Observing with the Station (LCU)

The chapter describe the different observation methods when directly interfacing with the station, rather than using any automated scripts.

The order in which topics are disccused also reflects the order in which commands need to be called to correctly configure the station for an observation.

Software Levels

In international mode, the station has 4 software levels, but only 3 of which are used extensively. Levels are applied consecutively, so entering software level 3 will turn on any services that are normally turned on by software level 1 and 2.

Software level is described and configured with the `swlevel` command. The two main forms of the command you will use are `swlevel <level>` and `swlevel -S`. Examples of this command are given below.

```
user1@lcu$ # Get the current software level
user1@lcu$ swlevel -S
0
user1@lcu$ # Go to software level 2 for all-sky imaging or antenna statistics collection
user1@lcu$ swlevel 2
<start-up messages>
user1@lcu$ # Go to software level 3 for beamformed observations
user1@lcu$ swlevel 3
<start-up messages>
```

In some cases, there may be an error during startup, or a service may crash during an observation. If a critical service has been effected, the station will change the software level to be in n invalid state, normally represented by your intended software level multiplied by -1. So if you were in software level 3 and an error occurred, the swlevel -S command will describe the station as being in level -3. To resolve this issue, you can either go to level 0 and back to your target level, or attempt to re-initialise the target level. Often, going to level 0 and back is the safest option and will resolve more issues, though at the cost of time.

At I-LOFAR, a cycle from level 0 to level 3 typically takes between 2 and 3 minutes. Consider this delay when planning your observations.

Software Level 0 (Off)

Software level 0 is the default mode the station is in after handover, and should be returned to before hand-over. In this state, all of the RSP, beaforming and calibration services are stopped and the station is inactive. Between observations the station is sometimes returned to software level 0 to ensure there are no rogue beams, statistic collectors, etc, still running from a previous observation so that the station can be brought up in a fresh state for the next observer.

Software Level 1 (Station Monitoring)

Software level 1 initialises some low-level daemons and the station monitoring and control software. Observations cannot be performed in this mode.

Software Level 2 (Hardware Initialised)

Software level 2 loads images onto the RSPs, enables the RCUs and performs some other low-level operations to prepare the station. While correlation statistic observations should be taken in this mode (XST, ACC), no beamformed observations can be performed in this mode.

Software Level 3 (Software Initialised)

Software level 3 starts the beamforming and calibration services. This is the default mode the station should be in while performing any beamformed observations (any non-XST/ACC observations, though attempting to take these will generate output files, but the correlations may be applied to post-beamforming delayed voltages).

Observing Bit Mode

There are 3 available bit modes available while observing with the station. 4, 8 and 16-bit, each of which describe the size of the signed word of each beamlet sample. 8-bit is the standard mode used for observations with I-LOFAR.

Adjusting the bit mode of the station varies the number of beamlets (pointings, frequencies) that can be formed at a given time, allowing for a trade off between digitisation accuracy and observing bandwidth.

Bitmode	Beamlets	Bandwidth (MHz)
16	244	47.65625
8	488	95.3125
4	976	190.625

The bitmode can be set using the `rspctl` command and the `--bitmode` flag, as demonstrated below.

```
user1@lcu$ rspctl --bitmode=16
user1@lcu$ rspctl --bitmode=8
user1@lcu$ rspctl --bitmode=4
```

4-bit Mode

4-bit mode is rarely used as unlike 8-bit mode, during the reduction operation from 8-bit to 4-bit data is not fully reduced to fit in the -8,7 range, resulting in samples being clipped. This can be compensated for by increasing the RCU attenuation, but requires analysis on a source-by-source basis due to the highly variable sky temperature at low frequencies. Data produced in this mode needs to be extracted in a special way, as each sample takes up the upper or lower half of a single byte.

When used correctly, it can allow for an extremely large sky coverage, or observations across multiple observing modes, at the cost of some sensitivity.

8-bit Mode

8-bit mode is the standard for all HBA and most LBA observations at I-LOFAR, offering up a 95MHz bandwidth for observers to cover an entire observing mode with the 200MHz clock (excluding parts of the Nyquist zones at multiples of 100MHz or 80MHz, depending on your clock mode).

16-bit Mode

16-bit mode is sometimes used for LBA observations to slightly increase the sensitivity at lower frequencies. This is often a useful trade-off due to the significant amount of RFI present below 30MHz from ionospheric reflections and local sources (AM radio, FM radio about 80MHz, etc, making sizeable fractions of the bandwidth unusable).

Antenna Sets, Subbands and RCU Bands (Modes)

Antenna Sets

In international mode, we recommend using either the `LBA_OUTER` or `HBA_JOINED` antenna sets, depending on the part of the instrument you intend to use. The remaining antenna sets are mostly used in the core and remote stations depending on the needs of observers to full out different parts of the UV-plane when observing with multiple stations

`LBA_OUTER` can be used with bands 10_90 or 30_90, while `HBA_JOINED` can be used with 110_190, 170_230, 210_250. The properties of each of these bands are described below.

The choice of antenna-set and band in your `beamctl` command, with the `--antennaset` and `--band` flags will chose how the station operates.

RCU Band	RCU Clock (MHz)	Lower Frequency (MHz)	Upper Frequency (MHz)	Subband Width (MHz)
10_90	200	0	100	0.1953125
30_90	200	0	100	0.1953125
110_190	200	100	200	0.1953125
170_230	160	160	240	0.15625
210_250	200	200	300 (~260 effective)	0.1953125

RCU Bands (Modes)

The RCU band (formally known as RCU mode, now referred to as 'bands', configured via the --band flag) allows you to set the observing window for a given set of RCUs. While the band of each antenna polarisation can be configured to be independent on any other antenna (allowing for configurations such as KAIRA's Mode 3-5-7), we recommend only using a single observing band and use multiple observing epochs to see a source over multiple modes to allow for the use of the entire antenna set to maximise the sensitivity and keep a reasonable and predictable beam shape for each observation.

The name of each band roughly describes the frequencies at which the electronics will have a negligible effect on the observed data. Outside of these ranges, the Nyquist zone or other filters may suppress emission at certain frequencies, even if they are available at specific subbands.

Mode 0, 1, 2

These modes are the OFF, Low-band Low (200MHz) and Low-Band Low (160MHz) observation modes, they should not be be used on international stations and are meant to selectively use different parts of the array in the core and remote stations.

Mode 3 (Band '10_90')

Mode 3 is the standard LBA observing mode, using the entire set of LBA antenna and the 200MHz clock, covering 0MHz to 100MHz.

Mode 4 (Band '30_90')

Mode 4 enables observation with the entire set of LBA antenna and the 200MHz clock, with an additional high-pass filter to reduce the effects of bright signals below 30MHz, but allowing for observations from 0MHz to 100MHz.

Mode 5 (Band '110_190', "HBALo")

Mode 5 is the standard HBA observing mode, enables observing with the HBA tiles and the 200MHz clock covering 100MHz to 200MHz.

Mode 6 (Band '170_230')

Mode 6 enables observing with the HBA tiles and the 160MHz clock, covering 160MHz - 240MHz.

Mode 7 (Band '210_250', "HBAHi")

Mode 7 enables observing with the HBA tiles and the 200MHz clock. It has a reduced bandwidth compared to the other modes as a result of significant amounts of RFI and antenna attenuation being present above ~240MHz, but allows for observations from 200MHz to 250MHz.

Subbands

A subband is 1 out of 512 slices created by the polyphase filterbank out of the total bandwidth available to a given observing band. The total bandwidth available is either 100MHz or 80MHz depending on your use of the 200MHz or 160MHz clock (refer to the table above or text below).

When trying to calculate the subbands you are interested in,you will need to consider the frequencies of interest and the chosen observing band. For example, while only 0-100 MHz is available to the LBAs, the HBA can observe either 100-200, 160-240, or 200-260MHz at one point in time. As a result, you should

- Chose your RCU band based on the frequencies of interest, and determine the lowest frequency available to you
- Find the difference between the lowest frequency and your target frequency, and divide the resulting value by the subband width (the RCU clock value divided by 1024), rounded downwards to encapsulate the target frequency, and you have one of your subband limits.
- Repeat for a chosen upper frequency, and ensure that your bitmode allows you to allocate enough beamlets (see previous chapter)

Overall, the subband calculation roughly follows this equation

```
target_subband = (target_frequency - band_base_frequency) / (band_rcu_clock / 1024)
target_subband = round_down(target_subband)
```

As some example calculations,

Frequency (MHz)	Band	Base Frequency (MHz)	RSP Clock (MHz)	Sub-band Calculation	Subband
25	10_90	0	200	(25 - 0) / (200 / 1024)	128
125	110_190	100	200	(125 - 100) / (200 / 1024)	128
185	170_230	160	160	(185 - 160) / (160 / 1024)	160
225	210_250	200	200	(225 - 200) / (200 / 1024)	128

Beamforming the Station

Beamforming is the process by which the station electronics collect the antenna signals and delays the signals to allow them to be coherently summed to point at a certain point in the sky. Depending on your bit-mode, you can control between 244 and 976 of these beams, each of which can take on a unique frequency and pointing direction.

Typically, all beams are pointed in the same direction, with the same antenna-set, and only differ in the selected frequencies. While other pages will discuss why/how we can deviate from that choice, this page will focus on observing a single mode, in bit-mode 8 (to use most of the available bandwidth in a given mode).

General Beamforming

In order to create a beam, you will need to execute a `beamctl` command. This command will require a spatial configuration, a spectral configuration and an antenna configuration. Roughly, a single beamctl command will look like either of the following stencils. The antenna sets, bands and subband components of these commands have been described elsewhere in this chapter.

```
user1@lcu$ # Observe with the LBAs
user1@lcu$ beamctl --antennaset=LBA_OUTER \
    --rcus=$rcus \
                --band=$bandLBA  \
                --beamlets=$beamletsLBA \
                --subbands=$subbandsLBA \
                --digdir=$pointing
user1@lcu$
user1@lcu$ # Observe with the HBAs
user1@lcu$ beamctl --antennaset=HBA_JOINED \
    --rcus=$rcus \
                --band=$bandHBA  \
                --beamlets=$beamletsHBA \
                --subbands=$subbandsHBA \
                --anadir=$pointing \
                --digdir=$pointing
```

Numeric Lists in LOFAR commands

The inputs to LOFAR commands can contain numeric lists. These lists use both semi-colon indexing to describe a range of inclusive values (e.g., 0:2 represents 0,1,2) and command separated values or lists to refer to multiple separate groups (e.g., 0,1:6,9). Spaces cannot be included in these groups.

When attempting to exclude values, be sure you aren't accidentally including them at the start/end of a range.

RCUs

RCUs represent the LBA antenna or HBA tiles you intend to use for your observations. As of writing, I-LOFAR currently has 1 malfunctioning LBA and 2 malfunctioning HBAs. reaching out to observers will give you the current definitive list of the antenna you should be using for your observations.

When flagging a given antenna or tile, it is recommended to always flag both polarisation to prevent biases from excess intensity in one polarisation than another when beamforming (e.g., you could introduce a +2% bias to Stokes V by only flagging one polarisation).

Beamlets

The allocated beamlets can be anywhere on the range allowed by your chosen software level (e.g., to allocate 200 beamlets in 16-bit mode, I can allocate them at 0:199 or 20:219). The selected beamlets will be used in set ascending order when producing BST data, or will be allocated as-given when transferring the data with CEP packets to your recording node (e.g., in 16-bit mode beamlets 0 - 60 will be transported on port 0, 61 - 121 will be transferred o port 1, etc.).

Your count of beamlets in each given beamctl command must match the amount of subbands used in each command. Beams will fail to be allocated if you attempt to use the same beamlets between multiple simultaneous beamctl commands, or if there are insufficient beamlets available (e.g., allocating more than 244 in 16-bit mode).

As an example, any of the following beamlet/subband pairings are allowed.

```
# Non-simultaneous beams
--beamlets=0:19 --subbands=0:19
--beamlets=0:19 --subbands=20:39
--beamlets=20:39 --subbands=0:19

# Simultaneous beams
--beamlets=0:19 --subbands=0:19 // --beamlets=20:39 --subbands=0:19
--beamlets=0:19 --subbands=0:19 // --beamlets=20:39 --subbands=20:39
```

Pointing (anadir/digidir)

Pointing is typically specific in right ascension and declination in the J2000 (ITRS) coordinate space, but any casacore-supported coordinate system is accepted by the station (https://casacore.github.io/casacore-notes/233.html#x1-23000A.2.3), such as SUN, MOON or JUPITER.

Depending on the coordinate basis, the input value differs. By default, for J2000, the first value ranges from 0 to 2pi in radians, covering 24 hours of right ascension, while the second range from -pi/2 to pi/2 in radians, covering 180 degrees of declination (overall, `[0.0, 6.283],[-1.571, 1.571],J2000`). For other coordinate systems, the inputs should be radians relative to a given source, but typically you can keep the values as `0.0,0.0,BASIS`.

LBAs

The LBAs allow for beams to be configured in any direction from 0MHz to 100MHz. Unlike the HBAs, they do not have any limitations that limit the spatial locality of the chosen beams on the sky.

Unlike HBAs, LBAs only require the `digidir` pointing component of the `beamctl` command, as they do not have analogue beamformers that need to be pointed prior to digital beamforming.

HBAs

Unlike the LBAs, the HBAs have an analogue beamformer to combine each of the 16 antenna in each tile into a coherent beam prior to any digitisation and electron beamforming. As a result, the HBAs require an extra configuration parameter, --anadir, and the chosen beams must be kept close to each other to prevent signal loss due to falloff in the analogue beam.

It is best to think of beaming for HBAs as a two step process. Firstly, the analogue beamformer limits our field of view to that of a single tile (~15deg.), and then the digital beamformer further limits the field of view to that of the selected tiles in the array (~0.5-2deg for the entire array). If the digital beam is not in the main lobe of the analogue beam, there will be a signfiicant degradation in signal quality.

The HBAs currently do not have any extra RFI filtering modes (potentially will be present in LOFAR2.0 to reduce the effects of DAB radio), though have three observing modes: HBALo (mode 5, band '110_190'), HBAHi (mode 7, band '210_250')

General Comments

It is recommended to leave a 30-60 second gap between starting a beam and recording the data, especially for the first observation of a run and for sequential HBA observations. This gives sufficient time for the antenna to power on and beamforming electronics to be fully initialised. Recording directly after a beam is allocated will often result in empty packets being sent to the

recording node, followed by a few seconds of noisy output as the beam settles on the correct point in the sky.

Example Observations (Currently only HBA)

Combining everything found in this chapter, we can provide an example observation. This code block describes initialisation of the station, beamforms using a selected set of RCUs on the I-LOFAR default subbands, then shuts down the station when the observation is complete. Overall we generally use

- A swlevel command to enter level 2,
- A pre-amble block to configure the RSPs to required bitmodes and other states,
- A swlevel command to enter level 3,
- A (or multiple) beamctl commands to perform a block of observations, each of which is manually killed after a specified beam length (beamctl does have a duration flag, but the process does not exit after the beam expires)
- A command to return the station to swlevel 0 for the next user

```
# Observation starting 2021-07-14T06:57, recording to beign at 2021-07-14T07:00
# 2021-07-14T07:00 - 2021-07-14T07:29 ;□]0139+3310 [0.43613087285590374, 0.5867200623795394, 'J2000']


bash ./sleepuntil.sh 20210714 065700
echo 'Initialising: SWLEVEL 2'
eval swlevel 2

# Using a preamble that has been passed down through the ages.
# I honestly cannot tell you the benefits / downsides of most of these rspctl commands.
rspctl --wg=0
sleep 1
rspctl --rcuprsg=0
sleep 1

# Swap to 8-bit mode to allow for use of the full bandwidth
rspctl --bitmode=8
sleep 1
killall beamctl
sleep 3
echo 'Initialising: SWLEVEL 3'
```

```
eval swlevel 3
sleep 2

# Ensure the SEREDES splitter is disabled and datastream is on (should not be needed, just encase
rspctl --splitter=0
sleep 1
rspctl --datastream=1
sleep 3



rcus='0:83,86:159,162:191'
pointing='0.4361308728559037 4,0.5867200623795394,J2000'
beamctl --antennaset=HBA_JOINED --rcus=$rcus --band=110_190  --beamlets=0:487 --subbands=12:499 --
anadir=$pointing --digdir=$pointing &
bash sleepuntil.sh 20210714 072910
killall -9 beamctl


swlevel 0
```

Checking Antenna Spectra

This is typically performed in either `swlevel 2` or `swlevel 3`, with the methodology differing between the modes. This is a sample of commands used while in `swlevel 3`,

```
user1@lcu$ # Initialisation
user1@lcu$ swlevel 3
user1@lcu$ rspctl --bitmode=8
user1@lcu$
user1@lcu$ # Beamforming the zenith to use RCUs
user1@lcu$ # LBA mode 3
user1@lcu$ beamctl --antennaset=LBA_OUTER --band=10_90 --rcus=0:191 --subbands=0:487 --
beamlets=0:487 --anadir=0,0.7853982,AZELGEO --digdir=0,0.7853982,AZELGEO
user1@lcu$ # HBA mode 5
user1@lcu$ beamctl --antennaset=HBA_JOINED --band=110_190 --rcus=0:191 --subbands=0:487 --
beamlets=0:487 --anadir=0.0.7853982,AZELGEO --digdir=0,0.7853982,AZELGEO
user1@lcu$ # HBA mode 7
user1@lcu$ beamctl --antennaset=HBA_JOINED --band=210_250 --rcus=0:191 --subbands=0:487 --
beamlets=0:487 --anadir=0,0.7853982,AZELGEO --digdir=0,0.7853982,AZELGEO
user1@lcu$
user1@lcu$ # Shutdown
user1@lcu$ swlevel 0
```

After initialisation, `rspctl --stati --select rcuN:rcuM,rcuA:rcuB` can be run in a separate shell (or the same shell if the `beamctl` commands are run in the background) and will plot the SST data for visual inspection, for a given range of RCUs. Using the `beamctl` method, RCUs not provided to a beam are not plotted by default.

If an antenna spectrum is looking suspicious, the RCUs used for the beamctl commands can be used to limit the range of antennas to make it easier to try down the misbehaving antenna.

`swlevel 2` method, courtesy of Pearse Murphy,

```
user1@lcu$ # Initialisation
user1@lcu$ swlevel 2
user1@lcu$
user1@lcu$ # RCU Warming, LBA, HBALo, HBAHi
```

```
user1@lcu$ rspctl --mode=3
user1@lcu$ rspctl --mode=5
user1@lcu$ rspctl --mode=7
user1@lcu$
user1@lcu$ # Shutdown
user1@lcu$ swlevel 0
```

Command References

rspctl (Expert mode)

```
-bash-4.2$ rspctl -X
rspctl usage:

--- RCU control ----------------------------------------------------------------------
rspctl --rcu                    [--select=<set>]  # show current rcu control setting
rspctl --rcu=0x00000000         [--select=<set>]  # set the rcu control registers
 mask       value
 0x0000007F INPUT_DELAY  Sample delay for the data from the RCU.
 0x00000080 INPUT_ENABLE Enable RCU input.

 0x00000100 LBL-EN       supply LBL antenna on (1) or off (0)
 0x00000200 LBH-EN       sypply LBH antenna on (1) or off (0)
 0x00000400 HB-EN        supply HB on (1) or off (0)
 0x00000800 BANDSEL      low band (1) or high band (0)
 0x00001000 HB-SEL-0     HBA filter selection
 0x00002000 HB-SEL-1     HBA filter selection
            Options : HBA-SEL-0 HBA-SEL-1 Function
                      0         0      210-270 MHz
                      0         1      170-230 MHz
                      1         0      110-190 MHz
                      1         1      all off
 0x00004000 VL-EN        low band supply on (1) or off (0)
 0x00008000 VH-EN        high band supply on (1) or off (0)

 0x00010000 VDIG-EN      ADC supply on (1) or off (0)
 0x00020000 LBL-LBH-SEL  LB input selection 0=LBL, 1=LBH
 0x00040000 LB-FILTER    LB filter selection
                    0    10-90 MHz
                    1    30-80 MHz
 0x00080000 ATT-CNT-4    on (1) is  1dB attenuation
 0x00100000 ATT-CNT-3    on (1) is  2dB attenuation
 0x00200000 ATT-CNT-2    on (1) is  4dB attenuation
 0x00300000 ATT-CNT-1    on (1) is  8dB attenuation
```

```
  0x00800000 ATT-CNT-0    on (1) is 16dB attenuation

  0x01000000 PRSG         pseudo random sequence generator on (1), off (0)
  0x02000000 RESET        on (1) hold board in reset
  0x04000000 SPEC_INV     Enable spectral inversion (1) if needed. see --specinv
  0x08000000 TBD          reserved
  0xF0000000 RCU VERSION  RCU version, read-only

rspctl [ --rcumode        |
         --rcuprsg        |
         --rcureset       |
         --rcuattenuation |
         --rcudelay       |
         --rcuenable      |
       ]+ [--select=<set>] # control RCU by combining one or more of these options with RCU selection

       --rcumode=[0..7] # set the RCU in a specific mode
         Possible values: 0 = OFF
                          1 = LBL 10MHz HPF 0x00017900
                          2 = LBL 30MHz HPF 0x00057900
                          3 = LBH 10MHz HPF 0x00037A00
                          4 = LBH 30MHz HPF 0x00077A00
                          5 = HB 110-190MHz 0x0007A400
                          6 = HB 170-230MHz 0x00079400
                          7 = HB 210-270MHz 0x00078400
       --rcuprsg[=0]              # turn psrg on (or off)
       --rcureset[=0]             # hold rcu in reset (or take out of reset)
       --rcuattenuation=[0..31]   # set the RCU attenuation (steps of 0.25dB)
       --rcudelay=[0..127]        # set the delay for rcu's (steps of 5ns or 6.25ns)
       --rcuenable[=0]            # enable (or disable) input from RCU's

rspctl --specinv[=0] [--select=<set>] # enable (or disable) spectral inversion
rspctl --mode=[0..7] [--select=<set>] # set rcumode in a specific mode
                                      # enable(or disable) input from RCU's
                                      # enable(or disable) spectral inversion
                                      # set the hbadelays to 253

--- Signalprocessing ---------------------------------------------------------
rspctl --weights                    [--select=<set>]  # get weights as complex values
  Example --weights --select=1,2,4:7 or --select=1:3,5:7
```

```
rspctl --weights=value.re[,value.im][--select=<set>][--beamlets=<set>] # set weights as complex value
OR      --weights="(value.re,value.im)(value.re,value.im)" [--select=<set>][--beamlets=<set>] # set multiple
weights
    as complex value for the same amount of selected beamlets
rspctl --aweights                   [--select=<set>]  # get weights as power and angle (in degrees)
rspctl --aweights=amplitude[,angle] [--select=<set>]  # set weights as amplitude and angle (in degrees)
rspctl --subbands                   [--select=<set>]  # get subband selection
rspctl --subbands=<set>             [--select=<set>]  # set subband selection
  Example --subbands sets: --subbands=0:39 or --select=0:19,40:59
rspctl --xcsubband                                    # get the subband selection for cross correlation
rspctl --xcsubband=<int>                              # set the subband to cross correlate
rspctl --wg                         [--select=<set>]  # get waveform generator settings
rspctl --wg=freq [--phase=..] [--amplitude=..] [--select=<set>]  # set waveform generator settings


--- Status info ------------------------------------------------------------------------------------
rspctl --version             [--select=<set>]  # get version information
rspctl --status              [--select=<set>]  # get status of RSP boards
rspctl --tdstatus            [--select=<set>]  # get status of TDS boards
rspctl --spustatus           [--select=<set>]  # get status of SPU board
rspctl --realdelays[=<list>] [--select=<set>]  # get the installed 16 delays of one or more HBA's
rspctl --regstate                              # show update status of all registers once every second
rspctl --latency                               # show latency of ring and all lanes


--- Statistics -------------------------------------------------------------------------------------
rspctl --statistics[=(subband|beamlet)]        # get subband (default) or beamlet statistics
            [--select=<set>]                   #
            [--duration=<seconds>]             #
            [--integration=<seconds>]          #
            [--directory=<directory>]          #
rspctl [--xcangle] --xcstatistics  [--select=first,second] # get crosscorrelation statistics (of pair of RSP boards)
            [--duration=<seconds>]             #
            [--integration=<seconds>]          #
            [--directory=<directory>]          #


--- Miscellaneous ----------------------------------------------------------------------------------
rspctl --clock[=<int>]                         # get or set the clock frequency of clocks in MHz
rspctl --rspclear            [--select=<set>]  # clear FPGA registers on RSPboard
rspctl --hbadelays[=<list>] [--select=<set>]   # set or get the 16 delays of one or more HBA's
rspctl --tbbmode[=transient | =subbands,<set>] # set or get TBB mode, 'transient' or 'subbands', if subbands
then specify subband set
```

```
rspctl --datastream[=0|1|2|3]                # set or get the status of data stream to cep
rspctl --swapxy[=0|1] [--select=<set>]       # set or get the status of xy swap, 0=normal, 1=swapped
rspctl --bitmode[=4|8|16]                    # set or get the number of bits per sample



--- Raw register control ---------------------------------------------------------------------
 ### WARNING: to following commands may crash the RSPboard when used wrong! ###
rspctl --readblock=RSPboard,hexAddress,offset,datalength     # read datalength bytes from given address

rspctl --writeblock=RSPboard,hexAddress,offset,hexData       # write data to given address
In all cases the maximum number of databytes is 1480
Address order: BLPID, RSP, PID, REGID
```

beamctl

- (Undocumented) -J/--remotehost: Use a remote server to host the beamserver?

```
-bash-4.2$ beamctl -h
Usage: beamctl <rcuspec> <dataspec> <digpointing> [<digpointing> ...] FOR LBA ANTENNAS
       beamctl <rcuspec> <anapointing> [<anapointing> ...] [<dataspec> <digpointing> [<digpointing> ...]]
FOR HBA ANTENNAS
       beamctl --calinfo
where:
  <rcuspec>      = --antennaset --rcus --band (or --antennaset --rcus --rcumode)
  <dataspec>     = --subbands --beamlets
  <digpointing>  = --digdir
  <anapointing>  = --anadir
with option arguments:
  --antennaset=name # name of the antenna (sub)field the RCU's are part of, may not conflict with band
                    # name = LBA_INNER | LBA_OUTER | LBA_SPARSE_EVEN | LBA_SPARSE_ODD |
                    # LBA_X | LBA_Y | HBA_ZERO |  HBA_ONE | HBA_DUAL | HBA_JOINED |
                    # HBA_ZERO_INNER | HBA_ONE_INNER | HBA_DUAL_INNER | HBA_JOINED_INNER
  --rcus=<set>      # subselection of RCU's
  --band=name       # name of band selection, may not conflict with antennaset
                    # name = 10_90 | 30_90 | 110_190 | 170_230 | 210_250
  --subbands=<set>  # set of subbands to use for this beam
  --beamlets=<list> # list of beamlets on which to allocate the subbands
                    # beamlet range = 0..247 when Serdes splitter is OFF
                    # beamlet range = 0..247 + 1000..1247 when Serdes splitter is ON
  --digdir=longitude,latitude,type[,duration]
```

```
                # lon,lat are floating point values specified in radians
                # type is SKYSCAN or almost any other coordinate system
                # SKYSCAN will scan the sky with a L x M grid in the (l,m) plane
  --anadir=longitude,latitude,type[,duration]
                # direction of the analogue HBA beam
  --rcumode=0..7    # Old-style RCU mode to use (DEPRECATED; only available for
                  compatibility with existing scripts. Please use antenna-
                  set + band selection. The rcumode selected here must not
                  conflict with the selected antennaset)
  --help            # print this usage

The order of the arguments is trivial.

This utility connects to the CalServer to activate the antennas in set --antennaSet
containing the selected RCU's. The CalServer sets those RCU's in the mode
specified by --rcumode. Another connection is made to the BeamServer to create a
beam on the selected antennafield pointing in the direction specified with --digdir.
```

Generating XST Data (Empty)

Mode 357 (Breaking the Station in the Name of Science) (Empty)