SERENDIP6 User Manual V1.0

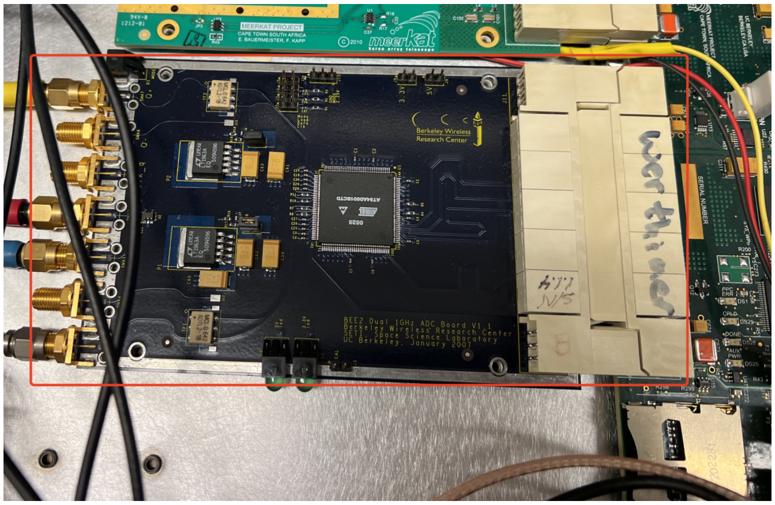
Wei Liu& Dan Werthimer 02/18/2023

This is a user manual for setting up SERENDIP6 at Medicina Radio Observatory(MRO).

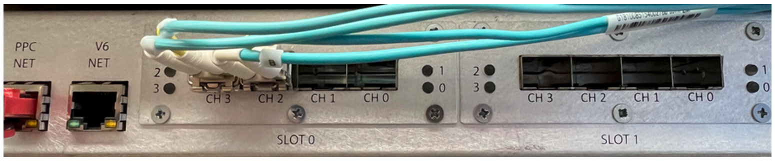
Hardware requirements

The SERENDIP6 receives packets from ROACH2 board, which connects an iADC board. The iADC board has to be attached to ZDOK1.

(**TODO: Take a new photo of the full board.)**



The ADC raw data for SERENDIP6 will be sent out from CH2 and CH3 on SLOT0, so please make sure the two fibers are connect to the two 10GbE ports.



Required libraries

1. Install HASHPIPE

   Please follow the instructions here: https://casper.astro.berkeley.edu/wiki/HASHPIPE

2. Other necessary libraries

```
sudo apt install libhiredis-dev
sudo apt install libcfitsio-devsudo
sudo apt install libmysqlclient-dev
```

Compile and install SERENDIP6

1. Clone the code from

```
git clone -b mro https://github.com/liuweiseu/serendip6.git
```

2. Compile the code

```
cd serendip6/src
./do_make mro
```

3. Install SERENDIP6

```
sudo make install-mro
```

Configure ROACH2

1. Clone the python scripts for ROACH2 configuration

```
git clone -b italy_seti https://github.com/liuweiseu/roach2_control_script.git
```

2. Follow the instructions in README.md to set up the python2 environment.
3. Copy the bof file to ROACH2 board

```
cd roach2_control_script
scp bof/italy_seti1_v1.172.bof user@roach2_ip:/boffile
```

Note: The bof file directory on ROACH2 board could be on somewhere else. It depends on what version of rootfs you are using for ROACH2 board.

4. Upload the bof file to FPGA

```
(roach2_py2)roach2_control_script$ python mb.py --ip 10.0.1.168
Connecting to server 10.0.1.168 on port 7147...
ok
--------------------
Programming FPGA with  italy_seti1_v1.172.bof ...  done
Configuring spectrometer "u0" fft_shift, fft_shift=0xFFFF ...  done
Configuring spectrometer "u0" scale coefficients, gain=0x1000100 ...  done
Configuring spectrometer "u0" bit selection, bit_select=0x55 ...  done
Configuring spectrometer "u0" accumulation length, acc_len=100 ...  done
Configuring spectrometer "u0" fft_shift, fft_shift=0xFFFF ...  done
Configuring spectrometer "u0" scale coefficients, gain=0x1000100 ...  done
Configuring spectrometer "u0" bit selection, bit_select=0x55 ...  done
Configuring spectrometer "u0" accumulation length, acc_len=100 ...  done
Initializing xgbe0 fabric mac: 02:02:C0:A8:10:DD, ip: 192.168.16.221, port: 33333 ... done
Configuring xgbe0 destination IP and port 239.2.0.1:12345 ...  done
Initializing xgbe1 fabric mac: 02:02:C0:A8:10:DE, ip: 192.168.16.222, port: 33333 ... done
Configuring xgbe1 destination IP and port 239.2.0.2:12345 ...  done
Initializing xgbe2 fabric mac: 02:02:C0:A8:10:DF, ip: 192.168.16.223, port: 33333 ... done
```

```
Configuring xgbe2 destination IP and port 239.1.0.4:12345 ...  done
Initializing xgbe3 fabric mac: 02:02:C0:A8:10:E0, ip: 192.168.16.224, port: 33333 ... done
Configuring xgbe3 destination IP and port 239.1.0.3:12345 ...  done
Issue reset signal...
done
```

Note: You can use `python mb.py --help` to get more options.

The default values are:

- IP address of ROACH2: 192.168.100.128
- port: 7147
- bof file: italy_seti1_v1.172.bof

5. Check the ROACH2 status **(Optional, not necessary)**

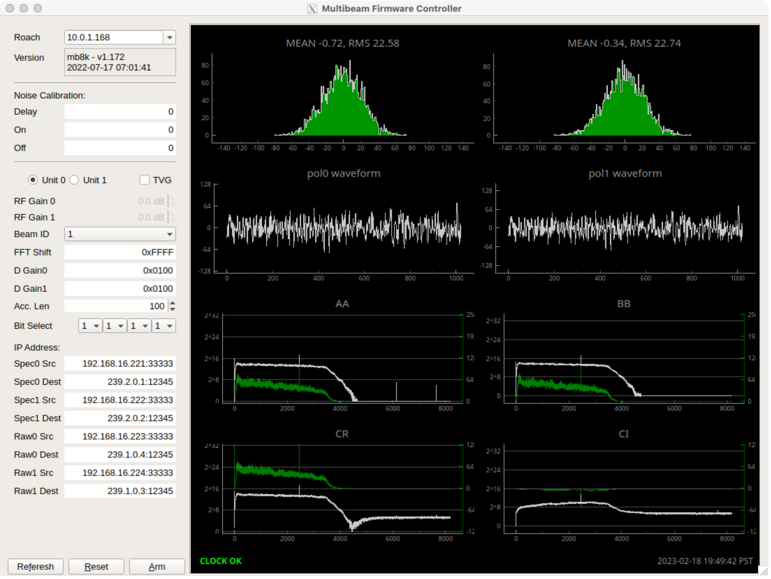
(**Please use `ssh -XY xx@xxx` to log into your server, or you will not be able to see the GUI.**)

```
(roach2_py2)roach2_control_script$ python mbc.py
```

You need to select the IP address of the ROACH2 board. If the IP doesn’t show up in the IP list, you need to edit **mbc.py**, and add the correct IP in “roach_list”

```
roach_list = ['192.168.100.128', '10.0.1.168','localhost']
```

If everything works well, you should see the following figures.



Multibeam Firmware Controller

Roach: 10.0.1.168
Version: mb8k - v1.172 2022-07-17 07:01:41

Noise Calibration:
Delay: 0
On: 0
Off: 0

● Unit 0 ○ Unit 1 □ TVG

RF Gain 0: 0.0 dB
RF Gain 1: 0.0 dB
Beam ID: 1
FFT Shift: 0xFFFF
D Gain0: 0x0100
D Gain1: 0x0100
Acc. Len: 100
Bit Select: 1 1 1 1

IP Address:
Spec0 Src: 192.168.16.221:33333
Spec0 Dest: 239.2.0.1:12345
Spec1 Src: 192.168.16.222:33333
Spec1 Dest: 239.2.0.2:12345
Raw0 Src: 192.168.16.223:33333
Raw0 Dest: 239.1.0.4:12345
Raw1 Src: 192.168.16.224:33333
Raw1 Dest: 239.1.0.3:12345

MEAN -0.72, RMS 22.58
MEAN -0.34, RMS 22.74

CLOCK OK
2023-02-18 19:49:42 PST

Start/Stop SERENDIP6

1. set environment variables

   You need to set three environment variables for SERENDIP6:

   - S6_DATA_DIR: data directory for SERENDIP6

- POL0_ETH: the 10GbE port for pol0
- POL1_ETH: the 10Gbe port for pol1

```
~$ export S6_DATA_DIR=/data01/serendip6_data
~$ export POL0_ETH=enp216s0f0
~$ export POL1_ETH=enp216s0f1
```

Note: “/data01/serendip6_data”, “enp216s0f0” and “enp216s0f1” are is the default values. If you’re happy with them, you don’t need to set the environment variables.

2. start SERENDIP6

Just run `s6_restart_mro.sh` to start SERENDIP6

```
~$ s6_restart_mro.sh
Data Path: /data01/serendip6_data
removing old semaphore, if any
rm: cannot remove '/dev/shm/sem.serendip6_gpu_sem_device_*': No such file or directory

Starting instance s6c/1
binding s6_pktsock_thread to enp216s0f1
numactl --physcpubind=16,17,18 --membind=0,1 hashpipe -p serendip6 -I 1 -o VERS6SW=0.8.0 -o VERS6GW=0.1.0 -o RUNALWYS=1 -o MAXHITS=2048 -o
Instance s6c/1 pid 646568

Starting instance s6c/2
binding s6_pktsock_thread to enp216s0f0
numactl --physcpubind=24,25,26 --membind=0,1 hashpipe -p serendip6 -I 2 -o VERS6SW=0.8.0 -o VERS6GW=0.1.0 -o RUNALWYS=1 -o MAXHITS=2048 -o
Instance s6c/2 pid 646569
Sleeping to let instances come up...
Resetting MISSEDPK count for s6c/1
Resetting NETDRPTL count for s6c/1
Resetting NETPKTTL count for s6c/1
Resetting MISSEDPK count for s6c/2
Resetting NETDRPTL count for s6c/2
Resetting NETPKTTL count for s6c/2
Turning on TESTMODE for /1
Turning on TESTMODE for /2
Turning on RUNALWYS for /1
Turning on RUNALWYS for /2
```

If SERENDIP6 starts up, you will see two hashpipe instances:

```
~$ ps -x
240550 pts/15   Sl+    0:14 hashpipe -p serendip6 -I 0 -o VERS6SW=0.8.0 ...
240551 pts/15   .l+    0:14 hashpipe -p serendip6 -I 1 -o VERS6SW=0.8.0 ...
```

You will also see data files created in the data directory

3. stop SERENDIP6

Just run `s6_stop_mro.sh` to stop SERENDIP6

```
~$ s6_stop_mro.sh
```

Check data file

Here is a simple script for check data file.

You can get it here:

```
git clone https://github.com/liuweiseu/seti_analysis.git
```

You need to change the file name in the .ipynb file

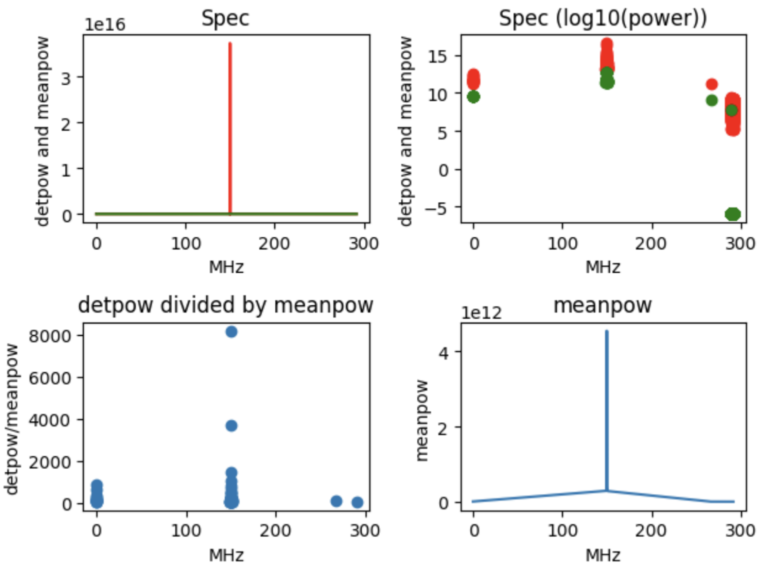
```
filename = '../data/serendip6_panoseti_sxp_1_1_20230217_154019.working'
```

Then you will see the metadata like this:

```
XTENSION= 'BINTABLE'           / FITS binary table
BITPIX  =                    8 / Binary data
NAXIS   =                    2 / 2-dimensional binary table
NAXIS1  =                    0 / width of table in bytes
NAXIS2  =                    0 / Number of rows in table
PCOUNT  =                    0 / size of special data area
GCOUNT  =                    1 / one data group (required keyword)
TFIELDS =                    0 / Number of fields per row
EXTNAME = 'MROSTATUS'          / name of this binary table extension
NMATRIX =                    1 / 1 dataset per row
COARCHID=                    0 / coarse channel ID
TIME    = '*       '           / unix time, decimal seconds
SOURCE  = '0406-127'
ONSOURCE=                    1
SITE    = 'Mc      '
RX_CODE = 'sxp     '
YEAR_DOY= '#2023-026/174535'
YEAR    =                 2023
DOY_UTC =                   26
UTC     =               174535
LO_FREQ =                8080.
TSYS    =            40.900002
XC      =              -331.09
YC      =              -306.92
Z1C     =                -4.51
Z2C     =               -53.14
Z3C     =                66.82
XA      =              -331.09
YA      =              -306.92
Z1A     =                 -7.7
Z2A     =               -48.77
Z3A     =                51.04
SUBMODE =                    0
RX_SUB  = 'sxp     '
SCU_STAT=                   63
EXTVER  =                    1 / auto assigned by template parser
```

You will also get a figure like this:

(A strong 150MHz signal is injected for test)


Spec
1e16
detpow and meanpow
MHz
Spec (log10(power))
detpow and meanpow
MHz
detpow divided by meanpow
detpow/meanpow
MHz
meanpow
1e12
meanpow
MHz