Breakthrough Listen Raw Data Format

Nick Joslyn – June 12, 2018

Breakthrough Listen (BL) aims to make data as publicly available as possible. In addition to loaning out your personal computer's hardware as part of the SETI@home project (https://setiathome.berkeley.edu/), you can actually access the data yourself at this link:

https://breakthroughinitiatives.org/opendatasearch

BL hardware is capable of extremely fast sampling over a wide range of frequencies. As a result, the data files are correspondingly large. A common technique is to compress data into a filterbank file for later analysis. However, this documentation relates to the uncompressed RAW file, listed as a 'baseband data' file type on the data archive.

Raw Data -- Overview

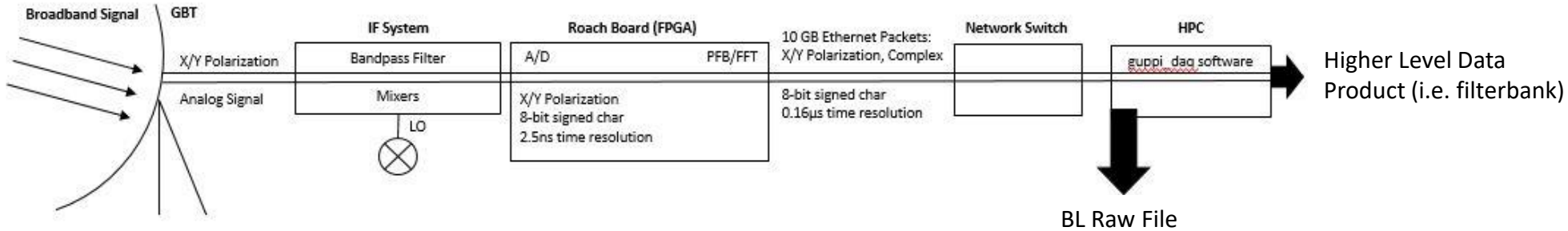
The Green Bank Ultimate Pulsar Processing Instrument (GUPPI) was one of the main backends for the Green Bank radio telescopes. It was employed to study pulsars, and the data was stored in the GUPPI raw data format. Specifically, the format stores channelized, complex voltages detected by the telescope*. The format of the GUPPI data file is a **.raw** file. The .raw files are binary -- a collection of bytes to be interpreted by the user.

The GUPPI Raw format is the inspiration for the Breakthrough Listen Raw Data Format. Chief Engineer Dave MacMahon improved upon the GUPPI format to produce the BL Raw Data Format currently in use. His quick overview is provided at: https://github.com/UCBerkeleySETI/breakthrough/blob/master/doc/RAW-File-Format.md. Additionally, Steve Croft posted some documentation here: https://github.com/UCBerkeleySETI/breakthrough/blob/master/GBT/waterfall.md. This PPT serves to amalgamate important information from these sources, figures of my creation, and thoughts from working with the data.

Overall, the files are about 17.2GB corresponding to approximately 22.8 seconds of observation. The BL Raw Data Files are read as a series of header data units (HDUs). An HDU consists of a header followed by data. More specifically, we interpret the header section as ASCII text and the data as a raw binary data segment (signed integers). The data section is referred to as a "block."

---

* A crude schematic and explanation of where the raw file fits in a radio telescope pipeline: The radio telescope receives radiation from the environment. The voltages detected are sent through a bandpass filter (with mixers) to identify a frequency range. Next, a (real to complex) Polyphase Filter Bank Fast Fourier Transform is applied to coarse channelize the data (i.e. large frequency bins) over small time increments (maintaining time resolution). The result is a "time packet" which holds the information of frequency bin vs amplitude (and phase information). Time packets are on the order of nanoseconds; many time packets are accumulated during an observing period and dumped to disk before more processing.



Raw Data – Header Section

The header section contains metadata to describe the following data block. These descriptions are given in “cards” (sometimes referred to as a record). Each card is an 80 character ASCII segment. The first 8 characters of each segment represent the keyword. If the keyword is greater than 8 characters, the keyword is truncated. If the keyword is less than 8 characters, it will be padded with spaces. The 9th and 10th characters are an equals sign and a space character, respectively. The remaining characters are filled with the value of the keyword (and spaces). The syntax is as follows (arbitrary example):

KEYWORD = 32
SCNDKYWD= ‘BL’

As expected, the “=“ is the ninth character. Even though “KEYWORD” is only seven characters, an extra space was added. Analogously, the term “SECOND KEYWORD” would exceed the character limit; so, it was renamed to fit in the character constraints. For both examples, the remaining characters in each line are space characters (recall that each card is 80 ASCII characters). The actual header file uses standardized keywords. There are many keywords declared (telescope, center frequency, observed number of channels, etc.). These will be described in more detail in the following slide.

A final note about the header section is its completion. The header section contains information to inform the user when the header file is finished and the data block is about to start. This is accomplished with the keyword *END* followed by 77 space characters. Following the space characters, the data block begins, with one added complexity.

BL uses Direct I/O. Direct I/O is a very fast file writer, but adds some requirements. Direct I/O requires that disk writing be a multiple of 512 bytes. Thus, a keyword called “DIRECTIO” in the header indicates a padding offset before the data begins. If DIRECTIO is 0 or unlisted, then there is no padding needed. If specified to a non-zero value, then the user must parse from their location until the next location corresponding to a multiple of 512 bytes. The padding byte values are unimportant, but must be skipped before reaching the data block.

BL Raw Data Format – Header Detail

```
BACKEND = 'GUPPI   '
DAQCTRL = 'start   '
DAQPULSE= 'Wed Dec 13 16:53:21 2017'
DAQSTATE= 'record  '
NBITS   =                    8
OFFSET0 =                  0.0
OFFSET1 =                  0.0
OFFSET2 =                  0.0
OFFSET3 =                  0.0
BANKNAM = 'BLP05   '
TFOLD   =                    0
DS_FREQ =                    1
DS_TIME =                    1
FFTLEN  =                  512
CHAN_BW =           -2.9296875
BANDNUM =                    0
NBIN    =                  256
OBSNCHAN=                   64
SCALE0  =                  1.0
SCALE1  =                  1.0
DATAHOST= 'blr2-1-10-5.gb.nrao.edu'
SCALE3  =                  1.0
NPOL    =                    4
POL_TYPE= 'AABBCRCI'
BANKNUM =                    5
DATAPORT=                60000
ONLY_I  =                    0
CAL_DCYC=                  0.5
DIRECTIO=                    1
BLOCSIZE=            134217728
ACC_LEN =                    1
OBSERVER= 'Emilio Enriquez'
CAL_MODE= 'OFF     '
PROJID  = 'AGBT17B_999_75'
OVERLAP =                    0
OBS_MODE= 'RAW     '
CAL_FREQ= 'unspecified'
DATADIR = '/datax/dibas'
OBSFREQ =        2120.21484375
PFB_OVER=                   12
SCANLEN =                300.0
PARFILE = '/opt/dibas.new/etc/config/example.par'
OBSBW   =               -187.5
FD_POLN = 'LIN     '
SCALE2  =                  1.0
BINDHOST= 'eth4    '
TELESCOP= 'GBT     '
NRCVR   =                    2
PKTFMT  = '1SFA    '
```

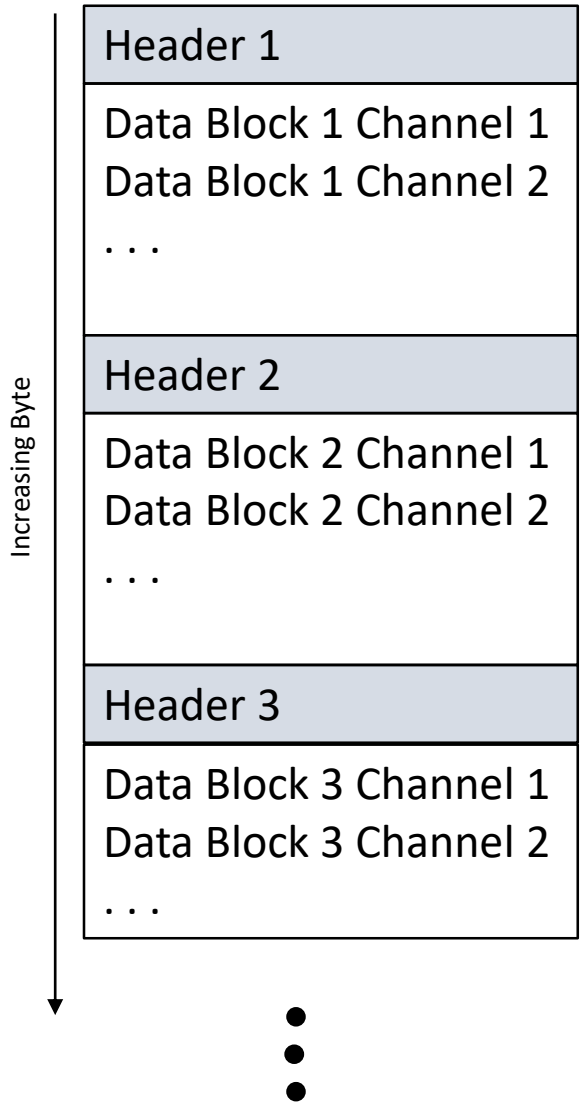
```
TBIN    = 3.41333333333333E-07
BASE_BW =                1450.0
SRC_NAME= 'OUMUAMUA'
CHAN_DM =                   0.0
FRONTEND= 'Rcvr2_3 '
BMAJ    =  0.08588421286626495
BMIN    =  0.08588421286626495
TRK_MODE= 'UNKNOWN '
RA_STR  = '23:21:39.7200'
RA      =              350.4155
DEC_STR = '+07:53:1.6800'
DEC     =                7.8838
LST     =                 79503
AZ      =              145.1531
ZA      =               35.1026
SCAN    =                    11
STT_SMJD=                 78802
STT_IMJD=                 58100
STTVALID=                     1
DISKSTAT= 'waiting '
NETSTAT = 'receiving'
PKTIDX  =                     0
DROPAVG =           0.000114062
DROPTOT =              0.790244
DROPBLK =                     0
PKTSTOP =              27459584
NETBUFST= '1/24    '
STT_OFFS=                     0
SCANREM =                   0.0
PKTSIZE =                  8192
NPKT    =                 16384
NDROP   =                     0
END
```

Sample image of the header section of a BL Raw Data File. This is an observation of Oumuamua from the online Breakthrough database. Due to the vertical limitations, the section was split into two pictures. In reality, this split does not exit. Immediately after the PKTFMT card, the TBIN card begins. These sequences of ASCII characters reveal information about the following data block. Some are self-explanatory. Important keywords are explained on the right.

Keyword	Description
OBSFREQ	The central frequency observed [MHz]
OBSBW	Observing Bandwidth (negative sign indicates spectral flip) [MHz]
OBSNCHAN	Number of channels
NPOL	Number of polarizations (if 4, including real/complex)
NBITS	Number of bits per real/imaginary value
TBIN	Sample period within a channel [seconds]
OVERLAP	0 – In GUPPI this indicated the number of samples repeated between the end of one block and start of another block
BLOCSIZE	The size of the data block in bytes (i.e. indicates when the next header file will start)
DIRECTIO	-If non-zero -> The user will need to include padding bytes up until the next multiple of 512 (as described on previous slide) -If 0 or unlisted -> No padding bytes
BACKEND	BL Raw files are set to ‘GUPPI’

BL Raw Data Format – General Design

Byte Domain



The byte domain represents how the computer sees the BL raw data. The BL raw data, in its most simplistic form, is a stream of bytes to be interpreted as HDUs.

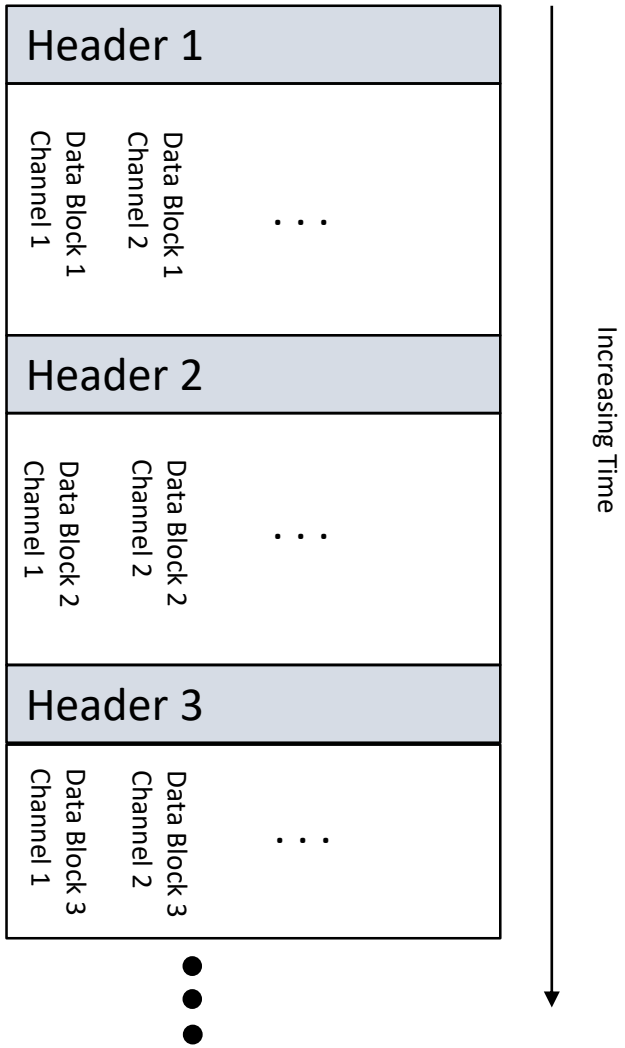
The time domain represents the reality of the stream of bytes. The channels in each data block are the same range of time. However, the channels come in at different points in the byte stream.

Essentially, a data block is the following (row-major) array:

[Channel, Time Sample, Polarization (including real/complex)]

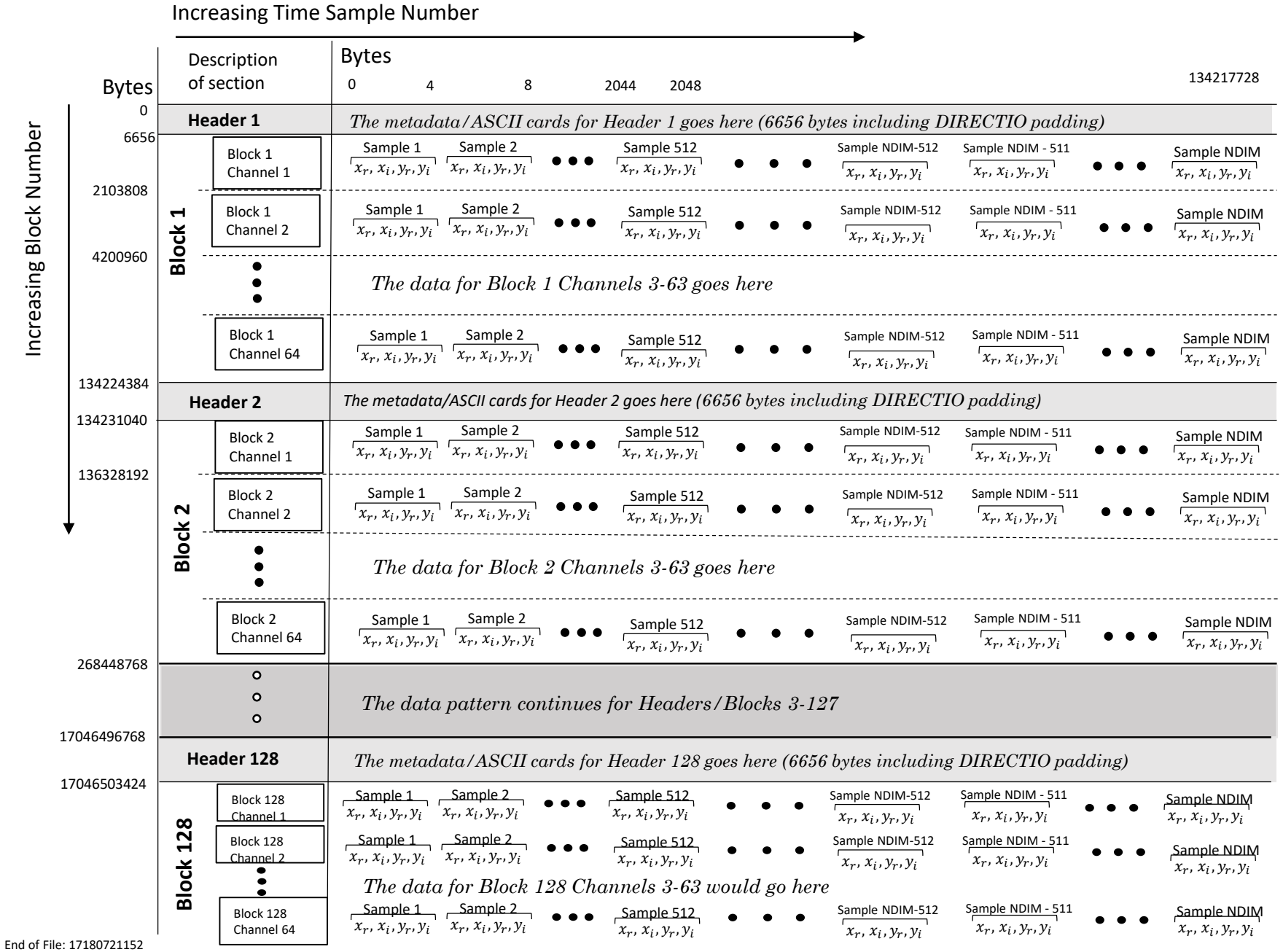
The next slide will go into this in more detail.

Time Domain**



**Note, the time domain is fictitious to the computer. The computer cannot see overlap in bytes. The time domain is merely a good illustration of what the data actually means physically.

BL Raw Data Format – Specific Design



For argument (and this is reasonable based on the values specified in the header sections of BL files), we assume the header file is 6656 bytes (with DIRECTIO padding to the nearest multiple of 512 bytes), OBSNCHAN = 64, NPOL = 4, NBITS = 8, BLOCSIZE = 134217728, and there are 128 HDUs.

Additionally, NDIM is the number of time samples per channel per block. The value is calculated by the following:

$$NDIM = \frac{BLOCSIZE}{OBSNCHAN * NPOL * \left(\frac{NBITS}{8}\right)}$$

In our example, NDIM = 524288. By multiplying by TBIN (=3.4X10^-7), the time observed during each block is 0.178 seconds.

The total bytes in a common BL raw file is 17,180,721,152. Thus, the size is approximately 17.2 GB corresponding to 22.8s.

Interpreting the BL Data File

Recall that BL raw files are binary. The data is a series of bytes (8-bit, 1s and 0s). However, the file is formatted in a very precise manner. The bytes are ordered: Header 1, Block 1, Header 2, Block 2, . . . , Header 128, Block 128. For each block, the bytes are ordered Channel 1, Channel 2, . . ., Channel 64. For each channel, the bytes are ordered Sample 1, Sample 2, . . ., Sample NDIM. For each time sample, the bytes are ordered x polarization real, x polarization imaginary, y polarization real, y polarization imaginary.

Because of the strict requirements of the byte ordering, we can think of the stream of bytes as an array of bytes. By imagining an array, not only can we gain a more intuitive feel of the data, but also we can create the visual representation on the left.

Understanding the Visual Representation

First, let's identify the axes. Our axis dimensions are in bytes. From left to right, we indicate the number of bytes as our time sample increases. From top to bottom, we indicate the number of bytes as our channel/block number increases.

The bytes of data in the .raw file correspond to the following path:

1. Start at Byte 0. The next 6656 bytes correspond to Header 1 + DIRECT I/O Padding
2. Arrive at Block 1: The first four bytes indicate the two voltage values (real and complex) for each polarization (x and y) of time sample 1 of Channel 1. The next four bytes indicate the two voltage values (real and complex) for each polarization (x and y) of time sample 2 of Channel 1. Continue through until time sample NDIM. Then, Channel 2 begins, and the pattern continues until the end of Channel 64.
3. Now, byte 134224384 begins Header 2.
4. The previous steps are repeated until the end of the file (i.e. Block 128)

Notes

Essentially, the string of binary data can be tracked by reading this visual representation like a book. Begin at the top left and read from left-to-right. The bytes along the top indicate how many bytes have been passed during that channel, and the bytes along the side indicate, collectively how many bytes have been passed. Thus, the byte number of any location can be determined by adding the two axis values.

BL Raw Data Format – Additional Links

- David MacMahon's Description:
  - https://github.com/UCBerkeleySETI/breakthrough/blob/master/doc/RAW-File-Format.md
- Steve Croft's Description:
  - https://github.com/UCBerkeleySETI/breakthrough/blob/master/GBT/waterfall.md

- The predecessor:
  - GUPPI
    - https://safe.nrao.edu/wiki/bin/view/CICADA/NGPPProjectDocuments
  - GUPPI Technical Paper
    - https://safe.nrao.edu/wiki/pub/Software/CicadaSoftwareStatus/SPIE08-GUPPI-preprint.pdf
  - GUPPI User's Guide
    - https://safe.nrao.edu/wiki/bin/view/CICADA/GUPPiUsersGuide