

Getting Started:

This document assumes you know how to set up and run beamforming for observing pulsars on the DE601 LCU.

Right now, LuMP is only available for recording data on the lofarN computers (lofar1, lofar2, lofar3, lofar4). I am waiting for some system administrative things to happen on the lofarNa computers before the software can run properly.

On the lofarN computers, as user observer, do

```
source /opt/lump/LOGIN.SH
```

or if you are a *csh person, do

```
source /opt/lump/LOGIN.CSH
```

This will set up your environment variables to run LuMP stuff.

Testing that the recording computer is receiving data:

On lofar1 you can run

```
sudo tcpdump -i eth1 udp port 4346
```

Getting basic pulsar information:

Suppose that you want to observe B0329+54. Run psrcat to get some basic information about this source.

```
psrcat -o short -nohead -nonumber -c "name raj decj p0 dm s400" B0329+54
```

```
B0329+54 03:32:59.3 +54:34:43.5 0.714520 26.83 1500.00
```

```
psrcat -x -c "P0 DM" B0329+54
```

```
0.714519699726 4.0e-12 26.833 1.0e-02
```

This gives you the RA and Dec of the source (03:32:59.3 +54:34:43.5), which you will need later. It also gives you the period (0.714519699726) and dispersion measure (26.833), which you might need later.

Recording data:

For recoding pulsar data in the LuMP format that can be read by dspsr (at least my hacked version of dspsr until Willem incorporates the patches I have sent to him into the main dspsr trunk), you run the command `Basic_LuMP_Recorder.py`

As an example, you might do

```
Basic_LuMP_Recorder.py --port=4346 --clock_speed=200 --num_beamlets=61 --physical_beamlet_offset=0 -
--station_name=Ef --RCUMODE=5 --physical_beamlet_array='[0:61]' --subband_array='[220:281]' --verbose --
duration=300 --filename_base=B0329+54_1
```

What does all of this mean? Well, you can always run `Basic_LuMP_Recorder.py --help` to get a detailed listing of all options available to the program. But for this quick guide, here is what is going on

`--port=4346` says to use port 4346, which is the port that DE601 sends data to lofar1 on

`--clock_speed=200` says that the station is in the 200 MHz clock mode

`--num_beamlets=61` says that you will record a total of 61 beamlets

`--physical_beamlet_offset=0` says that the starting beamlet for recording corresponds to beamlet 0 on the LCU

`--station_name=Ef` means that you are using DE601

`--RCUMODE=5` means that you are recording data from RCUMODE 5 (using HBAs, with the frequency range 110 to 190 MHz)

`--physical_beamlet_array='[0:61]'` means that this particular recorder will be listening to beamlets 0 to 61, in a python-ish notation. Note that this notation is different from the notation used by beamctl, where you would say 0:60. This piece of LuMP software is using python, and the notation for this program is always python notation. Note also that in most shells you will need to put quotes around the square brackets to make sure the shell does not try to interpret them.

`--subband_array='[220:281]'` means that this recorder will be recording subbands 220 to 281 (in python notation). Note that in most shells you will need to put quotes around the square brackets to make sure the shell does not try to interpret them.

`--verbose` says to turn on lots of message reporting

`--duration=300` says that you want to make a 300 second (5 minute) long observation. Because there is no start time specification, the recorder will start listening for data 1 or 2 seconds after you start this command. There is also an option to directly specify the start time of the recording. See below.

`--filename_base=B0329+54_1` says that the base output filename should be B0329+54_1 You should probably add on an `_1` (as here) for data recorded on lofar1, an `_2` for data on lofar2, and so on.

Processing data with dspsr:

This will record data to LuMP format files, which can be read in by dspsr. The LuMP format has a 4096 character text header at the beginning of the subband data files that you can read with `head -c 4096 filename`

The recording session will result in files with names

```

B0329+54_1.00.0000_raw
B0329+54_1.00.0001_raw
B0329+54_1.00.0002_raw
B0329+54_1.00.0003_raw
B0329+54_1.00.0004_raw
B0329+54_1.00.0005_raw
B0329+54_1.00.0006_raw
B0329+54_1.00.0007_raw
B0329+54_1.00.beamlets_raw
B0329+54_1.00.file_raw
B0329+54_1.00.info
B0329+54_1.00.subbands_raw
B0329+54_1.00.valid
B0329+54_1.01.0008_raw
B0329+54_1.01.0009_raw
B0329+54_1.01.000A_raw
B0329+54_1.01.000B_raw
B0329+54_1.01.000C_raw
B0329+54_1.01.000D_raw
B0329+54_1.01.000E_raw
B0329+54_1.01.000F_raw
...
B0329+54_1.06.003B_raw
B0329+54_1.06.003C_raw
B0329+54_1.06.beamlets_raw
B0329+54_1.06.file_raw
B0329+54_1.06.info
B0329+54_1.06.subbands_raw
B0329+54_1.06.valid

```

The ones you care about are the `B0329+54_1.???.0???`_raw files. These are the LuMP format data files. The other files contain extra information that you will probably not need at this time. Before you use dspsr, you need to fill in some extra information into the headers of the files, and correct for any residual timing offsets between files. To do this, run

```
LuMP_Pulsar_Cleanup.py --source_name=B0329+54 --source_RA=03:32:59.3 --source_Dec=+54:34:43.5 --
telescope=Effelsberg --filename_base=B0329+54_1.00 --filename_base=B0329+54_1.01 --
filename_base=B0329+54_1.02 --filename_base=B0329+54_1.03 --filename_base=B0329+54_1.04 --
filename_base=B0329+54_1.05 --filename_base=B0329+54_1.06
```

(I will improve the argument list for this program in the future.) This command will enter the name, RA, and Dec of the source into the data file headers, in case dspsr bothers to read this information. It also fudges the name of the telescope to Effelsberg, since dspsr refuses to accept LOFAR station names at this time.

Then, to process the raw data files into dspsr archive files, you might do

```
for i in B0329+54_1.*.0*_raw
do
dspsr -A -D 26.833 -c 0.714519699726 -L 10 -O $i.ar $i
done

psradd -R -o B0329+54_1.ar B0329+54_1.*.0*_raw.ar.ar
```

if you are using bash, or

```
foreach i (B0329+54_1.*.0*_raw)
dspsr -A -D 26.833 -c 0.714519699726 -L 10 -O $i.ar $i
end

psradd -R -o B0329+54_1.ar B0329+54_1.*.0*_raw.ar.ar
```

for csh. Look at the data with

```
pazi B0329+54_1.ar
```

Since you are probably a pulsar astronomer, you should know more about using dspsr than I do.

Recording more than 12 MHz using multipler recording computers:

Set up the beamctl stuff on the LCU. For eaxmple, to observe B0329+54 from 120 to 167 MHz, plus a couple of extra frequencies that are not contiguous, run on the LCU

```
beamctl --antennaset=HBA_JOINED --rcus=0:191 --rcumode=5 --beamlets=0:244 --
subbands=102:342,360,380,400,420 --anadir=0.929337,0.952579,J2000 --digdir=0.929337,0.952579,J2000&
```

Then, on lofar1 run

```
Basic_LuMP_Recorder.py --port=4346 --clock_speed=200 --num_beamlets=61 --physical_beamlet_offset=0 -
-station_name=Ef --RCUMODE=5 --physical_beamlet_array='[0:61]' --subband_array='[102:163]' --verbose --
start_date=2011-12-25T12:00:00Z --duration=300 --filename_base=B0329+54_1
```

on lofar2 run

```
Basic_LuMP_Recorder.py --port=4354 --clock_speed=200 --num_beamlets=61 --physical_beamlet_offset=61
--station_name=Ef --RCUMODE=5 --physical_beamlet_array='[61:122]' --subband_array='[163:224]' --
verbose --start_date=2011-12-25T12:00:00Z --duration=300 --filename_base=B0329+54_2
```

on lofar3 run

```
Basic_LuMP_Recorder.py --port=4350 --clock_speed=200 --num_beamlets=61 --
physical_beamlet_offset=122 --station_name=Ef --RCUMODE=5 --physical_beamlet_array='[122:183]' --
subband_array='[224:285]' --verbose --start_date=2011-12-25T12:00:00Z --duration=300 --
filename_base=B0329+54_3
```

and on lofar4 run

```
Basic_LuMP_Recorder.py --port=4358 --clock_speed=200 --num_beamlets=61 --
physical_beamlet_offset=183 --station_name=Ef --RCUMODE=5 --physical_beamlet_array='[183:244]' --
subband_array='[285:343,360,380,400,420]' --verbose --start_date=2011-12-25T12:00:00Z --duration=300 --
filename_base=B0329+54_4
```

At some point in the future, I hope to write a script to have all of this controlled automatically from the lofarx computer.

Note that in the above commands, I have used an explicit start time of noon on Christmas day. Please also note that the times are in UT, and the LCU and lofarN computers should all be running on UT time. Setting up the start time is important to get all of the data aligned to the same start boundary.

Processing data from multiple recorders:

Copy all of the data over to lofarsrv. There, as observer, you will need to do

```
source ~observer/anderson/LOGIN.SH
```

or if you are a *csh person, do

```
source ~observer/anderson/LOGIN.SH
```

This sets up the environment so that you can run stuff on lofarsrv.

Then, fix the headers because dspsr doesn't understand LOFAR station names:

```
LuMP_Pulsar_Cleanup.py --source_name=B0329+54 --source_RA=03:32:59.3 --source_Dec=+54:34:43.5 --
telescope=Effelsberg --filename_base=B0329+54_1.00 --filename_base=B0329+54_1.01 --
filename_base=B0329+54_1.02 --filename_base=B0329+54_1.03 --filename_base=B0329+54_1.04 --
filename_base=B0329+54_1.05 --filename_base=B0329+54_1.06 --filename_base=B0329+54_2.00 --
filename_base=B0329+54_2.01 --filename_base=B0329+54_2.02 --filename_base=B0329+54_2.03 --
filename_base=B0329+54_2.04 --filename_base=B0329+54_2.05 --filename_base=B0329+54_2.06 --
filename_base=B0329+54_2.00 --filename_base=B0329+54_2.01 --filename_base=B0329+54_2.02 --
filename_base=B0329+54_2.03 --filename_base=B0329+54_2.04 --filename_base=B0329+54_2.05 --
filename_base=B0329+54_2.06 --filename_base=B0329+54_4.00 --filename_base=B0329+54_4.01 --
filename_base=B0329+54_4.02 --filename_base=B0329+54_4.03 --filename_base=B0329+54_4.04 --
filename_base=B0329+54_4.05 --filename_base=B0329+54_4.06
```

Again, I hope to improve the argument parsing in this program to clean things up in the future and make this easier.

Then, process the data files in dspsr

```
for i in B0329+54_?.*.0*_raw
do
dspsr -A -D 26.833 -c 0.714519699726 -L 10 -O $i.ar $i
done
```

Processing data from multiple recorders:

```
psradd -R -o B0329+54.ar B0329+54_?.*.0*_raw.ar.ar
```

or

```
foreach i ( B0329+54_?.*.0*_raw)  
dspsr -A -D 26.833 -c 0.714519699726 -L 10 -O $i.ar $i  
end
```

```
psradd -R -o B0329+54.ar B0329+54_?.*.0*_raw.ar.ar
```

Then, do

```
pazi B0329+54.ar
```

There you are, 48 MHz worth of pulsar data.