order to start GWB Pulsar acquisition. This part will be similar to as explained int the section 2.1.

3.1 GWB Config Parameters

Below gives the parameter lists, which are saved/changed from the GWB Config window.

```
MODE            :       Sets the GWB operating mode (default is REALTIME, and selection is disabled
                        from GUI).
LTA             :       Minimum data acquisition rate for GWB, can be set to 1, 2, 4, 8, 16 and 32.
ACQ BW          :       selectable from the available bandwidth modes, viz. 50, 100, 150, 200, 400
                              MHz.
CHANNELS        :       User selectable range of channels available for the given GWB ACQ BW.
STOKES          :       STOKES parameter selections (Single POL USB-130, Total Intensity, Full
                        Stokes).
CONTROL         :       GWB control mode, either ONLINE or LOCAL/Manual. (default :ONLINE, selection
                        is disabled from GUI).
TPA Frequency   :
                    1)  Online : This will take TPA parameters from online machine, and disables the
                        Sideband Flag and  GAB LO entries at GUI.
                    2)  Local : This enables user to choose sideband, and  GAB LO Entries.
                        1.  Flipped (LSB- decreasing frequency over channels, RF < LO).
                        2.  Normal (USB – increasing frequency over channels, RF > LO).
GAIN            :       Default is ON.
FSTOP           :       Default is ON.
Beam1/Beam2     :       It can be set as IA(0) or PA(1) with beam stokes as 1 or 4.
Beam Integ      :       It can be set as 128 or 64 FFTs which corresponds to 1.3 ms or 0.65 ms
```

Now we present all three modes available with the GWB-II; for each mode we present a screen-shot and also provide the sample **gpu.hdr** file. Note that GWB (GPU) RF frequency is the same as GAB LO frequency.

3.1.1 Single POL CHAN 1

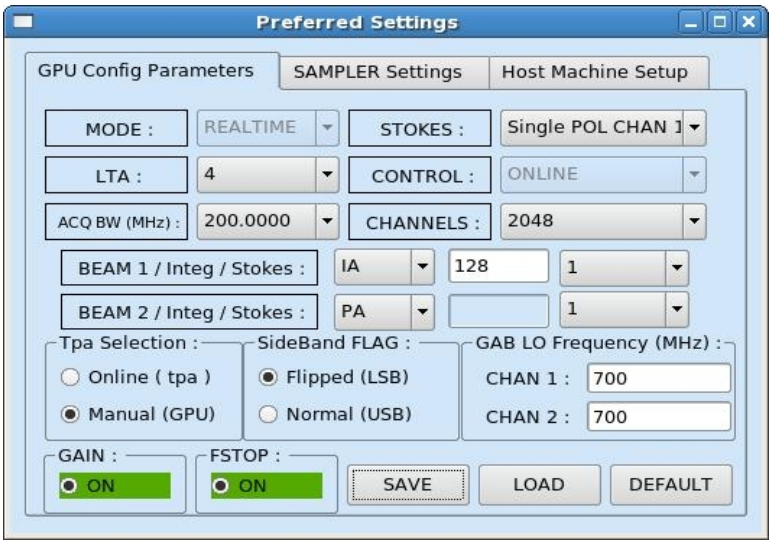


Illustration 11: GWB configuration parameters to save gpu.hdr file for Single channel USB 130 data acquisition

Below we also show corresponding `**gpu.hdr**' file, which is for GMRT observing assistants and not for general user.

gpu.hdr (sample file)

```
{ Corrsel.def
GPU_MODE    = 0              /* 0 - Realtime, 1 - RawDump          */
GPU_LTA     = 4              /* Value can be : 1,2,4,8,16,32 etc.  */
GPU_ACQ_BW  = 200.0000       /* 50, 100, 150, 200, 400 MHz. etc     */
GPU_FINAL_BW= 0              /* 0,4,8,16,32,64,128 As frac of Nyq, Val = OFF */
GPU_EDGE_FRQ= 0.000000       /* Freq Entry in steps of Nyq, Val = 0 */
GPU_CHAN_MAX= 2048           /* 512, 1024, and 2048 are possible.  */
GPU_CHAN_NUM= 0:2047:1       /* any range i:j:1;for i,j<chan_max   */
GPU_STOKES  = 1             /* 1 SinglePolUSB-130; 2 Total_Intensity; 4 Full_Stokes   */
GPU_RF      = 700  700      /* 150 235 325 610 610D 1060 1170 1280 1390 1420  */
GPU_LO-1    = 540  540      /* are in respective values of GWB RF  */
GPU_SIDEBAND= 1             /* 1 : LSB (LO > RF) and -1 : USB (LO < RF)  */
GPU_CNTRL   = 1              /* 0 -LOCAL, 1 -ONLINE, 2 -MANUAL     */
GPU_FSTOP   = 1              /* 1 - ON, 0 - OFF                    */
GPU_BEAM_1  = 1:1          /* 0-OFF,1-IA,2-PA */
GPU_BEAM_2  = 2:1          /* 0-OFF,1-PA,2-PA */
GPU_BM_INT  = 128              /* BEAM INTEGRATION (No. OF FFT CYCLES) i.e. 128
GPU_GAINEQ  = 1              /* 1 - ON, 0 - OFF                    */
GPU_TPA     = 1              /* 0 - From online tpa, 1 - From GWB GUI. */
}Corrsel
*
END_OF_HEADER                   /*  VERSION RELEASED */
```

3.1.2 Total Intensity

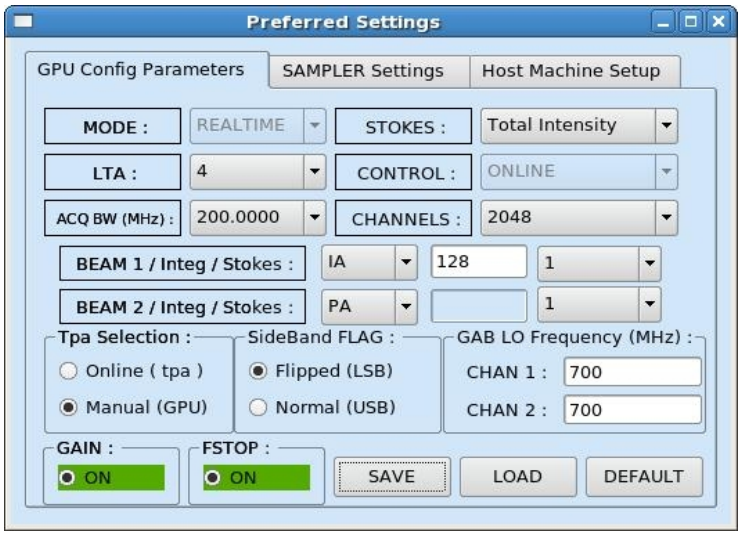


Illustration 12: GWB configuration parameters to save gpu.hdr file for Total Intensity data acquisition

Below we also show corresponding `**gpu.hdr**' file, which is for GMRT observing assistants and not for general user.

gpu.hdr (sample file)

```
{ Corrsel.def
GPU_MODE    = 0              /* 0 - Realtime, 1 - RawDump          */
GPU_LTA     = 4              /* Value can be : 1,2,4,8,16,32 etc.  */
GPU_ACQ_BW  = 200.0000       /* 50, 100, 150, 200, 400 MHz. etc     */
GPU_FINAL_BW= 0              /* 0,4,8,16,32,64,128 As frac of Nyq, Val = OFF */
GPU_EDGE_FRQ= 0.000000       /* Freq Entry in steps of Nyq, Val = 0 */
GPU_CHAN_MAX= 2048           /* 512, 1024, and 2048 are possible.  */
GPU_CHAN_NUM= 0:2047:1       /* any range i:j:1;for i,j<chan_max   */
GPU_STOKES  = 2             /* 1 SinglePolUSB-130; 2 Total_Intensity; 4 Full_Stokes   */
GPU_RF      = 700  700      /* 150 235 325 610 610D 1060 1170 1280 1390 1420  */
GPU_LO-1    = 540  540      /* are in respective values of GWB RF  */
GPU_SIDEBAND= 1             /* 1 : LSB (LO > RF) and -1 : USB (LO < RF)  */
GPU_CNTRL   = 1              /* 0 -LOCAL, 1 -ONLINE, 2 -MANUAL     */
```

```
GPU_FSTOP    = 1              /* 1 - ON, 0 - OFF                    */
GPU_BEAM_1   = 1:1          /* 0-OFF,1-IA,2-PA */
GPU_BEAM_2   = 2:1          /* 0-OFF,1-PA,2-PA */
GPU_BM_INT   = 128              /* BEAM INTEGRATION (No. OF FFT CYCLES) i.e. 128
GPU_GAINEQ   = 1              /* 1 - ON, 0 - OFF                    */
GPU_TPA      = 1              /* 0 - From online tpa, 1 - From GWB GUI. */
}Corrsel
*
END_OF_HEADER                   /*  VERSION RELEASED */
```

3.1.3 Full Stokes

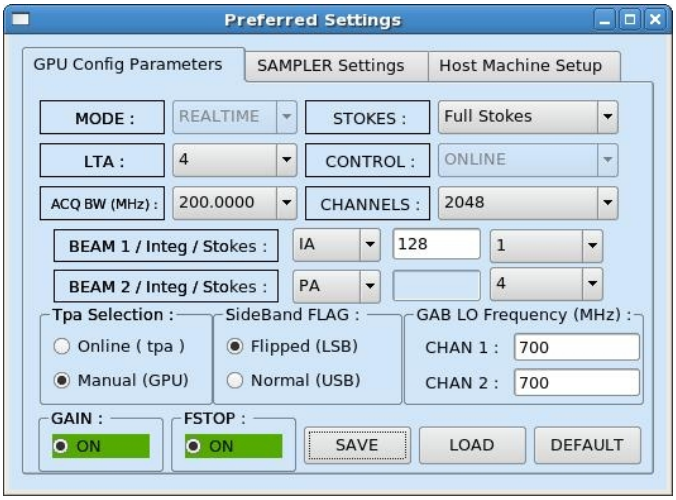


Illustration 13: GWB configuration parameters to save gpu.hdr file Full Stokes data acquisition

Below we also show corresponding `**gpu.hdr**' file, which is for GMRT observing assistants and not for general user.

gpu.hdr (sample file)

```
{ Corrsel.def
GPU_MODE     = 0              /* 0 - Realtime, 1 - RawDump          */
GPU_LTA      = 4              /* Value can be : 1,2,4,8,16,32 etc.  */
GPU_ACQ_BW   = 200.0000       /* 50, 100, 150, 200, 400 MHz. etc     */
GPU_FINAL_BW= 0               /* 0,4,8,16,32,64,128 As frac of Nyq, Val = OFF */
GPU_EDGE_FRQ= 0.000000        /* Freq Entry in steps of Nyq, Val = 0 */
GPU_CHAN_MAX= 2048            /* 512, 1024, and 2048 are possible.  */
GPU_CHAN_NUM= 0:2047:1        /* any range i:j:1;for i,j<chan_max   */
GPU_STOKES   = 4             /* 1 SinglePolUSB-130; 2 Total_Intensity; 4 Full_Stokes   */
GPU_RF       = 700  700      /* 150 235 325 610 610D 1060 1170 1280 1390 1420  */
GPU_LO-1     = 540  540      /* are in respective values of GWB RF  */
GPU_SIDEBAND= 1              /* 1 : LSB (LO > RF) and -1 : USB (LO < RF)  */
GPU_CNTRL    = 1              /* 0 -LOCAL, 1 -ONLINE, 2 -MANUAL     */
GPU_FSTOP    = 1              /* 1 - ON, 0 - OFF                    */
GPU_BEAM_1   = 1:1          /* 0-OFF,1-IA,2-PA */
GPU_BEAM_2   = 2:4          /* 0-OFF,1-PA,2-PA */
GPU_BM_INT   = 128              /* BEAM INTEGRATION (No. OF FFT CYCLES) i.e. 128
GPU_GAINEQ   = 1              /* 1 - ON, 0 - OFF                    */
GPU_TPA      = 1              /* 0 - From online tpa, 1 - From GWB GUI. */
}Corrsel
*
END_OF_HEADER                   /*  VERSION RELEASED */
```

3.2 Sampler Settings

Please refer **Section 2.2** to configure the antenna connections on SAMPLER ROACH Boards.

3.3 Hosts Machine Setup

Please refer **Section 2.3** to configure the ONLINE HOST, GWB CORR HOST and BEAM DATA HOSTS.

3.4 Steps to Run GWB Correlator and Applications

Please refer **Section 2.4** for steps to start GWB correlator.

3.5 GAB Power Equalise

This Part is already explained in **Section 2.5.**

3.6 Phasing the GWB data

This can be invoked from GWB-CORRELATOR Main Window from " **Tools -> Pulsar Tools -> GWB Phasing**", or pressing **Alt+P** as an accelerator.

This utility temporarily provided with small tool which calls the phase_gwb.pl from online machine. Phasing Widget allows to choose the following :

- **Reference Antenna** Name for selected sub-array.
- **Sub-array** Number for which to carry phasing iteration.
- **Data recording Time** on which Phasing will work for the solutions.
- **Project Code** to be entered for related subarray which is used.

Note : Antenna selection Button is provided, but code for 'antsel' is not yet ready.

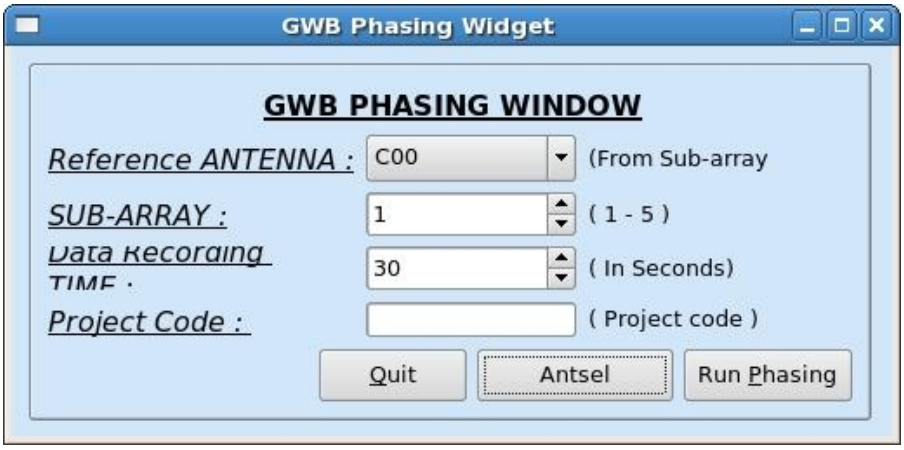


Illustration 14: Phasing widget to run and apply phasing iteration on GWB.

3.7 Pulsar Data Acquisition

Now we give the rest of the steps to start GWB-II pulsar acquisition.

1. logon to **observer@astro8** and enter commands as :
   cd ~/bin/gpucorr/ver3/
   ./gpu_psr or **./gwb-psr_das** (for node52).
   This will open a qt interface for gwb correlator (gwb-dasconsole).
   NOTE : files gpu.hdr, sampler.hdr, sampler_1pol.hdr, sampler_dual.hdr, default.hdr, hosts.dat, host.list, must be available in the '~/bin/gpucorr/ver3' directory.

OR

This can Also be Started from the main GPU-CORRELATOR Window as Explained above in earlier section GWB-CORRELATOR .

The GUI has two workspaces. First for Beam 1 and the other for beam 2. In each workspaces , client windows can be opened from start menu.

2. On Menubar go to "**Start -> All Windows"** or "**Ctrl+N"** or go to "**Start -> BEAM1 - Windows -> All "** to open all client processes to run gwb incoherent array pulsar mode processes on node54(192.168.16.54).

   This will popup the client processing windows for Beam 1 host machine (set from the Preferences of the Main DasConsole GUI), in the following order :

   1. "**node54::bm1_process_psr**":
      It can also be termed as incoherent array pulsar data acquisition and processing client.
   2. "**node54::collect_psr**":
      This dumps the incoherent array pulsar data into the Shared memory.
   3. "**node54::bm1_record_psr**":
      one can write the acquired incoherent pulsar data into specified .raw format file as per requirement.

3. Start the clients processes listed in 2.1 and 2.2 (bm1_process_psr and collect_psr) by pressing Blue (start) button on the Client windows.

4. On the ToolBar There are Four Different Buttons viz., InitBm1, StartBm1, StopBm1, FinishBm1, etc.

5. On Menubar go to "**Start -> All Windows"** or "**Ctrl+N"** or go to "**Start -> BEAM2 - Windows -> All "** to open all client processes to run gwb coherent array pulsar mode processes on node53(192.168.16.53).

   This will popup the client processing windows for Beam 2 host machine (set from the Preferences of the Main DasConsole GUI), in the following order :

   1. "**node53::bm2_process_psr**":
      It can also be termed as incoherent array pulsar data acquisition and processing client.
   2. "**node53::collect_psr**":
      This dumps the incoherent array pulsar data into the Shared memory.
   3. "**node53::bm2_record_psr**":

   one can write the acquired incoherent pulsar data into specified .raw format file as per requirement.

6. Start the clients processes listed in 2.1 and 2.2 (bm1_process_psr and collect_psr) by pressing Blue (start) button on the Client windows.

7. On the ToolBar There are Four Different Buttons viz., InitBm2, StartBm2, StopBm2, FinishBm2, etc.

8. In addition to this, There are Buttons to control data for pulsar beams which are named by InitBoth, StartBoth, StopBoth, FinishBoth. These four buttons will control the process in simultaneously, If user is working with the Both Beam data.

   1. **InitBm1/InitBm2/InitBoth :**
      Initializes the beam1 and beam2 Process Pulsar Beam Acquistion.
   2. **StartBm1/StartBm2/StartBoth :**
      Starts the pulsar DATA acquisition for beam1 and beam2 collect pulsar.
   3. **StopBm1/StopBm2/StopBoth :**
      Stops the pulsar DATA acquisition for beam1 and beam2 collect pulsar.
   4. **FinishBm1/FinishBm2/FinishBoth :**
      Halts the beam1 and beam2 Processes Pulsar Beam acquisition.

Note : All The above pulsar command process execution under Toolbar Buttons, can also be done from online machine terminal with commands as :

1. ***gwb_bm1.finish* / gwb_bm2.finish****
2. ***gwb_bm1.init* / gwb_bm2.init****
3. ***gwb_bm1.start* / gwb_bm2.start****
4. **gwb_bm1.stop* / gwb_bm2.stop***

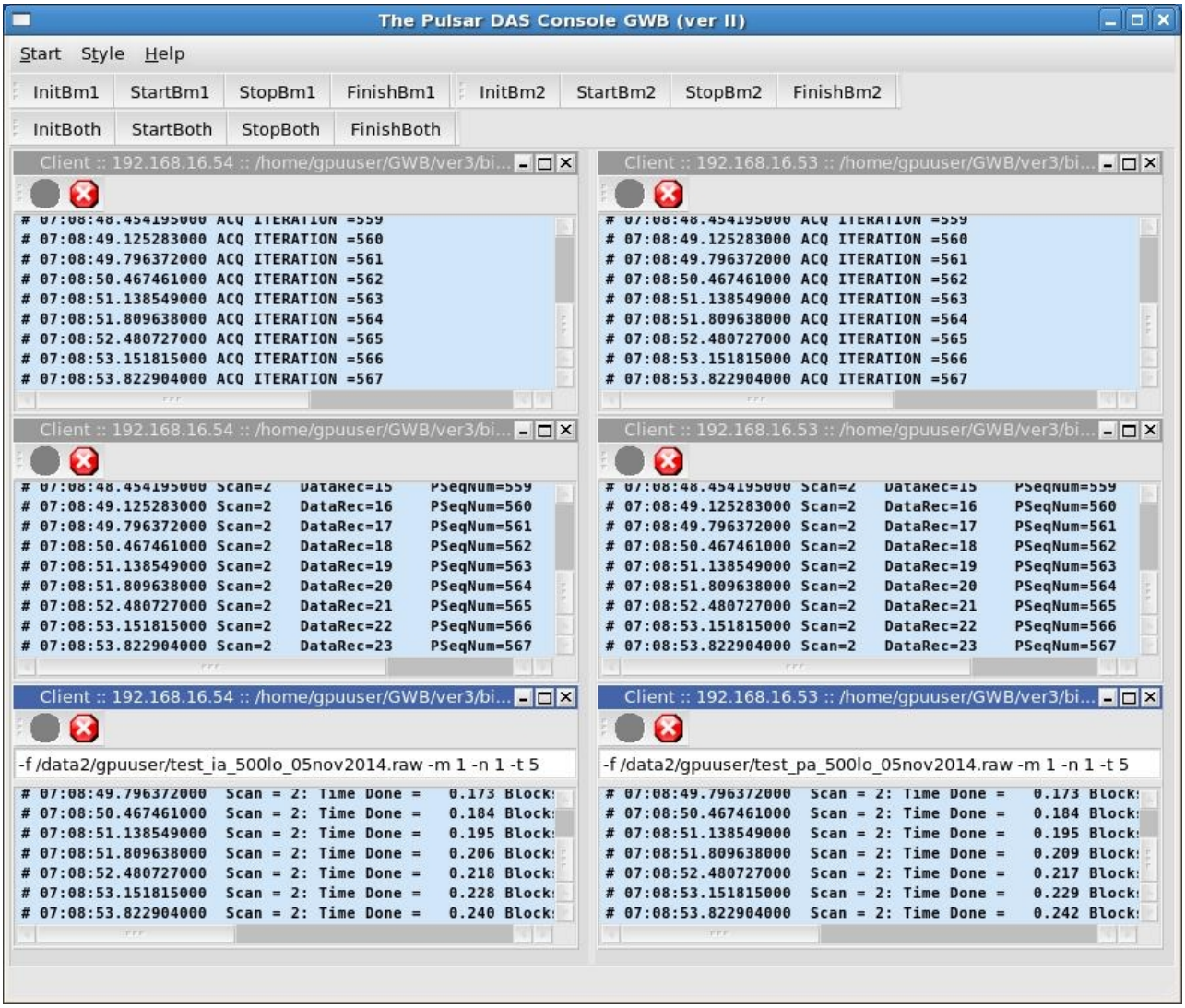


Illustration 15: Pulsar Data acquisition Interface (Pulsar DAS) showing IA (beam1) and PA (beam2) processes.

Chapter 5: Troubleshooting

Some Quick Checks

If acquisition program fails to run then check for the following :

1. Machines required to run gpu cluster are ON.
2. Check for the Processes DualAdcConfig and SetCluster are getting executed sucessfully, If not then **GWB ROACH-BOARDS may not be communicating / hanged / not in sync** with each other. Sometimes extension of this can be observed in acquisition program **"gwb_corr_released.sh"**
3. Check for the processes , shared memory segment which are not closed properly. According clear those processes and shared memory segments,using following commands on gpu node52
   a. */home/gpuuser/GWB/ver3/bin/released/clear_all_node_shm.csh* // for shm
   b. */home/gpuuser/GWB/ver3/bin/released/kill_all_nodes.csh* // for orte-clean
4. Check for background mpi processes and clear the same.

IMPORTANT Notes

i. GWB (GPU) can be run in parallel with GSB.
   1. Using above mentioned procedure. i.e. Using sockcmd.sh and dassrv-gpu processes.
   2. Using getcmd mode, while GSB is already running. No need to start sockcmd.sh and dassrv-gpu processes.
   3. Options to start client windows with sockcmd.sh and without sockcmd.sh are available in "Start->Gwb -Windows" options. And use gpucmd commands from online machine with arguments as initndas, initprj, strtndas, stpndas, etc.
   4. If no subar number is provided for gpucmd command then it will be executed for subar 4**.**

ii. Dasmon is released, and can be started with following command :
   ssh -X gpuuser@node52 -f dasmon

Antenna connections to GWB Roach boards

Cable No:	GWB Node No
CHB37	NODE06
CHB38	NODE06
CHB39	NODE05
CHB51	NODE05
CHB53	NODE04
CHB55	NODE04
SPR3	NODE03
CHB59	NODE03

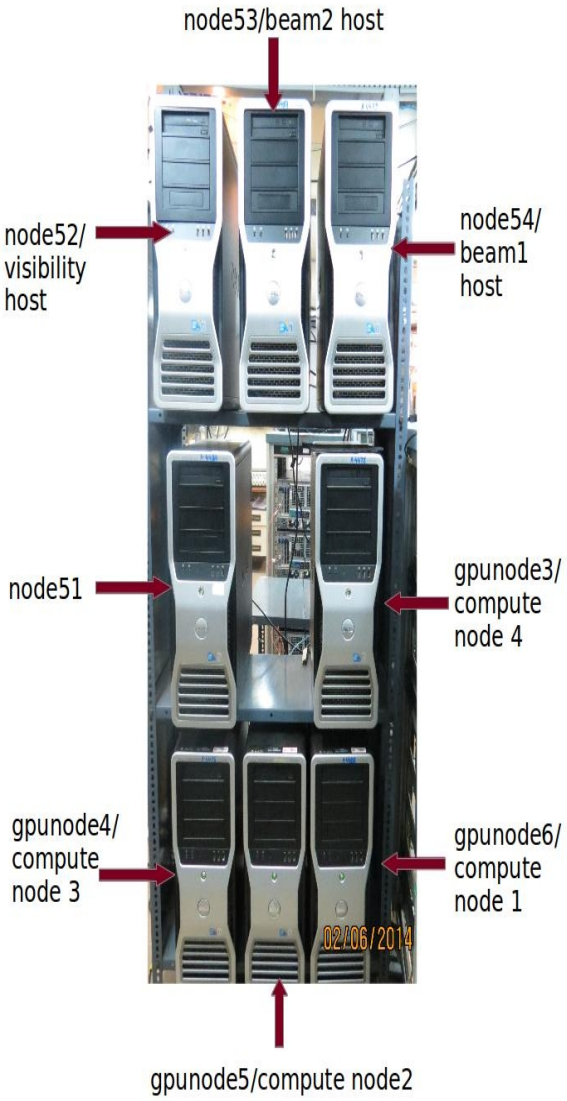
In case of dual polarization observations please connect the first 4 cables to first channel (polarization 1) and next 4 to another channel(polarization 2).

Appendix - 1 POWER ON/OFF PROCEDURE

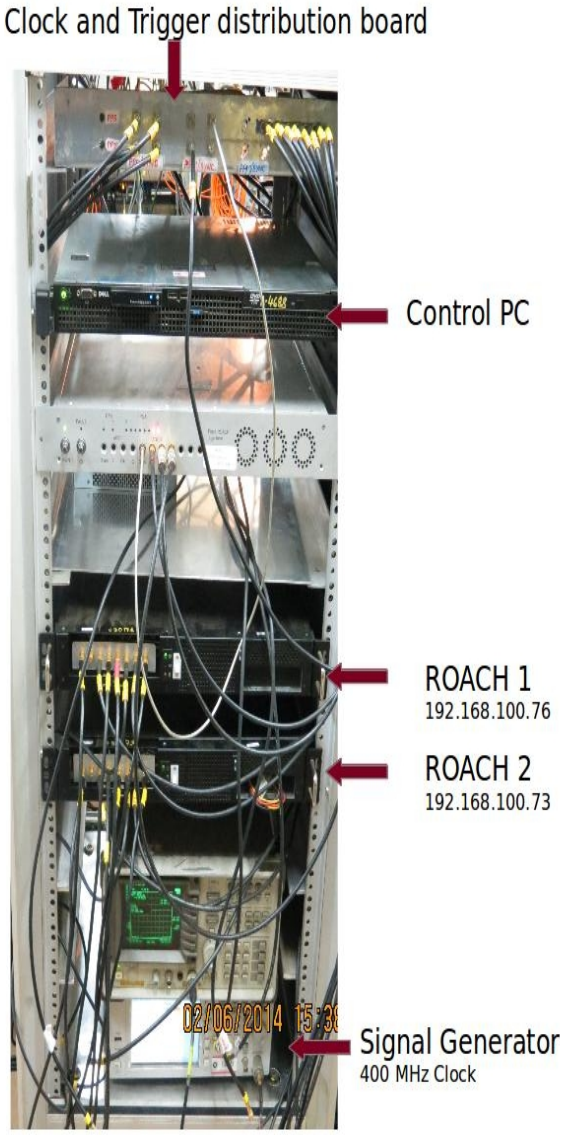
GWB racks

Image courtesy : Irappa M. Halagali

Compute and Host nodes rack



Sampler and Packetizer rack :



Images courtesy : Irappa M. Halagali

1. Switch OFF procedure

a. Switch off the ROACH UNITS 192.168.100.76 & 192.168.100.73 in the GWB rack by holding down the Black switch on the front panel for ~5 sec.

b. Switch off the Signal Generator, kept at bottom of the GWB rack. This feeds clock signal of 400 MHz, +17dBm to the ROACH borads.

c. No need to switch off the SMPS used for PPS unit. This will get switched off directly from mains.

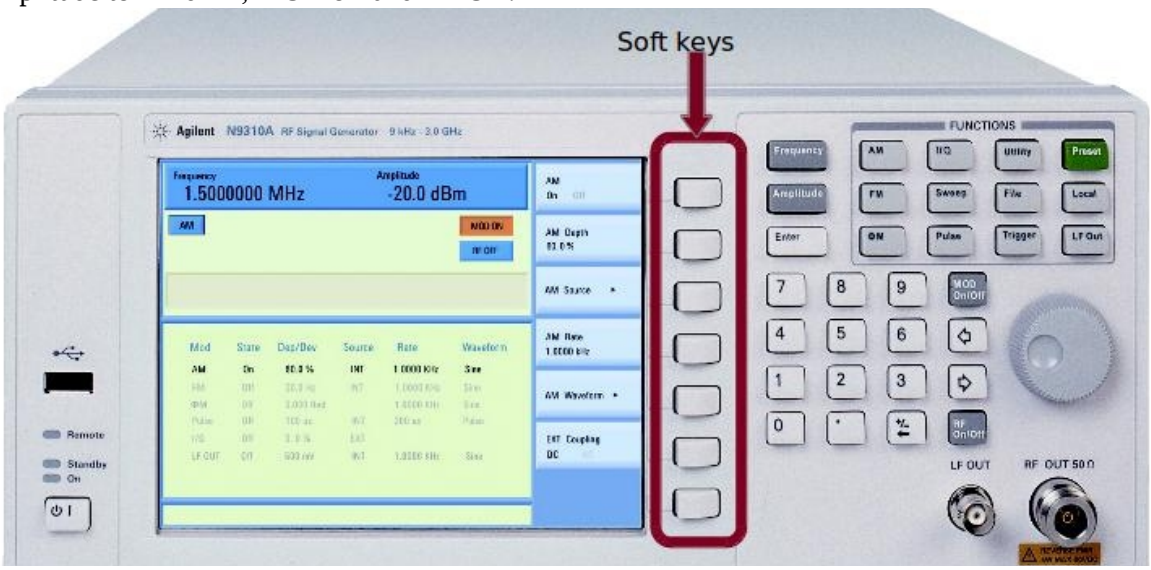
d. No need to switch off the infiniband switch. This will get switched off directly from mains.

e. Halt the control PC (192.168.4.68) which is a 1U pc in the GWB rack.
NOTE : a. ssh -X root@192.168.4.68 (gmrttifr) b. halt -p

f. Halt the compute nodes 192.168.4.75(gpunode6), 192.168.4.76(gpunode5), 192.168.4.77(gpunode4) & 192.168.4.78(gpunode3).
NOTE1 : a. ssh -X root@192.168.4.xx (123$abc) b. halt -p
NOTE2 : Switch off the 4 host nodes, node 192.168.16.51(node51), 192.168.16.52(node52), 192.168.16.53(node53) & 192.168.16.54(node54) with GSB system only.
NOTE1 : a. ssh -X root@192.168.16.xx (123$abc) b. halt -p

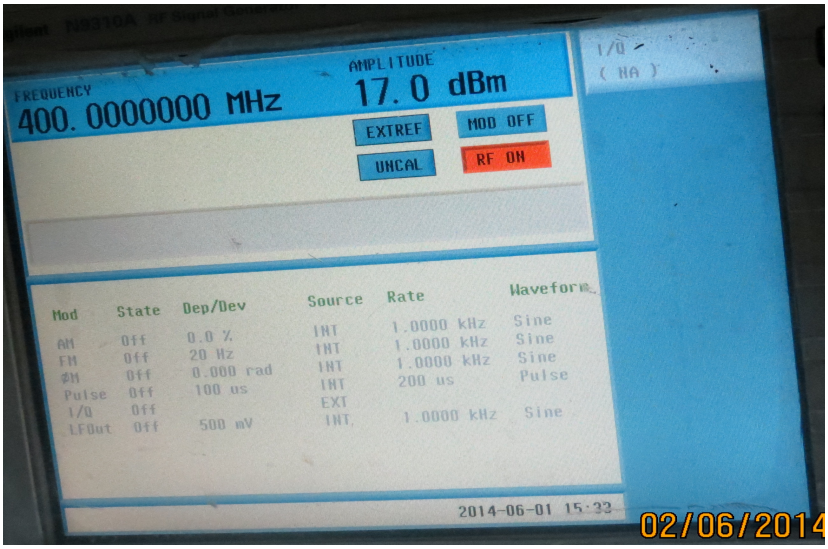
2. Switch ON procedure

a. Switch ON the control PC (192.168.4.68).
b. Make sure the infiniband switch is ON.
c. Make sure the PPS unit is Switched ON.
d. Switch ON the Signal Generator, kept at bottom of the GWB rack. Set the frequency to 400 Mhz, amplitude to +17dBm, MOD off and RF ON.



Steps : 1. Press “Frequency” key
2. Type “400” from the numbers
3. Select “MHz” from the “soft keys”
4. Press “Amplitude” key
5. Type “17” from the numbers
6. Select “dbm” from the soft keys
7. Press “MOD On/Off” key to turn it off.
8. Press “RF On/Off” key to turn it on.
9. Press “Utility” key
10. Select “Reference” from the soft keys
11. Select “EXT REF” from the soft keys

Final screen on the Signal generator should look like the one given in the picture below :



e. Switch ON the ROACH UNITS 192.168.100.76 & 192.168.100.73 in the GWB rack by holding down the Black switch on the front panel for ~2 sec.

f. Swich ON the compute nodes 192.168.4.75(gpunode6), 192.168.4.76(gpunode5), 192.168.4.77(gpunode4) & 192.168.4.78(gpunode3). And host nodes, node 192.168.16.51(node51), 192.168.16.52(node52), 192.168.16.53(node53) & 192.168.16.54(node54) with GSB system.

Appendix – 2 GWB NETWORK DIAGRAM

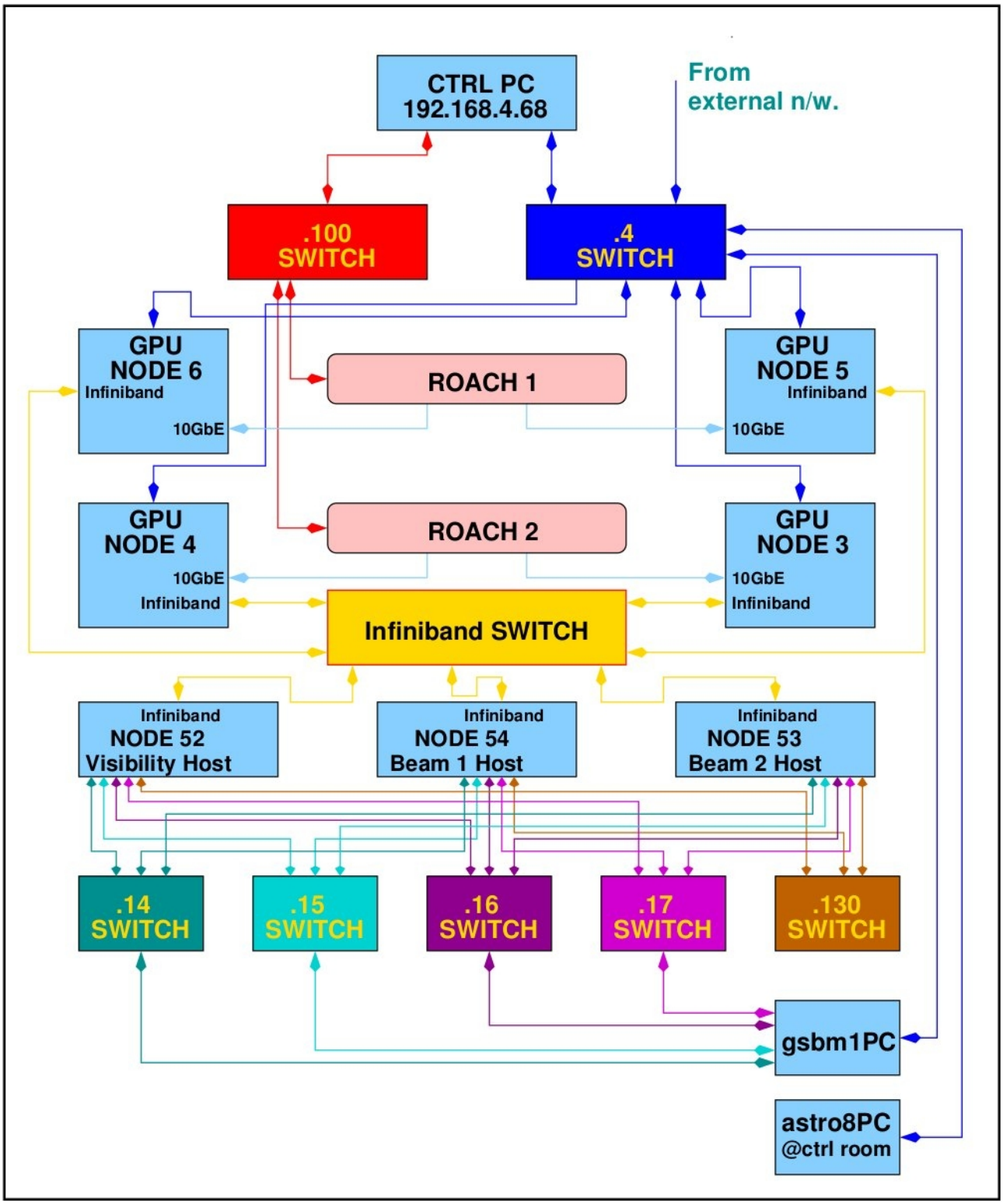


Image courtesy : Irappa M. Halagali