The Parkes Conversion Chain - User Guide

M.Kesteven, G.Graves, M.Bowen, S.Jackson

May 8, 2000

1 Overview

The conversion rack at Parkes now provides remote (computer) control over all aspects of the conversion chain.

1.1 Hardware

Figure 1 is the block diagram of the essential items of hardware.

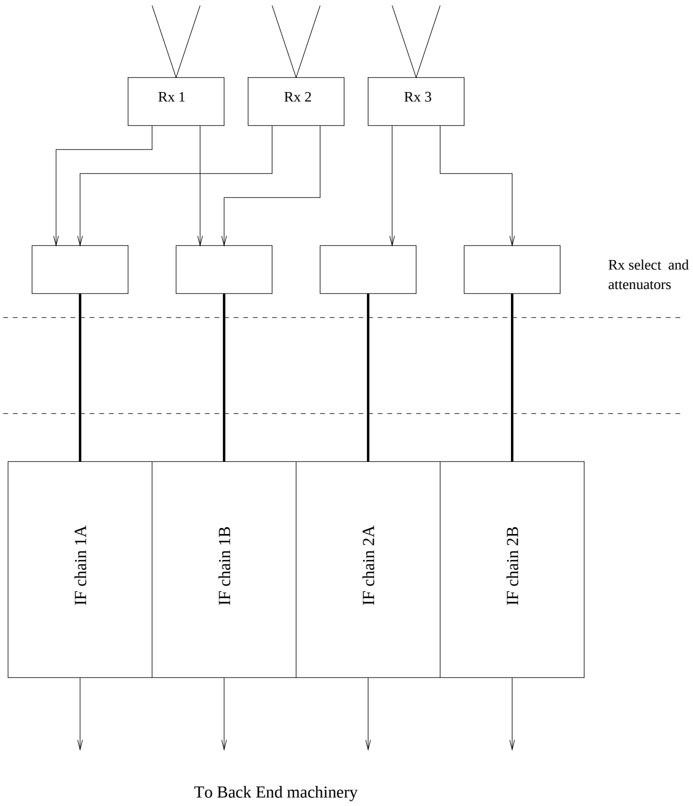

Figure 1: A view of the Parkes receiver and conversion machinery system

The designers' intention is that the observer should be able to connect any receiver in the focus cabin to the conversion rack (in the control room), perform all the frequency translation and conditioning necessary to match the IF to the backend - all under computer control.

- Four high quality cables are available to trasport the IF from the focus cabin to the control room. Each cable is gain stabilised to give uniform frequency response over each of the four IF bands (C-, S-, L- and UHF bands) currently implemented.
- In the focus cabin there are four patch panels, one for each cable. The connections within the panel are computer selectable; an attenuator is also provided.
- The conversion rack has four IF chains; these will always operate as pairs: (1A/1B) and (2A/2B), where A and B suggest two polarisation states. Each chain is connected to a focus cabin cable. Cross-over switches have been provided so that the "A" cables can connect to either "A" IF chain; and similarly for the "B" cables and IFs.

  The IFs within each pair share synthesisers.
- A variety of output ports are available on each IF chain.
- Frequency switching observations are available.

A block diagram of the layout within a conversion chain is shown in figure 2.

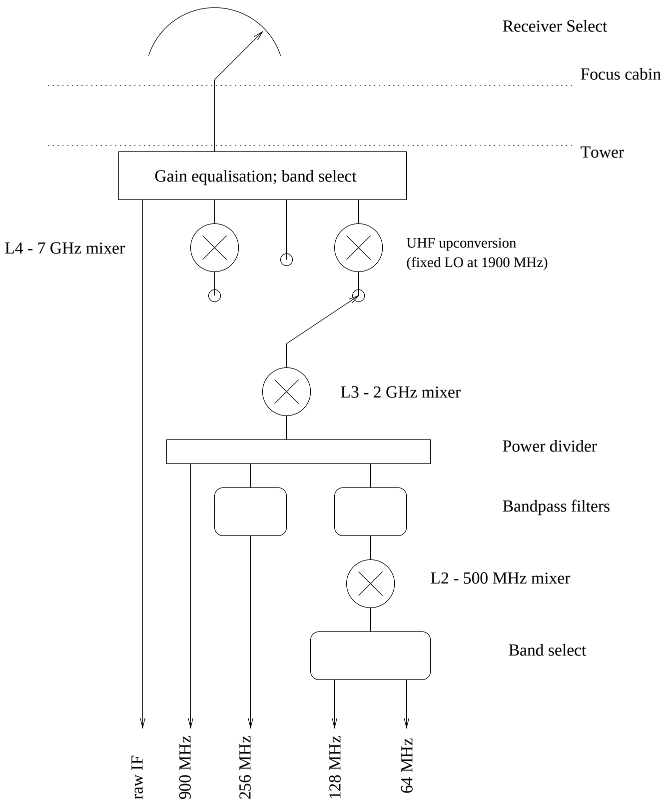

Figure 2: A block diagram of one conversion chain; there are four of these units within the new conversion rack

The technical specifications of the IF chains are given in table 1

Input bands (MHz)	Input levels (dBm)	
4500. - 6100.	xx	
2200. - 3600.	yy	
1200. - 1800.	zz	
300. - 750.	zz	
Output bandwidths (MHz)	**Band Centre (MHz)**	**Number of ports**
900.	500.	2 per channel
256.	384.	3
256.	640.	3
128.	192.	3
64.	96.	4
unfiltered RF	RF band centre	1

Table 1: The technical specification of the conversion system

Notes :
- One 64 MHz port in each IF chain is normally connected to the DAS.
- One 64, 128 and 256 port in each IF chain is connected to the AT correlator.
- One set of output ports is available on the rear panel.
- One set of output ports is available on the front panel.
- The observer can offset a band centre from the nominal settings listed above, with appropriate entries in the receiver setup file. The observatory staff must be consulted prior to this exercise.

A detailed account of the system is given in Graves et al, (2000).

1.2 Software

Figure 3 shows the software block diagram. The main task (LOBOSS) is a server process; it controls all the conversion hardware; it responds to requests (control and monitor) from the observing and monitoring tasks.

It is possible to control any item in the conversion system, but the observer will generally prefer to use the high level utilities: specify the receiver to use, the sky frequency and the bandwidth to be presented to the correlator.

Several support utilities have been provided to allow the observatory staff to check the operation of the system, to make adjustments (eg, to set IF levels), or to set up the conversion rack for non-standard or diagnostic operations.

The remainder of this document deals with the software operations. Section 2 details the control language; section 3 describes the associated utilities; section 4 addresses some design (operation) issues; section 6 describes the supporting infrastructure.

2 Control Language

Ascii strings are used throughout for the commands and the responses. Every message to LOBOSS will elicit a response; In general this will consist in a number of informative messages, followed by a final message containing the key “<done>”, and a status flag (0 .. OK). The informative messages will have a status prefix, “<OK>” or a“<ERR>”. The error flag is followed by a diagnostic message.

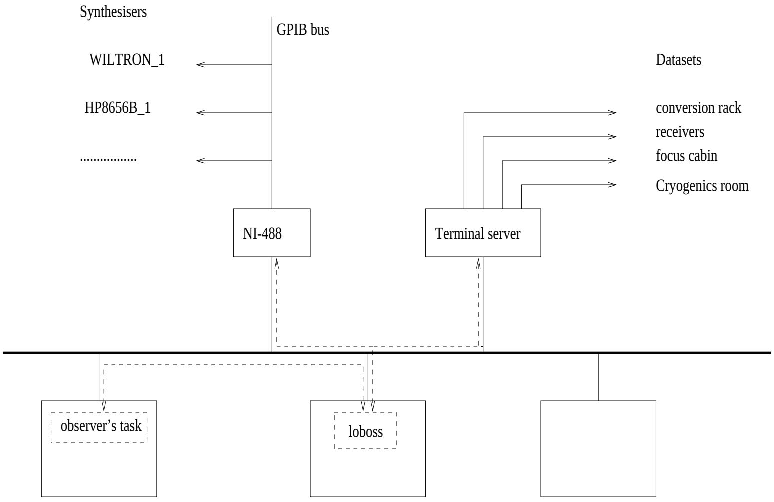

Figure 3: This figure shows the general software connections

2.1 Observer-level Operation

- Define the receiver setup to be used.

  lo_drv lo_name Sname

  Sname is the name of the receiver setup file. These files are maintained by the observatory, and describe:

  - how the receiver is connected to the focus cabin patch panels;

  - what conversions (if any) are required by the focus cabin synthesiser;

  - the conversion chain algorithm(s) to be used;

  - how the back-end is connected to the conversion rack.

  An annotated example is given in appendix A.

- Define the observation.

  lo_drv lochain fr1 freq bw [fr2 freq bw]

  freq is the frequency at band centre at the receiver, in MHz; bw is the bandwidth. fr1 refers to the (1A/1B) IF pair, fr2 to the (2A/2B) pair. Either or both can be specified, in any order.

  If the operation is successful, there will be an additional message, prefixed with "<REPLY>" containing the inversion state of the spectrum that will be produced by the conversion chain.

  The informative messages from LOBOSS will describe the new settings to hardware. An effort has been made to reduce the unnecessary traffic and the repeated setting of hardware to the same value; to do this, LOBOSS maintains an image of the conversion rack. This image built from scratch after each new receiver selection (after each **lo_name** call). The reply following an **lochain** call may therefore be quite terse if only a few items have needed to be reset.

  Users **MUST** specify both fr1 and fr2 in the same call if two-chain operation is intended. The reason is that both chains may share a synthesiser (as in the focus cabin setup); specifying both fr1 and fr2 allows loboss to optimise the synthesiser setting. Indeed, if the two chains are specified separately, then it is likely that the second call will render irrelevant (and wrong) the setting on the first chain.

- Set up the frequency switching

  lo_drv fqsw Δf

  where Δf is the sky frequency offset required. This is discussed below in section 4.2.

- Adjust the attenuator levels.

  lo_drv setlevel

  lo_drv autolevel_on This will force a level check whenever a new frequency is selected.

  lo_drv autolevel_off This disables the ato levelling.

- Initialise the four Digital Synchronous Demodulators.

  lo_drv init usecs

  where usecs is the integration time, in microsecs.

- Read the DSDs

  lo_drv get_dsd

  The response, in the message prefixed with "<REPLY>", will contain eight integers, the ON and OFF counts from the four DSDs.

Appendix B contains an annotated example of an exchange using lo_drv.

2.2 Low-level Operations

Terminology All the control and monitor points are identified in the following manner:

channel:module.ID

where,

channel [1..4] maps to [1A,1B,2A,2B]
module is the hardware unit,
ID is the item with the module.

Some abbreviations are allowed:

- channel 0 is implied if no channel is given. General purpose devices and monitor points may not be channel-specific.

- It is permissable to refer to the GPIB synthesisers directly by name - HP_2, etc.

- If the "module" is the name of a dataset (eg, C31_ds), then "pname" must be the index for the dataset operation.

set point value will set the hardware to the value specified, and then do a readback check.

show point will return with the current setting. An error condition will be signalled if the current value differs from the last requested setting.

some examples:

set 1:L4.freq 7500.	set the L4 synthesiser, IF chains 1A and 1B to 7500 MHz
show 2:C14.atten	get the c14 attenuator setting
set hp_2.freq 7500.	set the hp_2 synthesiser to 7500 MHz
set hp_3.rfon	set the hp_3 synthesiser RF power to ON
set c31_ds.64 111	set the c31 dataset, index 64, to the value 111

2.3 loboss operations

allocate, deallocate These commands will take/release exclusive control of the hardware. This feature is not implemented at present, although the main software hooks are in place.

enable, disable These commands mediate the control of the dataset and gpib bus.

lo_debug This will toggle a "verbose" mode of operation. (not recommended).

quit the preferred (graceful) way a task should close its connection to loboss.

reinit this forces loboss to clear its image of the setting of the conversion rack, and ensures that at the next setup, all the hardware will be set.

3 Utilities

NOTE : All these utilities require the user to have activated the ATCS option in their .login.packages file.

3.1 PC interface

3.2 logui

This is a Glish-based process that provides a view of all the control points, and allows the user to control each item in the rack.

3.3 lomon

This is a companion to logui - it provides a view of the monitor points.

3.4 loclient

This task will establish a connection to loboss, allowing the user to emulate, via the keyboard, an observing task. Each keyboard string will be sent, unmodified, to loboss. All the loboss messages will be sent to the screen.

Appendix B shows an example of its use.

3.5 check_lo_setup

This task will examine a given receiver setup file, and provide a synopsis of its understanding. It then operates much as lochain (as at narrabri/mopra), reporting the conversion chain details for a given observation - sky frequency and bandwidth.

3.6 gpib-enet

The GPIB bus which drives the synthesisers is controlled by an NI-488 GPIB-ENET device. This is an ethernet unit, PKGPIB00. National Instruments have provided the driver/library; this can be found in

/nfs/atapplic/GPIBENET/lib. The user guide to the library can be found in /nfs/atapplic/GPIBENET/doc

A critical parameter definition file (/nfs/online/local/gpib/gpibrc) is accessed via the environment variable GPIBCONF.

The observatory maintains a configuration file, detailing all the gpib devices that can be controlled. The file can be read/updated with the **ibconf** utility. See the gpibenet user guide for details.

ibic is another utility, similar to loclient - it allows you to check gpib commands.

3.7 dummy_gpib and dummy_dset

It is useful to be able to check out new software without actually controlling the hardware. These two utilities react benignly, but meaningfully, to gpib and dataset commands. The loboss.ini file needs to be reset to point to these processes, rather than to the terminal server (for the datasets) and the NI488 (for the gpib devices).

4 Design Issues

4.1 Communications

Loboss provides a server socket to the network. Up to ten simultaneous socket connections are allowed. All have equal priority, although as discussed above, exclusive control over hardware may be granted.

Each ascii message to and from loboss has a two-byte header; this contains a header ID and a byte count. This means that multiple messages can arrive in one socket read and be successfully extracted.

Tasks wishing to connect to loboss need to know the host name and port number. There is no restriction on where loboss or its clients operate. Loboss deposits its particulars in the file ;

/nfs/online/local/registry/loboss.reg

4.2 Frequency Switching

The synthesisers do not offer a simple mechanism to change frequencies in a strobed manner, synchronised with, for example, the correlator operations.

We have therefore adopted an alternative (somewhat heavy-handed) approach:

Two synthesisers are used, one tuned for the **signal** setting, the other tuned to the **reference** setting. An RF switch allows us to direct either synthesiser to the mixers. The RF switch requires a TTL logic signal, generally derived from the correlator.

There two RF switches, one for the MULTI-beam system, and a second unit in the conversion rack. The conversion rack unit is a cross-over switch, with two states:

L3 (IF chain #1) -> IF chain #1
L3 (IF chain #2) -> IF chain #2 or

L3 (IF chain #2) -> IF chain #1
L3 (IF chain #1) -> IF chain #2

This would allow frequency-switching with no loss of signal-to-noise:

IF chain #1 operates on the cycle SIG; REF; SIG; REF; ...
IF chain #2 operates on the cycle REF; SIG; REF; SIG; ...

The cross-over switches at the top of the rack ensure that the same RF is present in all IF chains.

The call **lo_drv fqsw Δf** will attempt to sort this out, in conjunction with the data in the receiver's setup file.

4.3 Frequency Switching Modes of Operation

- MULTI-beam operation. There are two synthesisers, and no choice: all 13 beams observe at the sky frequency dictated by one or the other synthesiser.
- Conversion rack operation, one IF pair.

  This is an efficient mode available at all frequencies.

  Set both chains (1A/1B) and (2A/2B) for the same sky frequency. Then offset L3 on the second chain by the frequency offset.

  Chain 1 will follow the sequence SIG; REF; SIG; .. and the second follows REF; SIG; REF; ..
- Two IF pair. This is available for C-band and above. In this case one can have the two chains set to quite different sky frequencies. This mode is possible provided the chains are set up so both have the same L3 synthesiser setting. This requires the two IFs to fall within the same input band .. (4000 to 5000 MHz) or (5000 to 6000 MHz). After the initial setup, offset one L3 by the frequency offset.

  For example: K-BAND observations at 22235. and 22500., both at 64 MHz bandwidth. The initial settings are:

  W_3 (focus cabin) = 17567.5 MHz

  IF(1) = 4667.5 MHz .. L4 = 7167.5; L3 = 2912. L2 = 572.
  IF(2) = 4932.5 MHz .. L4 = 7432.5; L3 = 2912. L2 = 572.

  Clearly, these are not modes for the faint-hearted - consult the observatory staff.

4.4 Level Checking and Attenuator Setting

The setting philosophy is:

- adjust the attenuator in the focus cabin until the IF presented to the patch panel is close to the level required for the band which will be sent to teh conversion system. The receiver's setup file will provide a good first estimate of the setting.

- progress down the chain, setting each attenuator to achieve the nominal levels at each mixer.

The focus cabin attenuator should be left alone after the initial setting, if possible - its phase/attenuation characteristics is not as good as those of the attenuators in the conversion rack.

4.5 Read Back Checks

Every **set** command will do a read back check. This does impose a difficulty, as some units, particularly the switches, take time to complete the operation. There will always be a minimum wait of 0.1 secs between setting and readback; a call for multiple hardware settings will incur just one wait.

To alleviate the latency we employ two strategies:

- The readback is postponed, if possible. This is relevant to the setting up of an entire conversion chain. One single command is received, but a dozen switches are reset. The check is made at the end of the sequence, after all the **set** commands have been issued.

- We avoid unnecessary operations. Loboss maintains an image of the conversion system, and can therefore tell if an operation is required. This technique is slightly risky - if hardware can be changed without loboss being aware of the change. To minimise this risk, the image is rebuilt from scratch whenever a new receiver is selected, if if the **reinit** command is received.

5 Programming Issues

5.1 Dataset Control and Monitor Points

The dataset protocol and programming issues are described in R. Ferris' guide to the dataset (xx). The software is a slightly enhavced version of the original narrabri code, modified for the terminal server which provides an ethernet interface to the dataset bus.

Hardware complexities have, however, made the programming somewhat more difficult than elsewhere – the multiplicity of control/monitor points, coupled with the paucity in the regicter count means that some registers control/monitor several points. To set a control point without affecting its register-companion means a two-step operation:

- readback the control point to obtain the current setting;

- mask the unwanted portions of the register word;

- write the new setting with the masked region of the register word;

- write the word to the dataset.

5.2 GPIB programming

At this stage we use a limited set of the gpib driver set -

- **ibfind (device)** initialise the device.

- **ibwrt (deviceID, msg, nbytes)** write to the device.

- **ibrd (deviceID, msg, nbytes)** read from the device.

Users should note that the configuration file (gpibrc) contains all the details needed by the driver, simplifying life somewhat.

6 Infrastructure

6.1 Parameter Files

The files containing the control parameters are kept, in readonly format, in the area /nfs/online/local/loboss

The files are :

Conversion Chain
Algorithm Files

input.def	This defines the input section (everything down to the L3 mixer)
M900.def	This sets up the 900 MHz output section
M256.def	The 256 MHz output section
M128.def	The 128 MHz output section
M64.def	The 64 MHz section

Hardware definition files

gpib_devices.def	This file describes all the devices currently known to the GPIB controllers. It also contains the synthesiser details : min/max frequency; command protocol.
loboss.ini	This file has makes the connections between generic names and the specific hardware implementation
points_db.def	This file has the list of all the control and monitor points associated with the conversion chain. It details their link to the hardware. An annotated copy is given in appendix C
synthesiser.def	This provides the link between generic synthesiser names and the hardware names.

The loboss log files will be found in /nfs/online/logs

7 Programmer Support

The code resides in the ATCS structure -

atcs/code/loboss for loboss and the utilities
atcs/code/libraries/lo for the sup[porting libraries.

The gpib directory is :

/nfs/atapplic/GPIBENET/

with lib, doc and other relevant subdirectories.

8 References

G.Graves, M. Bowen & S. Jackson, WARS_2000.

A Setup file

This is the setup file for the K-BAND receiver

```
receiver

  nfreq = 1 ! not a dual-frequency rx.

  freq_unit ! start defining the Rx characteristics

    freq.lo = 22000. ! lower band limit, in MHz
    freq.hi = 24000. ! upper
    npol = 2 ! number of polarisatoin channels

    switchA.sw = 1 ! poln A is connected to switch #1
    switchA.pl = 4 ! poln A is connected to pole #4
    switchB.sw = 2 ! plon B is connected to sqitch 2, pole #4
    switchB.pl = 4


    synth    = W_3 ! the name of the down-conversion synthesiser
    target   = 4800.0           ! the target for the first conversion
    sideband = 1.0 ! using the upper sideband


  #end ! end, Rx characteristics,


#end ! end, Rx section



module ! start defining the next conversion modules

  mod_name = LO_RACK    ! use the conversion rack

#end ! end module definition



backend ! start defining the backend connections

  nfreq = 2 ! number of independent IF pairs

  freq_unit ! start defining the first IF pair

    npol = 2 ! number of polarisation IFs
    port_typeA = 128 ! poln A port type
    port_typeB = 128    ! poln B port type
    portA      = 1 ! connect IF chain 1A to the A poln of the backend
    portB      = 2      !                 1B        B
```

```
    nbandw = 3          ! number of bandwidths defined

    bw_unit ! start defining bandwidth #1
      bandw = 64.       ! bandwidth, in MHz
      target = 160.     ! band centre, in MHz
      invert = 1        ! inversion at the correlator
    #end ! end, bandwidth #2

    bw_unit
      bandw = 8.
      target = 140.
      invert = -1
    #end

    bw_unit
      bandw = 4.
      target = 138.
      invert = 1
    #end

  #end ! end, first IF pair

  freq_unit

    npol = 2
    port_typeA = 128
    port_typeB = 128
    portA      = 3
    portB      = 4

    nbandw = 3

    bw_unit
      bandw = 64.
      target = 160.
      invert = 1
    #end

    bw_unit
      bandw = 8.
      target = 140.
      invert = -1
    #end

    bw_unit
      bandw = 4.
      target = 138.
      invert = 1
    #end

  #end

#end
```

B Lo_drv exchange

This is an example of the use of **loclient** emulating an observing task. The string following the > is sent to loboss. the strings following the "(from Boss)" are the loboss replies.

```
 loclient > lo_drv lo_name K-BAND
(from Boss) <OK> set C31_DS.64
(from Boss) <done> 0

 loclient > lo_drv lochain fr1 22235 6
(from Boss) <OK> set W_3 to 17435.000000
(from Boss)  <OK> LoBoss will set freq/bw 4800.000000/128.000000
(from Boss) <OK> 1:C51.SW = 128
(from Boss) <OK> 1:L4.FREQ =      7300.0000
(from Boss) <OK> 1:L3.FREQ =      2912.0000
(from Boss) <OK> 1:L2.FREQ =       572.0000
(from Boss) <OK> 1:C40FIL.SW = 128L
(from Boss) <OK> 1:C40BW.SW = 128
(from Boss) <OK> 1:C14.SW = C
(from Boss) <OK> 1:C30CX.SW = CN
(from Boss) <OK> 1:C30LS.SW = SC
(from Boss) <OK> 1:C52.SW = 128
(from Boss) <OK> 2:C51.SW = 128
(from Boss) <OK> 2:L4.FREQ =      7300.0000
(from Boss) <OK> 2:L3.FREQ =      2912.0000
(from Boss) <OK> 2:L2.FREQ =       572.0000
(from Boss) <OK> 2:C40FIL.SW = 128L
(from Boss) <OK> 2:C40BW.SW = 128
(from Boss) <OK> 2:C14.SW = C
(from Boss) <OK> 2:C30CX.SW = CN
(from Boss) <OK> 2:C30LS.SW = SC
(from Boss) <OK> 2:C52.SW = 128
(from Boss) <REPLY> invert = -1
(from Boss) <done> 0
```

C Annotated Control and Monitor Point List

The monitor point names are in the left-hand column; the key in "()" identifies the type : Control or Monitor. The associated number indicates whether the point is unique (*1), or attached to a channel (*4).

```
L4 (freq/lev/onof)  Synthesisers
L3 (freq/lev/onof)
L2 (freq/lev/onof)


C11_ds          focus cabin dataset
C12.sw          (C*4) rx select switch
C12.att         (C*4) attenuator
C12.rflev       (M*4) rflevel after the attenuator
C12.reg         (M*4) focus cabin regulated voltage

C13_ds          dataset
C13.alloc       (C*1) allocate
C13.reg         (M*1) Interface regulated voltage
C13.fan         (M*1) Fan status
C13.tempst      (M*1) Rack temperature status
C13.temp        (M*1) Rack temperature
C13.fqsw        (C*1) Frequency Switch (Sig/Ref/Switching)
C13.fqwf        (M*1) FQSW waveform status
L11.lolock      (M*1) Upconverter LO lock status
L11.lolev       (M*1) Upconverter LO level
L11.reg         (M*1) Upconverter LO Vreg. status

L11.refst       (M*1) 100 MHz ref. level status
L11.rflev       (M*1) 100 MHz rf. level

L12.reg         (M*1) 10 MHz Vreg. status

C14.sw          (C*4) Band select switch
C14.reg         (M*4) Band select Vreg. status
C60.reg         (M*4) IF buffer Vreg status
C61.sw          (C*4) Band Width detector select
C61.reg         (M*4) MUX Vreg. status

C31_ds          dataset
C30.alloc       (C*1) allocate
C30.att         (C*4) coarse attenuator
C30.reg         (M*4) conversion Vreg status
C30CX.sw        (C*4) CX switch
C30CX.lolev     (M*4) L4 - LO power level
C30CX.iflev     (M*4) L4 - IF power level
C30LS.sw        (C*4) LS switch
C30LS.lolev     (M*4) L3 - LO power level
C30LS.iflev     (M*4) L3 - IF power level
C31.reg         (M*1) C31 - Interface Vreg. status
```

```
C91.rflev       (M*4) UHF Splitter RF level
C91.lolev       (M*4) upconverter LO power level
C92.rflev       (M*4) L Splitter RF level
C93.rflev       (M*4) S Splitter RF level
C94.rflev       (M*4) C Splitter RF level

C41_ds          dataset
C40.alloc       (C*1) allocate
C40.att         (C*4) fine attenuator
C40.reg         (M*4) 64/128 MHz IF Vreg. statsu
C40.lolev       (M*4) L2 - LO power level
C40.iflev       (M*4) L2 - IF power level
C40.if64m       (M*4) 64 MHz output level
C40.if128m      (M*4) 128 MHz output level
C40FIL.sw       (C*4) 64/128 MHz image filter switch
C40BW.sw        (C*4) 64/128 MHz band select switch
C41.reg         (M*1) Interface Vreg. status
C50.if256m      (M*4) 256 MHz output level
C50.reg         (M*4) 256 MHz IF Vreg. status
C50.att         (C*4) fine attenuator
C50TST.sw       (C*4) 256 MHz input switch
C50FIL.sw       (C*4) 256 MHz filter select
C51.sw          (C*4) DSD MUX band select
C51.reg         (M*1) Digital synch Vreg. status
C52.sw          (C*4) Band select, analog synch
C52.wfsel       (C*1) Analog synch, Waveform select
C52.wfact       (M*1) Analog synch - Waveform status
C52.reg         (M*1) Analog Synch,  Vreg. status
C52.atp         (M*4) Average Total Power
C52.ptp         (M*4) Peak Total Power
C52.gtp         (M*4) Gated Total Power
C52.sdo         (M*4) Synch. Demod Output
C54.if900m      (M*4) 900 MHz output level
C54.reg         (M*4) 900 MHz IF Vreg. status

C51_ds          dataset
C51.dsdon       (M*4) DSD (ON)
C51.dsdoff      (M*4) DSD (OFF)

MB_ds           multi-beam dataset
MULTI.t20a      (M*1) T20(A) station temp
MULTI.t20b      (M*1) T20(B) station temp

MULTI.t70a      (M*1) T70(A) station temp
MULTI.t70b      (M*1) T70(B) station temp

MULTI.vacc      (M*1) vaccuum

MULTI.p15ra     (M*1) +15(A) regulated voltage
MULTI.p15rb     (M*1) +15(B) regulated voltage

MULTI.m15ra     (M*1) -15(A) regulated voltage
MULTI.m15rb     (M*1) -15(B) regulated voltage
```

```
MULTI.p5ra      (M*1) +5(A) regulated voltage
MULTI.p5rb      (M*1) +5(B) regulated voltage
```

---

file : /u/mkesteve/atcs/code/loboss/text/loboss_guide.tex