GMRT

National Center for Radio Astrophysics

Tata Institute of Fundamental Research

Pulsar Observing Manual

Modes : *IA bin, PA bin & PMT*

LastRevision: May 4, 2009

By: Reji Mathew C.Thomas

Section 1 : General Instructions

1.1 Introduction:Pulsar observation

This manual describes the general instructions for running the pulsar back-ends and data acquisition for the following :

(1) **Incoherent Array Receiver (IA bin -- USB only)**,
(2) **Phased Array Receiver (PA bin -- USB only)**
(3) PMT – IA mode (LSB and USB)
(4) PMT – PA mode (LSB and USB)

To set up the windows needed for acquisition from these different back-ends, individualised shell scripts are available in /home/pulsar/WorkDesk under the NIS "pulsar" login. The current working versions of the shell scripts are : ia_g3n (for IA bin), pa_g3n (for PA bin), ia_pa_g3n (for IA-PA bin on same desktop), pmt_lsb_g2n (for PMT LSB), pmt_usb_g2n (for PMT USB) & pmt_lsb_usb_g2n (for both USB & LSB of PMT). Use the appropriate script for the desired back-end. On login to any of the acquisition PCs -- gpsr, gpsrnode1, gpsrnode2 and gpsrnode3 -- ***use 'tcsh'*** and ***choose work or beta version appropriately***. For some of the windows created using shell script user may need to re-login to gpsrnode2 or gpsrnode3. This is explicitly mentioned in this manual. On acquisition PC the directory where all .hdr files are to be changed is: /home/pulsar/support/acqpsr/work. The directory that contains all executables is : /home/pulsar/bin/acqpsr/work.

1.2 Different Pulsar Back-ends at GMRT : Advance Briefing

A short description pertaining to the different pulsar observing back-ends available at GMRT is given below. Further details can be obtained at resource pages of GMRT observer help.

http://www.gmrt.ncra.tifr.res.in/gmrt_hpage/Users/doc/manual/UsersManual/

Three different back-end systems are attached to the GAC for processing the incoherent and coherent array outputs. These are the multi-bit incoherent array (IA) back-end, the multi-bit phased array (PA) back-end, and the polarimeter (PMTR).

IA pulsar back-end :(gpsrnode3)

The IA processor takes the corresponding GAC output signals and can integrate the data to a desired sampling rate (in powers of 2 times 16 µ-sec). It gives the option of acquiring either one of the polarizations or the sum of both. It can also collapse adjacent frequency channels, giving a slower net data rate at the cost of reduced spectral resolution. The data acquired is recorded on disk or directly on to SDLT tapes.

PA pulsar back-end :(gpsrnode3)

The PA pulsar back-end takes the dual polarization, coherent (voltage sum) output of the GAC and can produce an output which gives , the intensities for each polarization and the real and imaginary parts of the cross product and hence the four Stokes parameters are constructed. This hardware can be programmed to give a sub-set of the total intensity terms for each polarization or the sum of these two. The minimum sampling interval for this data is 32 µ-sec, as two adjacent time samples are added in the hardware. The data acquired is recorded on the disk or directly on to SDLT tapes.

PMTR back-end:(gpsrnode1 for USB & gpsrnode2 for LSB)

The PMTR back-end connected to the GAC produces four Stokes parameters I, Q, U and V. It has a pulsar mode for pulsar observations and a spectral mode for spectral averaging of the phased array sum of GAC. In pulsar mode, the input time series is folded to produce full Stokes integrated profiles at full spectral resolution, which are written to hard disk. In addition, there is a provision for on-line incoherent de-dispersion and pulse gating. In spectral mode, the instrument can record the spectral Stokes data after appropriate time averaging. The reduced data volumes are small and can be recorded to the local hard disk. A time series mode for this back-end is also available, wherein full spectral resolution data is available at sampling intervals ranging from 16 µ-sec upwards, which results in significantly higher data rates.

1.3 Some conventions followed in the manual:

The following conventions for the Fonts are followed through the manual if not otherwise stated.

All the executables are given in `Courier Font' :for eg.

```
> config_stokes_usb config_ia
```

All actions related to the proximate executable is displayed in italics with appropriate font sizes.:for eg.

Configuring the DSP bin:

All the notes are displayed either in italics or normal fonts. Bold face or light face fonts (with different Font sizes) are interspersed in some descriptions solely for the ease of discerning the priority of text. **The window names are designated in the manual as shown below . Corresponding windows which are displayed on the desktop can be easily identified with it.**

PA.gpsrnode3.sockngetcmd

Section 2. Initial set up:

2.1 Pulsar console

Login to the PC for data acquisition and do the following in a terminal (in case no console like the one in the bottom seen. If the following console is already seen then start with* step-1.** ***The help of operators are to be invoked to maximum in executing the initial set up.

Console login to pulsar machine " bhanu.gmrt.ncra.tifr.res.in'.

```
> ssh -X astro0
```

(login)
```
> pulsar
```

password
```
> psr1257+12
```

Open a console on a desktop `GAC'

```
> tcl
> ./pulsar_console
```

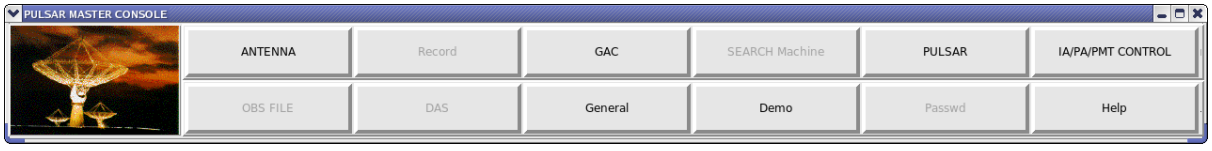
(*To set up all the pulsar parameter displays if they are not already opened*)

---

step-1: (designation valid through the manual)

The following console will open:

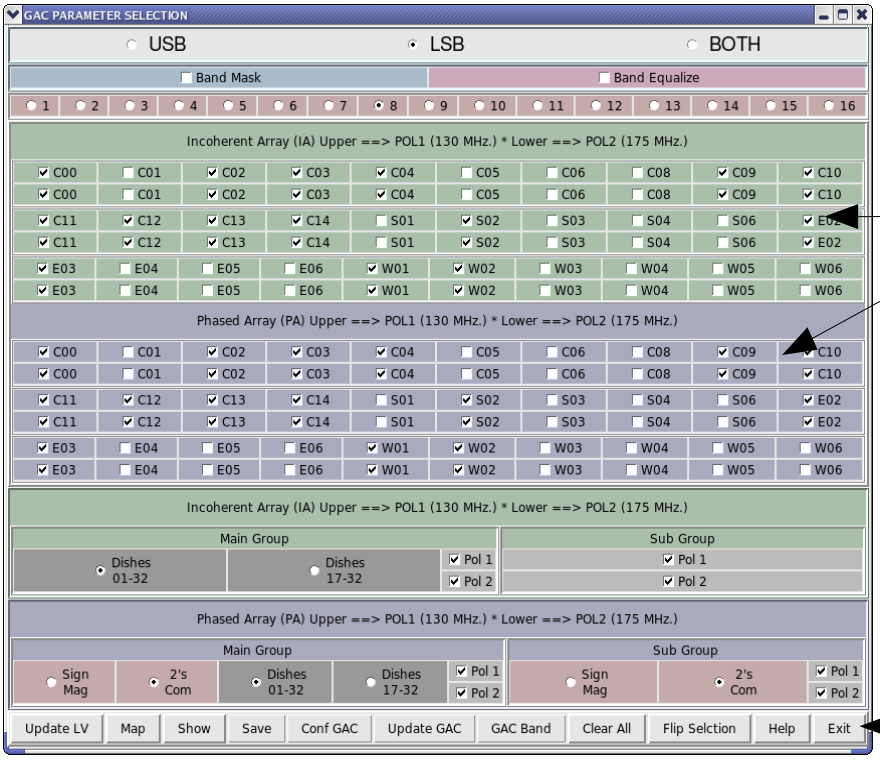
Figure.1



Select/click on **GAC**

Go to

GAC PARAMETER SELECTION (antennae names listed here)
***click 7* (for PA mode), *click 10* (for PMT mode)**
GAC PARAMETER SELECTION
then click the antenna -icons,
clicked icons display tick marks.



Tick mark appear in clicked icons for the chosen antennae

Click-sequence:Update LV,Show, Save, Conf GAC, GAC band.

2.2Antennae:Selection & Set-up:

(please seek help of operators)

Consult the operators to decide the number of antennae to be included. Please mark that all the 30 antennae might not be in concurrent working condition and hence discretion is solicited of the observer to select and include the antennae according to the specific needs of observation. Usually for pulsar observations the central square antennae and a few more neighbourhood antennae are included. Antennae that show noisy patterns are to identified and avoided. The arm antennae are usually excluded owing to the faster de-phasing.

2.3 Power Equalization and Phasing

(If IA bin is used then do not execute section 2.3)

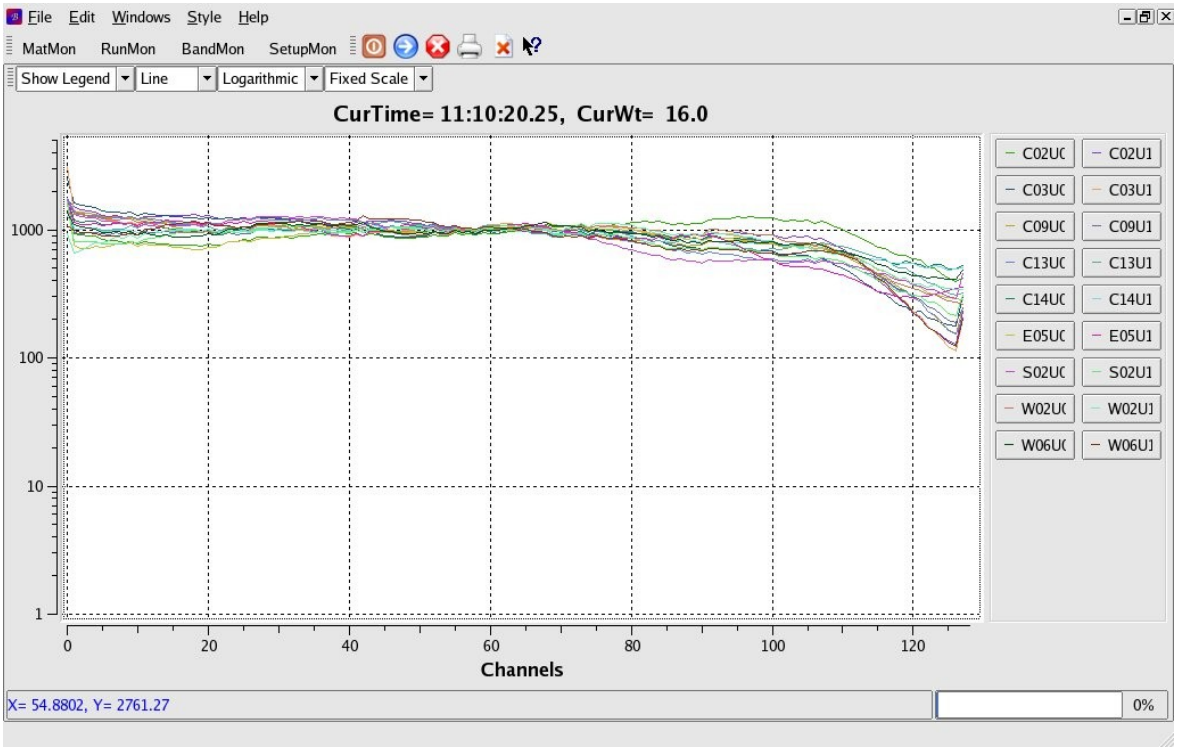
The optimum power level is kept at 1000 (arbitrary units) and all the included antenna are adjusted to this power level (with in a narrow +/- range). Operators are assigned do this job. Once this is over, equalize the phases of the antennae chosen. Be careful in executing this part since a de-phased antennae can kill the pulsar signal

Few sample plots are shown below:

The power equalization plot:ideal case

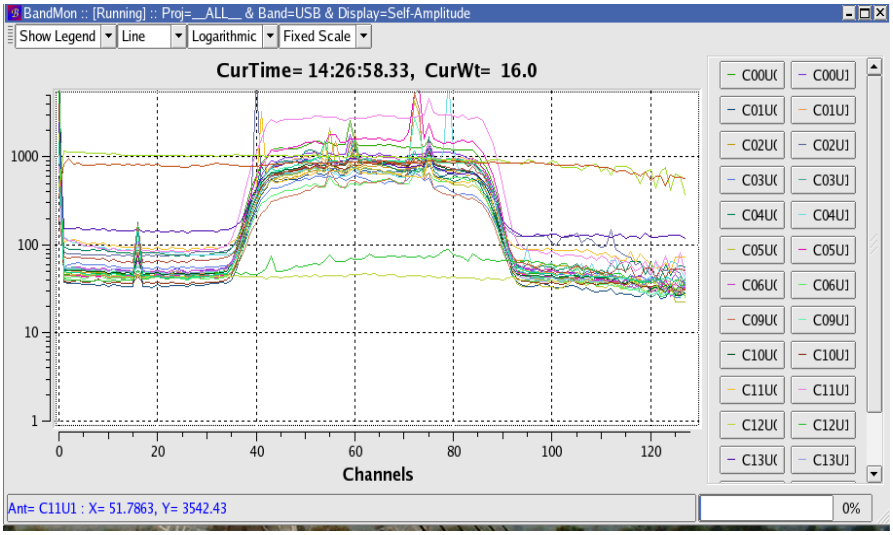
Figure.2

The channels numbers are shown on the horizontal axis at the bottom. The boxes seen on the right side margin designates each antennae. Individual power spectra of antenna can be seen in isolation by clicking the box corresponding to it.



RFI is seen as spikes in the following figure

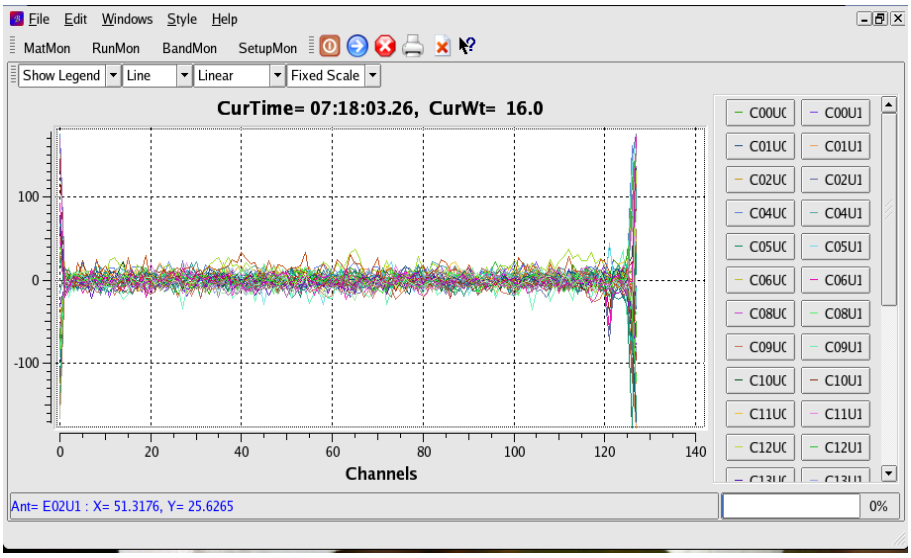
Figure.2



Phase plot:Ideal case

Figure.3

The channels numbers are shown on the horizontal axis at the bottom. The boxes seen on the right side margin designates each antennae. The phase in degrees are shown on the vertical axis. Individual power spectra of antenna can be seen in isolation by clicking the box corresponding to it.



-

2.4 Choosing the band-width (BW):

Default BW in general is 16 MHz. But for Low Freq. Observations at 150 MHz and 240 MHz the default BW is 6 MHz. Consult with the operators for alternative settings .

The pulsar pgplot window (applicable at a later stage only)

Figure.4

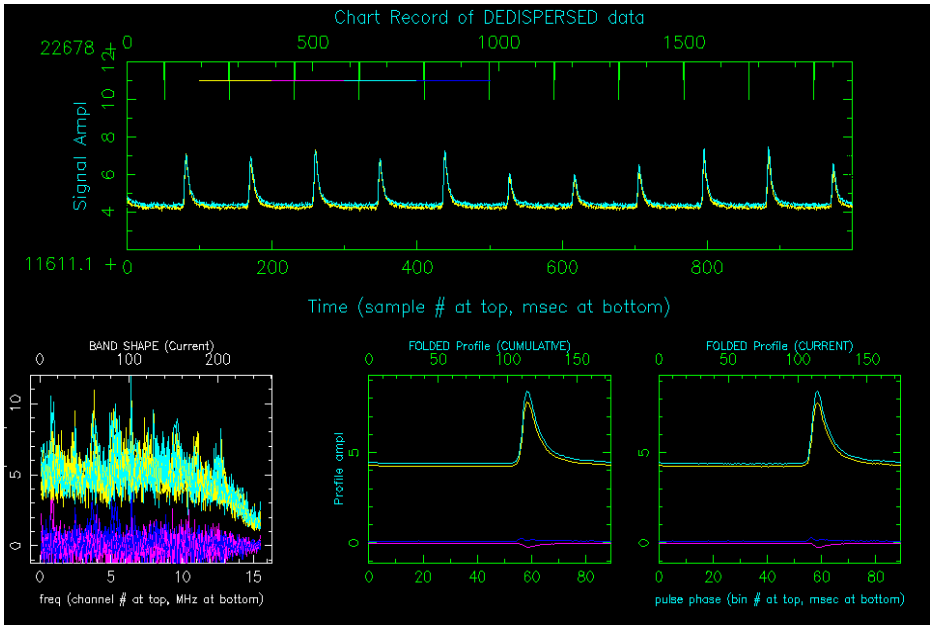


Table.1:the integration constants

(for later use only)

Number	μ-seconds **(PMT bin)**	μ-seconds **(PA bin)**
8	128	256
16	256	512
32	512	1024

Once phasing and power equalization done, then start setting the appropriate mode of observation as described in later pages.

Section 3 . Using the IA bin

Open a `Konsole' in desktop marked IA and do the following

```
> ia_g3n
```

(to get necessary windows)

step-4:

```
> exec tch
> w
```

Important: Do the step :4 in all the windows first, followed by the respective commands given under the window title

IA.gpsrnode3.run

Do step-4, then

```
> cd /home/pulsar/bin/IA_CTRL
>./edit
```
(For setting number of integration)

<Number of integrations> For setting number of integrations

(8 integrations- 256 micro secs) **> 8**

(16 integrations-512 micro secs) **Or >16**

(32 integrations-1024 micro secs) **Or *>32***

`< Number of channels>` ***> 256***

`<for upper polar>` ***> 2***

`<for lower polar >` ***> 5***

`<dual polar>` ***> 7***

`<Number of adjacent channnels added>` ***> 0***

`<Shift in data (LSB to MSB)>` ***> 0***

- *give the command* `./run_mk` → *boot the nodes*

Acquisition Program

IA.gpsrnode3.acqpsr

Do step-4, then

```
> acqpsr -Lp  0 1 [Integ]
```

(example)

`> acqpsr -Lp  0 1 32`

Integ →*Total number of integrations done in h/w*

IA.gpsrnode3.process_psr

Do step-4, then

`> vim ia_init.hdr`

'INTEG' has value of total number of hardware integrations. (This value is EQUAL to the Number of integrations specified in 'edit'). All other information is normally unchanged.

`> vim ia_scan.hdr`

Entry of 'TIME_INT' (time integrations) and 'FREQ_INT' (frequency integration), the number of post integrations that are to be performed should be proper.

`> vim hosts.dat`

Entry of name of the host where data is to be recorded with 'LogSock' as a socket is uncommented

`> process_psr -Lmpg -M 0 1`

IA.gpsrnode3.collect_psr

Do step-4, then

`> collect_psr`

IA.gpsrnode3.sockngetcmd

Do Step.4 , then

```
> sockngetcmd 2
> dasia  init  ia_init.hdr
```

commands need to be given from sockngetcmd window. ***Initialization of process_psr and collect_psr***

(Socket connection beween process_psr and collect_psr is formed)

```
> dasia  start  ia_scan.hdr
> dasia mon <sample number>  <channel number>
```

IA.gpsrnode3.psr_mon

Do step-4, then

```
> psr_mon -c  -n [] [] [] -m [] -s [] -i [] -M 0 -
                    d [] -P<>
```

(Example : For IA total-intensity data)

```
>  psr_mon  -c -n 1 0 0 -m  2.0 -M 1 -i  1
```

(psr_mon reading data from shm of collect_psr:)

(pgplot display prompt) **> /XS**

(opens the pgplot for the pulsar profile)

descriptions:

- -a : reading data from acqpsr
- -c :reading data form collect_psr
- -f : reading from a an already recorded file
- -n : [Total pols in data] [start pol] [stop pol] (0 is starting pol)
- -m: magnification factor
- -s: Stokes _sel ,can have values 0,1,2. default 0
- -i: 1 : invert, 0 : do not invert the data.
  -set i=0 for -c option and set i= 1 for -a option. (valid for PA/IA data only ,for Pmtr use i=0 always.).
- -d: Data length; 2 – short int (automatically this will be taken, if not specified otherwise), 4 – float

- -M :Which shm to attach. 0 - ia, 1 -pa, 2 -pmtr
- -P<>: Dont plot selected plot;
  d: chart plot
  a: accumulated (cumulative) profile
  c: current profile
  b: bandshape
- change the parameters for psr_online.in for the particular pulsar. Enter appropriate values of sampling interval, DM, Period of Pulsar.

Note: ***If only the observer is satisfied with the pulsar profile corresponding to the already acquired initial data , then go for recording data as given in further steps.***

IA.gpsrnode3.record_psr

<u>Do step-4</u>, ***then***

`> cd <data directory>`

`(/data/rawdata/<dir.name>)`
`or (/data2/rawdata/<dir.name>)`

`> record_psr -q -s/-f [] -m [] -t/-b [] -n [] -T`

descriptions:

- "record_psr -p 'dir name' -f 'file name' -n 3 -t 20.0"
- 'record_psr -p /data/rawdata/23nov -f b0826-34.raw1 -n 3 -t 20.0'
- -q -- quiet mode (with no display of recording information)
- -s/-f -- -f -- follows a file name for recording. -s -- suffix to the file name created by program, name format GP_yyddmm_mmhh_<suffix>.raw
  suggested suffix: 3 characters - hardware[d/p] mode[i/p] sideband[u/l]
  d - dsp bin, p - polarimeter, i - ia, p - pa, u - upper sideband, l - lower sideband
- -m -- data mode: 0 Write data and header in the SAME file
  1 Write data and header in the DIFFERENT files, header will be written in the file with `.hdr' extension.
- -T -- tape mode: When not defined, create new file with extension, if file with the given name exists. When defined, overwrite the existing file, doesnt produce file with extension, useful when using tapes (**mention device name using -f option).** It wont close the file till the session is over.

- -t/-b -- Count, -t for time count (in minutes), -b block count
  0 - for infinite reading or till ctrl-c is pressed or record_psr stopped using dasia stop
- -n -- number of scans, 0 -

Recording data with out processing , using read_acqpsr

IA.gpsrnode3.read_acqpsr

Do step-4, *then*

```
> read_acqpsr -Lmp <data file> <Number of total H/W integrations> 0
```

When recoding is over use 'ctrl-c' to stop the acquisition.

2.4 Finishing the Observation:

IA.gpsrnode3.sockngetcmd

```
> dasia   finish
```

IA.gpsrnode3.psr_mon

To stop 'psr_mon' use 'ctrl-c'.

IA.gpsrnode3.read_acqpsr

To stop acquisition from read_acqpsr use 'ctrl-c'.

Backup of data:

Insert the data tape in to the respective drive (in gpsr) and start the backup SDLT -2 tape drive is connected to the `gpsr' machine and which facilitates the backup of data from gpsrnode3.

Example: > tar cvf /dev/nst0 <file or directory name>

Section 4. Using the PA bin

Open a Konsole on the workspace (desktop) marked ` PA'.

step 2: `> cd WorkDesk`

step-3: `> pa_g3n`

(by default logged in to gpsrnode3)

the command generates all the necessary windows.

step-4:

(*on each of the window on gpsrnode3*)

> tcsh

> w

Important: Do the step :4 wherever mentioned followed by the respective commands falling under the window title

Configuring the PA back-end : Here the inverse FFT bin and the DSP bin need to be configured.

PA.gpsrnode3.run

Configuring the inverse FFT bin

Login: - `> das`

password - `> new.das`

```
> cd /home/das/pa_cnt1/ifft/fft_config
> ./pa_ifft_config
```

Configuring the DSP bin:

> cd /home/das/pa_cnt1/dsp/dsp_config

> ./edit1

<Number of integrations> For setting number of integrations

(8 integrations- 256 micro secs) **> 8**

(16 integrations-512 micro secs) **Or >16**

(32 integrations-1024 micro secs) **Or >32**

NOTE : The final integrated data is twice of the above-given hardware integrations !!

<Number of fft channels > ***> 256***

- *<Number of adjacent channels added > Generally used value:* ***> 0***

<Number of LSB's to be dropped/output shift > Generally used value: ***> 0***

For PA full polar mode. **> ./run**

or For PA Total Intensity mode. **> ./run_totn**

PA.gpsrnode3.acqpsr

Do Step.4, ***then***

> acqpsr -Lp [mode] [Pols] [Integ]

Note: Integ. value should be twice the value <Number of integrations> given earlier

Example: if 8 is is the value given earlier , then type for Integ --> 8X2 =16

---

example: For full-polar

>acqpsr -Lp 1 4 16

example: For total intensity

>acqpsr -Lp 1 1 16

description:

- mode: 1 -> *For PA mode*
- Pols -> Total number of polarization;
  for total intensity Pols =1, for full polar Pols = 4
- Integ →Total number of integrations done in h/w
  (Errors will be logged in file: pa_acqerror.log when option -Lp is mentioned)

(i) process_psr : Reads pulsar data from shared memory created by acqpsr, processes the data (marker check , GPS time stamping are done).

(ii) collect_psr : Collects the data over socket sent by process_psr. Writes this received data into another shared memory.

(iii) record_psr : Reads data from shared memory created by collect_psr and writes that data file on local disk.

(iv) sockngetcmd : Takes commands from user and passes these to process_psr.

PA.gpsrnode3.process_psr

Do step. 4 *then*

```
> vim pa_init.hdr
```

'INTEG' -> *total number of hardware integrations.* INTEG *value is twice the number specified in 'edit1'.* **change the value of INTEG accordingly.** *The rest is unchanged.*

```
> vim pa_scan.hdr
```

Entry of 'TIME_INT' *(time integrations) and* 'FREQ_INT' *(frequency integration), the number of post integrations that are to be performed should be proper.*

This provides us with the scope of doing integration (in addition to the hardware integration) with in process_psr.

```
> vim hosts.dat
```

Entry of name of the host where data is to be recorded with 'LogSock' *as a socket is uncommented (# is a comment character). Some possible names are written in the file. Uncomment the name of the host where data is to be acquired.*

PA Total intensity:

```
> process_psr -Lmpg  -M  1  1
```

PA full polar (4 stokes):

```
> process_psr -Lmpg  -M  1  4
```

PA full polar (2 stokes):

```
> process_psr -Lmpg  -M  1 2
```

PA.gpsrnode3.collect_psr

Do step. 4 *then*

```
> collect_psr
```

PA.gpsrnode3.sockngetcmd

Do step. 4 *then*

```
> sockngetcmd 2
>daspa init pa_init.hdr
```

The command that initialize process_psr and collect_psr Socket ,connection between process_psr and collect_psr is formed

```
>daspa start pa_scan.hdr
```

-command to be given at sockngetcmd to start collecting data.

PA.gpsrnode3.psr_mon

Do step. 4 *then*

```
> vi   psr_online.in
```

Edit `psr_online.in' for the pulsar period (milli-sec) and DM

(For PA total-intensity data)

```
> psr_mon -c -n 1 0 0 -m 2.0 -M 1 -i 1
```

(For PA 4-stokes polarization data)

```
> psr_mon -c -n 4 0 3 -m 2.0 -M 1 -i 1
```

(psr_mon reading data from shm of collect_psr:)

----

description:

- -a : reading data from acqpsr
- -c :reading data form collect_psr
- -f : reading from a an already recorded file
- -n : [Total pols in data] [start pol] [stop pol] (0 is starting pol)
- -m: magnification factor
- -s: Stokes _sel ,can have values 0,1,2. default 0
- -i: 1 : invert, 0 : do not invert the data.
  -set i=0 for -c option and set i= 1 for -a option. (valid for PA/IA data only ,for Pmtr use i=0 always.).
- -d: Data length; 2 – short int (automatically this will be taken, if not specified otherwise), 4 – float
- -M :Which shm to attach. 0 - ia, 1 -pa, 2 -pmtr
- -P<>: Dont plot selected plot;
  d: chart plot
  a: accumulated (cumulative) profile
  c: current profile
  b: bandshape
- change the parameters for psr_online.in for the particular pulsar. Enter appropriate values of sampling interval, DM, Period of Pulsar

(pgplot display prompt) **> /xs**

(see figure.4)

Note: If only the observer is satisfied with the pulsar profile corresponding to the already acquired initial data, then go for recording the data.

PA.gpsrnode3.record_psr

Do step. 4 *then*

```
cd <data directory>
eg:  /data/rawdata/<dir.name>
   or  /data2/rawdata/<dir.name>
> record_psr -m [] -n [] -f []-T
```

example :

```
>record_psr -m 1 -n 1 -f <filename> -t 20.0
```

- "record_psr -p /data/rawdata/27nov -f b0826-34.raw1 -m 1 -n 1 -t 20.0"
- -f -- follows a file name for recording. -s -- suffix to the file name created by program, name format GP_yyddmm_mmhh_<suffix>.raw (optional) suggested suffix: 3 characters - hardware[d/p] mode[i/p] sideband[u/l]

  d - dsp bin, p - polarimeter, i - ia, p - pa, u - upper sideband, l - lower sideband
- -m -- data mode: 0 Write data and header in the SAME file

  1 Write data and header in the DIFFERNET files, header will be written in the file with `.hdr' extension. (generally choose m=1)
- -T -- tape mode: When not defined, create new file with extension, if file with the given name exists. When defined, overwrite the existing file, doesnt produce file with extension, useful when using tapes **(mention device name using -f option).** It wont close the file till the session is over.
- -t/-b -- Count, -t for time count (in minutes), -b block count
  0 - for infinite reading or till ctrl-c is pressed or record_psr stopped using daspa stop
- -n -- number of scans, 0 - for infinite

To repeat the data recording: `ctrl -c' and type the command ` record_psr' etc.

Recording data without processing using read_acqpsr:

Windows -(i) read_acqpsr : Reads data from shared memory of acqpsr and records it on local disk. It does not do any processing of the data.

(ii) sockngetcmd: Takes commands from user and passes to process_psr.

PA.gpsrnode3.read_acqpsr

Do step. 4 *then*

PA Total Intensity:

> read_acqpsr -Lmp <data file><Number of total H/W integrations>1 1

PA Full polar (4 stokes):

>read_acqpsr -Lmp <data file><Number of total H/W integrations>1 4

PA Full polar (2 stokes):

>read_acqpsr -Lmp <data file> <Number of total H/W integrations>1 2

When recoding is over use 'ctrl-c' to stop the acquisition.

➢

Finishing the Observation:

PA.gpsrnode3.sockngetcmd

> daspa finish

PA.gpsrnode3.psr_mon

To stop 'psr_mon' use 'ctrl-c'.

PA.gpsrnode3.read_acqpsr

To stop acquisition from read_acqpsr use 'ctrl-c'.

-

Backup of data:

Insert the data tape in to the respective drive (in gpsr) and start the backup SDLT -2 tape drive is connected to the `gpsr' machine and which facilitates the backup of data from gpsrnode3.

Example: > tar cvf /dev/nst0 <file or directory name>

Section5. Using PMT (IA bin)

Open a Konsole and execute the following for opening the windows

----

(For USB)

```
> cd WorkDesk
 > pmt_usb_g2n
```

Or (For LSB)

```
> cd WorkDesk
 > pmt_lsb_g2n
```

Note: Configuring polmtr for usb (similar steps for lsb)

`_usb' for USB and `_lsb" for LSB. 16_pa for 16 integrations

32_pa for 32 integrations etc. Replace accordingly in the following

PMT_USB:gpsrnode2:Run

login: **> observer**

password **> obs@gmrt**

```
> cd  /home/observer/SPECTRA/LSB_USB_work

> config_stokes_usb config_ia
> run_spec_usb 16_ia ionly16_n
```

Step.3 (Do wherever mentioned)

(For USB)

```
> ssh  gpsrnode1
 > exec tcsh
 >  b
 > cd USB
```

(For LSB)

```
> ssh  gpsrnode2
> exec tcsh
> b
> cd LSB
```

(Note: gpsrnode2 for USB gpsrnode1 for LSB)

PMT_USB:gpsrnode2:acqpsr

Do Step.3 then

```
 > acqpsr -Lp 2 1 8 1
```

Note:

- mode:
  - 2 → For PMT mode
  - port for USB port=1, for LSB port=0
  - Hard ware Integration time(8,16,32)

(Errors will be logged in file: pmt_acqerror.log when option -Lp is mentioned)

PMT_USB:gpsrnode2.process_psr

Do Step.3 , then

> vim pmt_init.hdr

'INTEG' has value of total number of hardware integrations. Depending on the number of integrations, the Marker pattern and OUTCLK values has to be changed accordingly. All other information is normally unchanged.

> vim hosts.dat

Entry of name of the host where data is to be recorded with 'LogSock' as a socket is uncommented (# is a comment character.)

> vim pmt_scan.hdr

Entry of 'LTA' (time integrations) and 'CHAN_NUM' (frequency integration), the number of post integrations that are to be performed should be proper.

CHAN_NUM is expressed as range. So last number in range (due to historical reasons) indicates required value (i.e. In ff:mm:ll, 'll' represents 'CHAN_NUM' value.

PMT Total intensity: e.g.

```
>process_psr -Lmpg  -M 2  1 -S
```

PMT:gpsrnode2.collect_psr

Do Step.3 , then

```
>collect_psr
```

PMT:gpsrnode2.sockngetcmd

Do Step.3 , then

```
>sockngetcmd 2
>daspmt init pmt_init.hdr
```

Socket connection beween process_psr and collect_psr is formed)

(Starting data recording)

PMT:gpsrnode2.record_psr

Do Step.3 , then

```
> cd <data directory>
> record_psr -q -s/-f [] -m [] -t/-b [] -n [] -T
```

Notes:

- e.g "record_psr -p /data/rawdata/23nov/ -f filename -n 1 -m 1 -t 20
- q -- quiet mode
- -s/-f -- -f -- follows a file name for recording. -s -- suffix to the file name created by program, name format GP_yyddmm_mmhh_<suffix>.raw
  suggested suffix: 3 characters - hardware[d/p] mode[i/p] sideband[u/l]
  d - dsp bin, p - polarimeter, i - ia, p - pa, u - upper sideband, l - lower sideband
- -m -- data mode: 0 Write data and header in the SAME file
  1 Write data and header in the DIFFERNET files, header will be written in the file with `.hdr' extension.
- -T -- tape mode: When not defined, create new file with extension, if file with the given name exists. When defined, overwrite the existing file, doesnt produce file with extension, useful when using tapes **(mention device name using -f option).** It wont close the file till the session is over.
- -t/-b -- Count, -t for time count (in minutes), -b block count
  0 - for infinite reading or till ctrl-c is pressed or record_psr stopped using `das stop`
- -n -- number of scans, 0 - for infinite.

PMT:gpsrnode2.sockngetcmd

Do Step.3 , then

```
>daspmt start pmt_scan.hdr
> daspmt mon <sample number>  <channel number>
```

PMT:gpsrnode2.psr_mon

Do Step.3 , then

```
> vi psr_online.in
```

Edit ` psr_online.in' for the pulsar period (milli-seconds) and DM

```
psr_mon -c -n [] [] [] -m [] -s [] -i [] -M 2 -d [] -P<>
```

Example:

> psr_mon -c -n 1 0 0 M 2 -i 0 -m 5

- -a use acqpsr's shm
- -c use collect_psr's shm
- -f use file for reading data
- -n : [Total pols in data] [start pol] [stop pol] (0 is starting pol)
- -m: magnification factor
- -s: Stokes _sel can have values 0,1,2
- -i: 1 : invert, 0:do not invert the data . (use i = 0 for pmtr data)
- -d: Data length; 2 – short int(default) , 4 – float
- -P<>: Dont plot selected plot;
  - d: chart plot
  - a: accumulated (cumulative) profile
  - c: current profile
  - b: bandshape

PMT:gpsrnode2.sockngetcmd

Do Step.3 , then

Stop sending data

> daspmt stop

Start the second scan by the giving command

> daspmt start pmt_scan.hdr

Recording data without processing using read_acqpsr:

PMT:gpsrnode2.read_acqpsr

Do Step.3 , then

PMT Total Intensity:

>read_acqpsr -Lmp <data file> <No.total H/W integrations> 2 1

PMT Full polar (4 stokes):

> read_acqpsr -Lmp <data file> <No. total H/W integrations> 2 4

PMT Full polar (2 stokes):

```
>read_acqpsr -Lmp <data file> <No. total H/W
integrations> 2 2
```

- 'ctrl-c' to stop the acquisition.

Finishing the Observation:

PMT:gpsrnode2.sockngetcmd

```
> daspmt finish
```

PMT:gpsrnode2.psr_mon

To stop psr_mon use 'ctrl-c'.

PMT:gpsrnode2.read_acqpsr

To stop acquisition from read_acqpsr use 'ctrl-c'.

Section .6 Using PMT (PA bin)

Open a console on the desktop marked ` PA'. Then do the following

step-2 :

```
> cd WorkDesk
        > pmt_usb_g2n   (for USB only)
or      > pmt_lsb_g2n   (for LSB only)
```

PMT_USB_run

login: observer

password: obs@gmrt

```
> cd  /home/observer/SPECTRA/LSB_USB_work
```

Note:` _usb' for USB and ` _lsb' for LSB. 16_pa for 16 integrations 32_pa for 32 integrations etc. Replace accordingly in the following

Case 1.-Total Intensity:

```
> config_stokes_usb 4stokes

> run_spec_usb 16_pa ionly16_n
(for 32 integrations
```

case-2 -rr-ll:

```
> config_stokes_usb rr_ll
> run_spec_usb 16_pa stokes16_n
```

case-3 -4stokes :

```
>config_stokes_usb 4stokes
>run_spec_usb 16_pa stokes
```

step-3 : (**to be executed wherever mentioned)**

(For USB only)

```
> ssh gpsrnode1
 > exec tcsh
 > b
> cd USB
```

OR (For LSB only)

```
> ssh gpsrnode2
 > exec tcsh
 > b
 > cd LSB
```

PMT_USB:gpsrnode1:acqpsr

Do step.3

```
> acqpsr -Lp [mode] [Pols] [Integ] ,
```

example:

>acqpsr -Lp 2 1 16 1

---

descriptions:

- mode:
  - 2 → For PMT mode
- Pols → Total number of polarization;

for total intensity Pols = 1;

for full polar Pols = 4

- Integ →Total number of integrations done in h/w

(Errors will be logged in file: pmt_acqerror.log when option -Lp is mentioned)

PMT_USB:gpsrnode1:process_psr

Do step.3

```
> vim pmt_init.hdr
```

'INTEG' *has value of total number of hardware integrations. All other information is normally unchanged. Marker pattern and outclk has to be changed according to the*

'INTEG' *value selected.*

```
> vim pmt_scan.hdr
```

Entry of **LTA** *(time integrations) and* **CHAN_NUM** *(frequency integration), the number of post integrations that are to be performed should be proper.*

CHAN_NUM *is expressed as range. So last number in range (due to historical reasons) indicates required value (i.e. In ff:mm:ll, 'll' represents* **CHAN_NUM***value.*

```
> vim hosts.dat
```

Entry of name of the host where data is to be recorded with 'LogSock' as a socket is uncommented (# is a comment character.)

PMT Total intensity:

```
>process_psr   -Lmpg  -M 2 1 -S
```

PMT full polar (4 stokes)

```
>process_psr -Lmpg -M 2 4 -S
```

PMT full polar (2 stokes)

```
> process_psr -Lmpg -M 2 2 -S
```

PMT_USB:gpsrnode1:collect_psr

Do step.3

Note: user can decide to collect data in some other pc like gpsr to reduce load on the pc on which process_psr is running

```
> collect_psr
```

PMT_USB:gpsrnode1:sockngetcmd

Do step.3

```
> sockngetcmd 2
> daspmt init pmt_init.hdr
```

PMT_USB:gpsrnode1:record_psr

Do step.3

```
> cd <data directory>
>record_psr -m 1 -n 1 -f <filename> -t <minutes>
```

- -m -- data mode: 0 Write data and header in the SAME file
  1 Write data and header in different files, header will be written in the file with `.hdr' extension.
- -n -- number of scans, 0 - for infinite.

PMT_USB:gpsrnode1:sockngetcmd

Do step.3

```
> daspmt start pmt_scan.hdr
```

PMT_USB:gpsrnode1:psr_mon

Do step.3

```
> vi   psr_online.in
```

Edit `psr_online.in' for the pulsar period (milli-seconds) and DM

```
> psr_mon -c -n 1 0 0 -M 2 -i 0 -m 5
```

descriptions:

- -c use collect_psr's shm
- -n : [Total pols in data] [start pol] [stop pol] (0 is starting pol)
- -m: magnification factor
- -s: Stokes _sel can have values 0,1,2
- -i: 1 : invert, 0:do not invert the data . (use i = 0 for pmtr data)
- -d: Data length; 2 – short int(default) , 4 – float

PMT_USB:gpsrnode1:sockngetcmd

Stop sending data

```
> daspmt  stop
```

Start the second scan by the giving command

```
> daspmt start  pmt_scan.hdr
```

(As explained earlier.)

PMT_USB:gpsrnode1:acqpsr

Recording data without processing using read_acqpsr:

Total Intensity:

```
> read_acqpsr  -Lmp  <data file>  <Number of
total integrations>   2   1
```

PMT Full polar (4 stokes):

```
> read_acqpsr  -Lmp  <data file>  <Number of
total integrations>   2   4
```

PMT Full polar (2 stokes):

```
>read_acqpsr -Lmp  <data file>  <Number of total
integrations>  2  2
```

When recoding is over use 'ctrl-c' to stop the acquisition.

Finishing the Observation:

PMT_USB:gpsrnode1:sockngetcmd

```
> daspmt finish
```

PMT_USB:gpsrnode1:psr_mon

To stop 'psr_mon' use 'ctrl-c'.

PMT_USB:gpsrnode1:acqpsr

To stop acquisition from read_acqpsr use 'ctrl-c

Section .7 Using PMT (PA bin): Both sidebands (USB & LSB)

Step.2: `> cd WorkDesk`

`> pmt_lsb_usb_g2n`

(this opens two sets of windows one for USB and another set for LSB, configuring polmtr is done at only one window),

Important: Do Step.3 first followed by the commands written in the respective part in all the specified windows

Step.3

(for USB windows)

```
> ssh gpsrnode1
> exec tcsh
> b
```

(for LSB windows)

```
> ssh gpsrnode2
> exec tcsh
> b
> cd LSB
```

PMT_USB_run

login : observer

password : obs@gmrt

> **cd /home/observer/SPECTRA/LSB_USB_work**

total intensity mode:

>config_stokes_usb 4stokes

>config_stokes_lsb 4stokes

>run_spec_both 16_pa ionly16_n

(16_pa is for 16 integrations)

rr-ll:

> config_stokes_usb rr_ll

> config_stokes_lsb rr_ll

> run_spec_both 16_pa stokes16_n

4 stokes :

> config_stokes_usb 4stokes

> config_stokes_lsb 4stokes

> run_spec_both 16_pa stokes16_n

PMT:gpsrnode2:acq_psr

Do Step.3

> acqpsr Lp [mode] [pols] [Integ] [port]

(Example) > acqpsr -Lp 2 1 16 1

- mode:
  - 2 → For PMT mode
- Pols → Total number of polarzation;

for total intensity Pols = 1;

for full polar Pols = 4

- Integ →Total number of integrations done in h/w

- Port for USB port = 0, for LSB port =1

(Errors will be logged in file: pmt_acqerror.log when option -Lp is mentioned)

PMT:gpsrnode2:process_psr

Do Step.3

> vim pmt_init.hdr

'INTEG' has value of total number of hardware integrations. All other information is normally unchanged.

> vim hosts.dat

Entry of name of the host where data is to be recorded with 'LogSock' as a socket is uncommented (# is a comment character.)

> vim pmt_scan.hdr

Entry of 'LTA' (time integrations) and 'CHAN_NUM' (frequency integration), the number of post integrations that are to be performed should be proper.

CHAN_NUM is expressed as range. So last number in range (due to historical reasons) indicates required value (i.e. In ff:mm:ll, 'll' represents 'CHAN_NUM' value.

PMT Total intensity:

>process_psr -Lmpg -M 2 1 -S

PMT full polar (4 stokes)

> process_psr -Lmpg -M 2 4 -S

PMT full polar (2 stokes)

> process_psr -Lmpg -M 2 2 -S

Collecting the data

PMT:gpsrnode2:collect_psr

Do Step.3

user can decide to collect data in some other pc like gpsr to reduce load on the pc on which process_psr is running.

```
> collect_psr
```

PMT:gpsrnode2:sockngetcmd

Do Step.3

```
> sockngetcmd 2
> daspmb init pmt_init.hdr
> daspmb start pmt_scan.hdr
```

PMT:gpsrnode2:psr_mon

Do Step.3

```
> vi psr_online.in
```

Edit `psr_online.in' for the pulsar period (milli-seconds) and DM

(For PA total-intensity data)

```
> psr_mon  -c -n 1 0 0 -m  2.0 -M 1 -i  1
```

(For PA 4-stokes polarization data)

```
> psr_mon -c -n  4 0 3 -m 2.0 -M 1 -i 1
```

(psr_mon reading data from shm of collect_psr:)

----

descriptions:

- -a : reading data from acqpsr4.167000
- -c :reading data form collect_psr
- -f : reading from a an already recorded file
- -n : [Total pols in data] [start pol] [stop pol] (0 is starting pol)
- -m: magnification factor
- -s: Stokes _sel ,can have values 0,1,2. default 0
- -i: 1 : invert, 0 : do not invert the data.

-set i=0 for -c option and set i= 1 for -a option. (valid for PA/IA data only ,for Pmtr use i=0 always.).

- -d: Data length; 2 – short int (automatically this will be taken, if not specified otherwise), 4 – float
- -M :Which shm to attach. 0 - ia, 1 -pa, 2 -pmtr
- -P<>: Dont plot selected plot;
  - d: chart plot
  - a: accumulated (cumulative) profile
  - c: current profile
  - b: bandshape
- change the parameters for psr_online.in for the particular pulsar. Enter appropriate values of sampling interval, DM, Period of Pulsar

(pgplot display prompt) **> /XS**

(see figure.4)

Note: If only the observer is satisfied with the pulsar profile corresponding to the already acquired initial data, then go for recording the data.

Recording the data

PMT:gpsrnode2:record_psr

Do Step.3

```
> cd <data directory>
> record_psr -m 1 -n 1 -f <filename> -t
<minutes>
```

Filename: (Example) b2053+21_pmt_usb_157.raw0

- -s/-f -- -f -- follows a file name for recording. -s -- suffix to the file name created by program, name format GP_yyddmm_mmhh_<suffix>.raw
  suggested suffix: 3 characters - hardware[d/p] mode[i/p] sideband[u/l]
  d - dsp bin, p - polarimeter, i - ia, p - pa, u - upper sideband, l - lower sideband
- -m -- data mode: 0 Write data and header in the SAME file
  1 Write data and header in the DIFFERENT files, header will

be written in the file with `.hdr' extension.

- -t for time count (in minutes), 0 - for infinite reading or till ctrl-c is pressed or record_psr stopped using `das stop`
- -n -- number of scans, 0 - for infinite.

PMT:gpsrnode2:sockngetcmd

To stop sending data

```
> daspmb stop
```

Start the second scan by the giving command

```
> daspmb start pmt_scan.hdr
```

(As explained earlier.)

5.3(b) Recording data without processing using read_acqpsr:

PMT:gpsrnode2:read_acqpsr

PMT Total Intensity:

```
> read_acqpsr -Lmp <data file> <Number of total H/W integrations> 2 1
```

PMT Full polar (4 stokes):

```
> read_acqpsr -Lmp <data file> <Number of total H/W integrations> 2 4
```

PMT Full polar (2 stokes):

```
> read_acqpsr -Lmp <data file> <Number of total H/W integrations> 2 2
```

> When recoding is over use 'ctrl-c' to stop the acquisition.

Finishing the Observation:

PMT:gpsrnode2:sockngetcmd

```
> daspmt finish
```

PMT:gpsrnode2:psr_mon

To stop 'psr_mon use ***'ctrl-c'.***

PMT:gpsrnode2:read_acqpsr

To stop acquisition from read_acqpsr use 'ctrl-c'.

Section8. Using PMTR (IA bin) Both Side Bands(USB & LSB)

Step-2 : > cd WorkDesk

Step-3 : > pmt_lsb_usb_g2n

(this opens two sets of windows one for USB and another set for LSB, configuring polmtr is done at only one window),

Step-4 : For USB do cd USB in all respective windows, same for LSB]

Configuring polmtr

PMT:gpsrnode2:Run

login: > observer

password: > obs@gmrt

> cd /home/observer/SPECTRA/LSB_USB_work

total intensity mode:

> config_stokes_usb config_ia

> config_stokes_lsb config_ia

> run_spec_both 16_ia ia16_n

(16_pa is for 16 integrations)

(the rest follows similar procedure as of the PA bin).