NESIE module RF board

Telnet Test Interface

Version 1.17 (Draft) 12 October 2018

Author Paul Bray

Contents

Introduction

The Telnet Test Interface provides Telnet Access to the BIST for both testing NESIE Module RF boards and calibration. The Interface is provided by a python telnet server script that can be run on a PC, or on a Linux OS running on the target. It will communicate with the BIST functionality provided by a NESIE DA Module via the SM SAP interface, using TCP Sockets. Its functionality has grown with time and now allows testing of rather more than the RF Boards. There is support for both Flight/Nesie2 and Covert hardware, some commands are common, some are specific.

This document covers Flight, NESIE2, Covert and Tactical Hardware.

General

Currently the Telnet Server opens port 51234 for the client to connect to:

```
    HOST = ''
    PORT = 51234.
```

[TBD - Make this configurable from the command line when running the script]

The Telnet Server needs to be configured to communicate with the desired NESIE DA Module (Currently Board 0 on the local processor, hardcoded into the python code

```
    DA_MODULE_IPADDR   = '127.0.0.1'
    DA_MODULE_PORT     = 20000
```

[TBD - Make this configurable from the command line when running the script]

The Telnet Server provides a Simulation Mode, currently enabled or disabled in the python script

```
    #BIST_SIMULATION = False
    BIST_SIMULATION = True
```

[TBD - make configurable from the command line]
When Simulation Mode is enabled, the Server does not attempt to communicate with the NESIE DA Module over the SM-SAP interface, It generates sensible responses to the supported commands and queries. This is useful when developing the telnet clients when other downstream components are not available.

Common

All commands / queries sent to the Telnet Server cause a response to be sent back to the client.

Each command/response should be terminated with a '\n' character a.k.a. <line-feed>; ^J etc.

All response lines are terminated with a '\n' character a.k.a. <line-feed>; ^J etc.

All error responses are of the form "ERR:'ERROR MESSAGE'\n" for instance:

```
    "ERR:\'Unrecognised Tx Enable/Disable Command\'\n"
```

Queries will cause a non-error response, for instance, the report BB Rx RSSI query:

“RX:RSSI? BB”

Will generate a response such as:

“-80\n”

A positive ACK to a command is represented by an empty line:

“\n”

Each command should have an associated query, formed by appending a “?” to the command. This leads to some redundant queries e.g. “TX:ENAB?\n” And “TX:DISA?\n”.

Supported Commands / Queries

Text shown in lower-case in Commands and Queries is optional.

Tx Command Summary

TX:ENABle TX:ENABle? TX:DISAble TX:DISable?	Enable / Disable the Transmitter Query the Transmitter Enable State
TX:MUTE TX:UNMUte TX:MUTE? TX:UNMUte?	Mute / Unmute the Transmitter Query the Transmitter Mute State
TX:LOOP ENABle TX:LOOP DISAble TX:LOOP?	Enable / Disable Tx Loopback Query the Tx Loopback State
TX:TS:FREQ <ts_freq_in_hz_parameter> TX:TS:FREQ? TX:TS:ABS <ts_freq_in_hz_parameter> TX:TS:ABS? TX:TS:LEVEL <level in dBFS> TX:TS:LEVEL? TX:TS:ENABle TX:TS:DISAble TX:TS:ENABle? TX:TS:DISAble?	Transmitter Test Source Commands Set / Query the Test Source Frequency Set / Query the Test Source Absolute Frequency Set / Query the output level of the Test Source Enable / Disable the Test Source Query the Test Source State
TX:BAND <fwd_rev_parameter> <band_parameter> TX:BAND?	Set / Query the Tx Band
TX:ATTN <attn_parameter> TX:ATTN?	Set / Query the Tx Attenuator Setting
TX:SIGS?	Query the Tx Signal Strength
TX:PORT <port_parameter> TX:PORT?	Set / Query the Tx Output Port
TX:DDS:FREQ <DDS Freq in Hz> TX:DDS:FREQ? TX:DDS:ENABle TX:DDS:ENABle? TX:DDS:DISAble TX:DDS:DISAble?	Enable/Disable and set freq of TS DDS, with queries. Seems to replicated TX:TS commands

TX:DAC:Write (<reg_addr> <reg value>) {1,}
TX:DAC:Read <reg addr}{1,}
	Read and Write Flight/Nesie2 AD9122 DAC registers

TX:FREQ <TX LO Freq in Hz > TX:FREQ? \n”	Covert/Tactical Only. Set/Query AD9361 Tx LO Frequency
TX:PA ENABle TX:PA ENABle? TX:PA DISAble TX:PA DISAble?	Covert Only Enable/Disable/Query State of PA
TX:BW <Tx Bandwidth in Hz > TX:BW?	Covert/Tactical Only Set/Query AD9361 Tx Bandwidth.
TX:DUP <Duplexer> TX:DUP?	Tactical Only Set/Query the RF Board Tx Duplexer
TX:FORREV [SET|CLEAR] TX:FORREV?	Tactical Only Set/Query the Forward/Reverse Control line for Tx Band.
TX:PAPATH <PA path> TX:PAPATH?	Tactical Only Set/Query the RF Path on the PA Board (Directs RF signal to/from a particular PA)
TX:PAEN <PA path> TX:PAEN?	Tactical Only Enable/Query the specified PA on the PA Board.
TX:PAPWR?	Tactical Only Query the Tx Power
TX:PATEMP?	Tactical Only Query the PA Board Temperature.

Rx Command Summary

RX:ENABle RX:ENABle? RX:DISAble RX:DISable?	Enable / Disable the Receiver Query the Receiver Enable State
RX:BAND < fwd_rev_parameter> <band_parameter> RX:BAND?	Set / Query the Rx Band
RX:LNA <lna_parameter> RX:LNA?	Set / Query the LNA in use
RX:RSSI? RF RX:RSSI? IF RX:RSSI? BB RX:RSSI? OF RX:RSSI? INPUT <freq in Hz>	Query the RSSI of the specified type
RX:GAIN <gain_parameter> RX:GAIN?	Set / Query the Rx RF Gain
RX:IFATtn <if_attn_parameter> RX:IFATtn?	Set / Query the Rx IF Attenuation Setting
RX:CAPT? <capture_length_parameter> | PRS | AD9361_TEST	Capture a block of BaseBand (BB) Rx Data of the specified length, or use Pseudo Random Sequence from the ADC to test data path, or control the AD9361 Rx Test Patterns.
RX:TEST ENABLE <test_code> RX:TEST DISABLE 0	Set ADC into test mode generating test patterns on Output lines. Data can be inspected using RX:CAPT? command
RX:LRSSIEN RX:LRSSIEN? RX:LRSSIDIS RX:LRSSIDIS?	Dis/En-able/Query the Rx LNA RSSI measurement
RX:FPGA	Read/Write a register in the memory mapped interface to the ADC block in the DA board.
RX:ADCPWR:DOWN RX:ADCPWR:UP	Power Up or Down the AD9467 ADC on Flight/Nesie2
RX:ADCREAD <hex register address>	Read the specified register contents from

	the AD9467 ADC on Flight/Nesie2
RX:ADCWRITE <hex register address> <hex 8-bit register value>	Write the specified value to the specified AD9467 ADC register on Flight/Nesie2
RX:FREQ <Freq Value in Hz> RX:FREQ?	Set/Query the AD9361 Rx LO Frequency on Covert/Tactical.
RX:BW <Bandwidth in Hz> RX:BW?	Set/Query the AD9361 Rx bandwidth on Covert/Tactical.
RX:DUP <Duplexer> RX:DUP?	Tactical Only Set/Query the RF Board Rx Duplexer
RX:FORREV [SET|CLEAR] RX:FORREV?	Tactical Only Set/Query the Forward/Reverse Control line for Rx Band.
RX:INPUT [ANT|DAISY] RX: INPUT?	Tactical Only Set/Query the RF Path from the RF In Socket to the Duplexers/RF Out Socket. Input can be from an Antenna or the RF Out of another Tactical unit (Daisy-Chained.

Cal Command Summary

WRITE <binary file name>	Write the binary file to the Calibration data EEPROM
READ? <page number>	Read the specified calibration data page from the RF board EEPROM. Any valid page can be read (numbers 0 .. 63) including the IDENT pages (numbers 0 .. 2), the CAL pages (numbers 3 .. 62) or the CRC page (number 63). Note that the page ranges are implementation defined and may be changed.
READ? F <file name>	Read the calibration data pages from the RF board EEPROM and write them to the specified file (in binary format).
ZERO	Reset the CAL data pages of the RF board EEPROM to their default contents (all zeroes). The correct CRC is written for each affected page.
STATUS?	Read the status of the calibration data. If present and valid, the data held in the RF board EEPROM will be used; if this data is not present the hardcoded default values held in the unit will be used; if these values are not present, the calibration data is deemed to be invalid.

STATUS Command Summary

There are 3 STATUS subdivisions namely RF, DSP and DA corresponding to the RF Board, the DSP BIST SAP Connection and the DA board (Main board) respectively.

STATUS:RF Command Summary

TEMP?	Query the RF Board temperature measurement
POST	Rerun and return the results of, the RF BoardPower On Self Tests
POST?	Return the last set of results for the RF Board Power On Self Tests

STATUS:DSP Command Summary

STATUS:DSP?	Query for BIST SAP Information

STATUS:DA Command Summary

TEMP?	Query the DA Board temperature measurement

OCXO Command Summary

OXCO <ocxo_parameter> OCXO?	Set / Query the OCXO Digipot/DAC Setting

LADON Command Summary

LADON:PRES?	Query ‘Ladon Present’
LADON:INIT?	Query ‘Ladon Initialised’

LADON:PATTN?	Query Ladon PA Attenuation
LADON:PATTN	Set Ladon PA Attenuation
LADON:FWDP?	Query Ladon Forward Power
LADON:TXFREQ	Set Ladon Tx Frequency
LADON:PENA	Enable/Disable Ladon PA
LADON:STATUS?	Query Ladon “Status and Error” registers
LADON:TXFBAND	Set Ladon Tx Filter Band
LADON:TXFBAND?	Query Ladon Tx Filter Band
LADON:RXFBAND	Set Ladon Rx Filter Band
LADON:RXFBAND?	Query Ladon Rx Filter Band
LADON:RXCMNAMP	Enable/Disable Ladon Rx Common Amplifier
LADON:RXCHANAMP	Enable/Disable Ladon Rx Channel Amplifier
LADON:RXGAIN?	Query Ladon Rx Gain
LADON:MOD	Select Ladon Modulation

ID Command Summary

ID:RFSN?	Query the RF board serial number
ID:DASN?	Query the DA board serial number
ID:DASN	Set the DA board serial number
ID:TRXSN?	Query the Transceiver serial number
ID:TRXSN	Set the Transceiver serial number

Eng Commands

The 'ENG' instrument hosts a number of engineering commands. These are only intended to be used during development and should not be used at other times.

OCXO	Set the evolved OCXO value for one of the Air Interface applications. The value can be reset so that the system falls back to the calibrated value.
OCXO?	Query the evolved OCXO value for the given Air Interface.
RXGAIN?	Query the Receiver Gain calculation for the supplied frequency. The calculation is performed using the system calibration data.
LOOPPWR?	Query the Transmit Power calculation (loopback mode). The calculation (of the transmitted power based on the receive signal strength) is performed using the system calibration data.
ENG:SYNTH:W RX <reg0> <reg1> <reg2> <reg3> <reg4> <reg5> ENG:SYNTH:R RX ENG:SYNTH:W TX <reg0> <reg1> <reg2> <reg3> <reg4> <reg5> ENG:SYNTH:R TX ENG:SYNTH:W UP <reg0> <reg1> <reg2> <reg3> <reg4> <reg5> ENG:SYNTH:R UP	Write to, or read from the 6 registers in each of the 3 synthesisers in the Flight/Nesie2 Radio
ENG:TEMP?	Read the RF Board Temperature on Flight/Nesie2
ENG:CPLD?	Read the Flight/Nesie2 RF Board Shadow CPLD Register contents.

ENG:FWLD ENG:FWLD?	Load or Query the DSP Firmware.

Miscellaneous Commands

!!POWEROFF!!	Power Down the Transceiver. Issues a poweroff command to the OS.

Information Commands

The ‘INFO’ instrument seems to provide a duplicate to the RX:IFATTN / RX:IFATTN? Pair.

IFATtn <if_attn_parameter> IFATtn?	Set / Query the Rx IF Attenuation Setting

DATEMP Command

The ‘DATEMP’ instrument seems to provide a deprecated version of DA:TEMP?

RFTEMP Command

The ‘RFTEMP’ instrument seems to provide a deprecated version of ENG:TEMP?

DA Commands

The ‘DA’ instrument allows the user to query the ZYNQ’s temperature …

TEMP?	Read the DA Temperature, this uses the ZYNQ XADC measurement.

APP Commands

The ‘APP’ instrument hosts a number of commands that allow the user to communicate with and configure applications running on the DSP. The majority of the commands are handled by the DSP BIST SAP’s General Purpose Application Command, but the 3GCELL command is handled by the 3G Application’s SCPI interface.

Common APP Commands

BIST_MODE_ENABLE BIST_MODE_DISABLE	GP App Command “4” with a “1” or “0” parameter respectively. If the application has registered a handler for the GP App Command, it is called with Command “4” and a single Parameter of either “1” or “0”, Used to inform the application that BIST Mode is active or not. The application can take measures to be quiescent with BIST Mode is active.

GSM APP Commands

PE_START <GSM Channel> <Power Level> PE_STOP PE_DELTA <Power Level Delta>	GP App Command “1”, “2” and “3” respectively. Used in the GSM application for Phase Error Testing against a CMU200 test set. PE_START causes the GSM Application to generate and transmit a GMSK modulated signal, ramping up the output power in the normal way. PE_STOP stops the transmission. PE_DELTA specifies a power step to be applied on every frame, the power will alternate between the power level specified in the PE_START command and that power level less the delta. The Closed Loop Power Control is opened when a non-zero delta is applied. The “up” step is used to trigger the CMU200 to make the phase error measurement.

3G APP Commands

When the first 3GCELL command is encountered the DSP will be loaded with the 3G Application and the application started.

3GCELL CREATE <freq> <attn.> 3GCELL BLOCKERS <index> <freq> <gain> 3GCELL START 3GCELL STOP	Allow the user to generate a 3G Grabber signal on the transmit path. The CREATE and BLOCKERS command allow the user to specify the grabber and blocker configuration. The signal is only transmitted between the START and STOP commands.

Commands / Queries Details

Calibration Data Write

“CAL:WRITE <binary file name>\n”

The binary file will be written to the calibration data EEPROM. The file must be accessible to the python telnet server script and one of two approaches may be used to allow the server to access the file:

- The file is copied to the FlightNesie filesystem present on the unit’s SDCard. This approach can be used when the telnet server is running on the FlightNesie unit.
- If the telnet server is running on a networked computer rather than the FlightNesie unit, the file can be copied on that computer.

Calibration Data Read to Screen

“CAL:READ? <page number>\n”

Read the specified calibration data page from the RF board EEPROM. Any valid page can be read (i.e. numbers 0 .. 63) including the IDENT pages (numbers 0 .. 2), the CAL pages (numbers 3 .. 62) or the CRC page (number 63). Note that the page ranges are implementation defined and may be changed.

If the read operation is successful, a text string will be returned. The text string will contain 256 integer values in hexadecimal and prefixed by ‘0x’ and will be terminated with ‘\n’.

If the read operation is unsuccessful, the returned text string will be of the form:

“ERR: status=x\n”

Where x is an implementation defined integer error number.

Calibration Data Read to File

“CAL:READ? F <file name>\n”

Read the calibration data pages from the RF board EEPROM. The calibration data page number range is implementation defined and is currently pages 3 to 62.

If the read operation is successful, the file will be created and the calibration data will be written in binary format to the specified file which will be overwritten if it already exists. The following string is returned:

‘\n’.

If the read operation is unsuccessful, the returned text string will be of the form:

“ERR: status=x\n”

Where x is an implementation defined integer error number.

Calibration Data Zeroize

“CAL:ZERO\n”

The CAL data pages of the RF board EEPROM will be set to their default contents (all zeroes). The correct CRC is written for each affected page.

The following string is returned:

“\n”

Calibration Read Status

“CAL:STATUS?\n”

The CAL data status is read. If present and valid, the data held in the RF board EEPROM will be used; if this data is not present the hardcoded default values held in the unit will be used; if these values are not present, the calibration data is deemed to be invalid.

If the RF board EEPROM is available and it contains valid calibration data, the following string is returned:

"CAL Status: valid, using nvdata\n"

If the RF board EEPROM is unavailable or uninitialized and the hardcoded default values contain valid calibration data, the following string is returned:

"CAL Status: valid, using hardcoded defaults\n"

If the RF board EEPROM is unavailable or uninitialized and the hardcoded default values do not contain valid calibration data, the following string is returned:

"CAL Status: INVALID\n"

Tx Enable / Disable

“TX:ENABle\n”
“TX:ENABle? \n”
“TX:DISAble\n”
“TX:DISable? \n”

The 2 queries are equivalent to each other, both will generate one of the following responses depending on whether the Transmitter is enabled or disabled:

“ENABLED\n”
“DISABLED\n”

Tx Mute / Unmute

“TX:MUTE\n”
“TX:UNMUte\n”
“TX:MUTE? \n”
“TX:UNMUte? \n”

The 2 queries are equivalent to each other, both will generate one of the following responses depending on whether the Transmitter is muted or unmuted:

```
“MUTED\n”
“UNMUTED\n”
```

Tx Loopback

```
“TX:LOOP ENABle\n”
“TX:LOOP DISAble\n”
“TX:LOOP? \n”
```

The possible responses to the query are:

```
“ENABLED\n”
“DISABLED\n”
```

Tx Test Source

```
“TX:TS:FREQ <ts_freq_in_hz_parameter>\n”
“TX:TS:FREQ? \n”
 “TX:TS:ABS <ts_freq_in_hz_parameter>\n”
“TX:TS:ABS? \n”
 “TX:TS:LEVEL <level in dBFS>\n”
“TX:TS:LEVEL?\n”
“TX:TS:ENABle\n”
“TX:TS:ENABle? \n”
“TX:TS:DISAble\n”
“TX:TS DISAble? \n”
```

Note, the Tx Test Source must be enabled before the frequency is set.

The test signal is provided by a hardware Direct Digital Synthesiser (DDS), on the ZYNQ device. This produces a I/Q sinusoidal signal at the Transmitter Baseband Sampling rate of 103.68 MHz.

The frequency parameter of the “TX:TS:FREQ” command is a text integer value representing the frequency in Hz.

Valid Range : 0 <= TS Frequency <= 100000000 (100 MHz)

The “TX:TS:FREQ?” Query will cause a response which is the frequency in Hz as a text integer, e.g:

```
“0\n”
“100000000\n”
“76543210\n”
```

The “TX:TS:ABS” command is similar to the “TX:TS:FREQ” command except that the frequency parameter is the absolute frequency (in Hz) that will be generated by the DDS. The frequency will be range checked based on the selected Tx frequency band (see the “TX:BAND” command). Attempting to select a frequency that is outside the selected band will be rejected as invalid.

The “TX:TS:ABS?” query returns the frequency selected in the preceding ABS command. The returned frequency string format will follow that for the “FREQ” query.

The dBFS level parameter for the TX:TS:LEVEL command is a text floating point value in the range -96.0 to 0.0 dBFS. The dBFS value is used to generate a scale factor that will be applied to 16 bit I and Q outputs of the DDS. Specifying a level between -96.0 and -100.0 dBFS will result in a scale factor of zero and any level greater than -100.0 will be rejected as invalid.

The “TX:TS:LEVEL?” Query will cause a response which is the current dBFS level as a text floating point value, e.g:

“-32.0\n”
“0.0\n”

Note that the “TX:TS:LEVEL” command sets the DDS level output regardless of the command used to set the DDS output frequency (which can be either of: TX:TS:FREQ, TX:TS:ABS).

The “ENABle? \n” and “DISAble? \n” queries are equivalent, both will generate one of the following responses depending on the state of the test source:

“ENABLED\n”
“DISABLED\n”

Tx Band

“TX:BAND < fwd_rev_parameter> <band_parameter>\n”
“TX:BAND? \n”

Setting the Tx band will:

a) Cause an appropriate LO value to be applied
b) Select the appropriate band filter path.

Valid fwd/rev parameters

“F”
“R”

Valid band parameters:

“GSM850”
“EGSM900”
“DCS1800”
“PCS1900”
“UMTS_1”
“LTE_7”
“LTE_20”
“SPARE”

The query will cause a response with the applied fwd/rev parameter and band parameter, e.g.:

“F GSM850\n”
“R LTE_7\n”

Tx Attenuation

“TX:ATTN <attn_parameter>\n”
“TX:ATTN? \n”

Tx Attenuation is an integer dB value between 0 and 15.

The query causes a response with an integer value between 0 and 15, e.g.:

“0\n”
“15\n”
“8\n”

Tx Signal Strength

“TX:SIGS? \n”

Query causes a response that contains the Tx Signal Level measurement, an integer representing the raw ADC reading.

TBD – Add a parameter so that an dB value in dBm or a raw ADC reading value can be returned.

Tx Port

“TX:PORT <port_parameter>\n”
“TX:PORT? \n”

Valid port parameters are:
“PORT1”
“PORT2”
“PORT3”
“PORT4”

The query will cause a response with the port selected, e.g.:
“PORT1\n”
“PORT2\n”
“PORT3\n”
“PORT4\n”

Tx DDS

“TX:DDS:FREQ <DDS Freq in Hz>\n”
“TX:DDS:FREQ?\n”
“TX:DDS:ENABle\n”
“TX:DDS:ENABle?\n”
“TX:DDS:DISAble\n”
“TX:DDS:DISAble?\n”

Allows control of the DDS test signal. Seems to duplicate TX:TS functionality.

Tx DAC

Flight/Nesie2 AD9122 DAC Register read and write.

“TX:DAC:Write (<reg_addr> <reg value>) {1,}\n”
“TX:DAC:Read <reg addr}{1,} \n”

i.e 1 or more <reg addr>,<reg value> pairs for the write, and 1 or more <ref addr> for the read.

Tx Freq

Covert/Tactical Only, for Flight/Nesie2 the band selection controls the synthesiser LO values. Set / Get the AD9361 Tx LO value.

“TX:FREQ <TX LO Freq in Hz >\n”
“TX:FREQ? \n”

Tx PA

Covert Only. Enable/Disable PA, or query its state

“TX:PA ENABle\n”
“TX:PA ENABle?\n”
“TX:PA DISAble \n”
“TX:PA DISAble? \n”

Tx BW

Covert/Tactical Only. Set / Query the AD9361 Tx Bandwidth value

“TX:BW <Tx Bandwidth in Hz >\n”
“TX:BW?\n”

Tx Duplexer

Tactical Only. Set / Query the RF Board Tx Duplexer

“TX:DUP <Duplexer_parameter>\n”
“TX:DUP?\n”

Valid Duplexer_parameter values are:
“NONE”
“DUP100”
“DUP101”
“DUP102”
“FIL100”
“FIL101”
“FIL102”
“DUP103”
“DUP104”
“DUP105”
“DUP106”

“DUP107”
“DUP108”

These refer to the Duplexer/Filter sites on the 860 Board Schematic. Since we can have different filter bands in a particular site it make sense to refer to physical location on the board.
We can select the RF input to a duplexer independent to the RF Output from a duplexer, “NONE” selects a different duplexer input and output for isolation.

The query will give a response as to which path site is selected, for instance “NONE\n” or “FIL100\n”

Tx Forward/Reverse

Tactical Only. Set / Query the RF Board Tx Forward/Reverse Line

“TX:FORREV <Line_parameter>\n”
“TX: FORREV?\n”

Valid Line_parameter values are:

“SET”
“CLEAR”

Each Tx Duplexer has two outputs commonly called Tx and Rx and typically this is as viewed on a handset, so Tx is the Uplink/Reverse band and Rx the Downlink/Forward band.

The Line_parameter controls whether the Tx or Rx path is selected. Unfortunately “SET” can give Tx or Rx depending on the Duplexer/Filter fitted to a particular site.

The query will give a “SET\n” or “CLEAR\n” response depending on if the line is high or low.

Tx PA Path

Tactical Only. Set / Query the PA Board RF Path:

“TX:PAPATH <PA_parameter>\n”
“TX: PAPATH?\n”

Valid PA_parameter values are:
“PA_OFF”
“PA_LOW”
“PA_HIGH”
“PA_3GHZ”

“PA_OFF” means the RF Signal is not routed to or from any PA, the others mean that RF Switches on the board route the signal from the RF in connector to the chosen PA and then route that PA’s output to the RF out connector.

The query will give the currently selected path e.g. “PA_3GHZ\n” or “PA_OFF\n”.

Note this command does not change the enable state of any of the PAs, there is a separate command for that. This allows us to sequence things appropriately.

Tx PA Enable

Tactical Only. Enable/Disable a Particular PA and Query the state.

“TX:PAEN <PA_parameter>\n”
“TX: PAEN?\n”

Valid PA_parameter values are:
“PA_OFF”
“PA_LOW”
“PA_HIGH”
“PA_3GHZ”

“PA_OFF” means no PAs are enabled, the other parameters cause the selected PA to be enabled.

The query will give the currently enabled PA e.g. “PA_3GHZ\n” or “PA_OFF\n” if no PA is enabled

Tx PA Power

Tactical Only. Measure the Tx Power on the PA board

“TX:PAPWR?\n”

Uses the AD5593R device on the PA board to measure the Tx power via an LT5538I RF Power Detector.

The AD5593R ADC gives a 12 bit result. Scaling to power in dBm TBD, may need to be calibrated.

Tx PA Temp

Tactical Only. Measure the temperature on the PA board

“TX:PATEMP?\n”

Uses the temperature sensor on the AD5593R device on the PA board to measure the Temperature.

The AD5593R ADC gives a 12 bit result, this is converted in S/W into a value in degrees C. May need to be calibrated.

Rx Enable / Disable

“RX:ENABle\n”
“RX:ENABle? \n”
“RX:DISAble\n”
“RX:DISable? \n”

The 2 queries are equivalent to each other, both will generate one of the following responses depending on whether the Receiver is enabled or disabled:

“ENABLED\n”
“DISABLED\n”

Rx Band

“RX:BAND < fwd_rev_parameter> <band_parameter>\n”
“RX:BAND? \n”

Setting the Rx band will:

a) Select the appropriate band filter path.
b) Cause an appropriate LO value to be applied
c) If the “High Power” LNA selection is in force, pick the appropriate LNA path

Fwd/rev and band parameters are as for the Tx Band Command.

The query will cause a response with the applied fwd/rev parameter and band parameter, e.g.:

“R EGMS900\n”
“F UMTS_1\n”

Rx LNA

“RX:LNA <lna_parameter>\n”
“RX:LNA?\n”

Valid lna parameters are:

“BYPASS”
“LOW_NOISE”
“HIGH_POWER”

When “HIGH_POWER” is selected the actual LNA used will depend on the Rx Band in force.

The query will cause a response to be sent that will report the LNA in force:

“BYPASS\n”
“LOW_NOISE\n”
“HIGH_POWER\n”

Rx Signal Strength

“RX:RSSI? <rssi_type_parameter>\n”

“RX:RSSI? INPUT <freq in Hz >\n”

Valid RSSI type parameters are:

“RF”
“IF”

“BB”
“OF”
”INPUT”

“RF” gives the measurement after the LNAs; “IF” gives the measurement after the mixer; “BB” gives the measurement based on an RMS calculation on a captured block of baseband data; “OF” gives the current latched settings of the overflow flags; “INPUT” uses the BB RMS calculation and calibration information, along with knowledge of the Rx LNA, Rx Band, Rx Gain and Rx IF Attenuation in force to estimate a signal level at the input to the transceiver

The RF and IF responses will currently be the raw ADC reading.

[TBD add a parameter so a dB value or raw value can be requested].

The BB response will be a dBFS value, i.e. dB re the RMS measurement for a full-scale sinusoidal signal.

The OF response will be a 16-bit hex value with leading “0x” representing the current overflow flag values. Note currently only 6 overflow flags are defined so the top 10 bits of the word should be 0, e.g. :

“0x0000\n”
“0x003F\n”
“0x001A\n”

Note that the act of reading the latched overflow flags will clear the latches.

The INPUT response will be an estimated dBm value.

Rx Gain

“RX:GAIN <gain_parameter>\n”
“RX:GAIN?\n”

Valid gain parameters are:

“15”
“0”
“-10”
“OPEN”

These represent 4 different paths in the receiver in front of the mixer: a 15 dB amplifier; a direct connection; a 10 dB attenuation; and open circuit i.e. no connection.

The query will generate a response that contains the gain parameter in force.

Rx IF Attenuation

“RX:IFATtn <if_attn_parameter>\n”
“RX:IFATtn?\n”

Sets the IF attenuation before the ADC.

The attenuator has a range of 0 to 31.5 dB in 0.5 dB steps.

Valid if attn parameters are:

"0.0"
"0.5"
"1.0"
"1.5"
"2.0"
"2.5"
….
"29.0"
"29.5"
"30.0"
"30.5"
"31.0"
"31.5"

The query will generate a response with the attenuation parameter in force.

Rx Block Capture

"RX:CAPT? <capture_length_parameter>\n"

The Rx Block Capture query causes the capture of a 65536 (64K) or 262144 (256K) block of baseband Rx samples.

Valid capture length parameters are:

"8K"	(Flight/Nesie2)
"16K"	(Covert/Tactical)

ADC Data Test (Flight/Nesie2 Only)

Also a "PRS" subcommand may be used, i.e.

"RX:CAPT? PRS\n"

This puts the AD9467 ADC into 23 bit pseudo-random sequence mode, and an 8K block is captured. The DSP attempts to synchronise to the sequence and then predict subsequent sample values, these are checked against the values read from the receiver. The number of samples needed before the DSP has synchronised and the number of samples that don't match the predicted values are sent back in the response:

"Sync Count 3; Failure Count 8"

The above response represents a success, the DSP declares it is synchronised on the 3rd sample, and after sync there are 8 samples that don't match. These are the last 8 samples of the 8K. This is due to a hardware limitation, data is collected on the hardware in a large FIFO, and we can only store 8K – 8 samples in it. So that last 8 samples of the 8K are repeats of the sample before. For the 8K block capture the code zeros these samples.

Transceiver Data Test (Covert/Tactical Only)

On Covert/Tactical we can control the data mode of the Rx path on the AD9361 Transceiver, along with the collection bandwidth.

```
"RX:CAPT? AD9361_TEST <Enable/Disable Control> <AD9361 Test Freq>
    <AD9361 Test Level> <FPGA Data Bandwidth>\n"
```

<Enable/Disable Control> should be either "ENABLE" or "DISABLE", this enables or disables the test mode.

<AD9361 Test Freq> is 0..3, A sinusoid is generated with a frequency of 30.72 MHz x (1 + <AD9361 Test Freq>)/32

<AD9361 Test Level> is 0..3, the sinusoid generated will be FS / $2^{\text{<AD9361 Test Level>}}$

<FPGA Data Bandwidth> can be "30M72" or "3M2768". If "30M72" is selected, the raw data from the AD9361 will be collected. If "3M2768" the 30.72 MHz data from the AD9361 is filtered and decimated in the FPGA down to a 3.2768 MHz rate with bandwidth 2 MHz. The data to the operational DDC is unaffected.

Note that this control is persistent, and can be used when the test mode is disabled, so we can capture 3.2768 MHz data off-air.

The response to a data capture consists of multiple lines of data, the number of lines depending on the block size requested, and textual header and footer are also sent, i.e.:

```
"BLOCK_DATA_STARTS\n"
    8192 or 16384 lines of RAW DATA
"BLOCK_DATA_ENDS\n"
```

Each RAW DATA line consists of 4 Hexadecimal digits:

```
"WXYZ\n"
```

The 4 hex digits should be interpreted as a signed 16-bit value.

Note that the transfer of the data can take some time.

Rx ADC Test Mode

```
“RX:TEST ENABLE <test_code>\n”
“RX:TEST DISABLE 0\n”
```

“ENABLE” Sets the Analog to Digital Converter to produce the specified test pattern as its output digital words.

Test codes from 0 to 7 are valid. See table 10 of the AD9467 data sheet for definitions of the test codes (ADC data, midscale fixed, +ve full scale, -ve full scale, alternating checkerboard, Long PRS, short PRS, alternating 1s and 0s).

“DISABLE” resets the convertor back to operational mode.

Once in the test mode the ADC data can be inspected via the RX:CAPT? Query.

Rx LNA RSSI Enable / Disable

```
“RX:LRSSIEN\n”
“RX:LRSSIEN?\n”
“RX:LRSSIDIS\n”
“RX:LRSSIDIS?\n”
```

The 2 queries are equivalent to each other, both will generate one of the following responses depending on whether the LNA RSSI measurement is enabled or disabled:

```
“ENABLED\n”
“DISABLED\n”
```

Rx FPGA Read

```
“RX:FPGA:R <register address>\n”
```

This command is used to read the contents of a register in the ADC control block in the DA board FPGA. The register is specified using its byte address offset in the memory mapped interface to the block. The ADI reference design documentation should be consulted for information on the block register addresses and their expected contents.

If the command is successful, the register contents are returned as a string containing a hexadecimal value:

```
“0x1234\n”
```

Otherwise, the command returns a string prefixed with “ERR:” followed by information on the nature of the error that occurred.

Rx FPGA Write

```
“RX:FPGA:W <register address> <value>\n”
```

This command is used to write to a register in the ADC control block in the DA board FPGA. The register is specified using its byte address offset in the memory mapped interface to the block. The ADI reference design documentation should be consulted for information on the block register addresses and their expected contents.

If the command is successful, the register contents are reread and returned as a string containing a hexadecimal value:

“0x1234\n”

Otherwise, the command returns a string prefixed with “ERR:” followed by information on the nature of the error that occurred.

Rx ADCPWR

Flight/Nesie2 Only

```
“RX:ADCPWR:DOWN\n”
“RX:ADCPWR:UP\n”
```

Power Down / Power up the AD9467 ADC device.

Rx ADCREAD

Flight/Nesie2 Only

```
“RX:ADCREAD  <hex register address>\n”
```

Read the specified register on the AD9467 ADC device.

Rx ADCWRITE

Flight/Nesie2 Only

```
“RX:ADCWRITE <hex register address> <hex 8-bit register value> \n”
```

Write the specified register on the AD9467 ADC device with the specified value.

Rx Freq

Covert Only

```
“RX:FREQ <Freq Value in Hz> \n”
“RX:FREQ? \n”
```

Set / Read the AD9361 Rx LO value.

Rx BW

Covert Only

```
“RX:BW <Bandwidth in Hz> \n”
“RX:BW? \n”
```

Set/ Read the AD9361 Rx Bandwidth

Rx Duplexer

Tactical Only. Set / Query the RF Board Rx Duplexer

“RX:DUP <Duplexer_parameter>\n”
“RX:DUP?\n”

Valid Duplexer_parameter values are:

“NONE”
“DUP300”
“DUP301”
“DUP302”
“FIL300”
“FIL301”
“FIL302”
“DUP303”
“DUP304”
“DUP305”
“DUP306”
“DUP307”
“DUP308”

These refer to the Duplexer/Filter sites on the 860 Board Schematic. Since we can have different filter bands in a particular site it make sense to refer to physical location on the board.
We can select the RF input to a duplexer independent to the RF Output from a duplexer, “NONE” selects a different duplexer input and output for isolation.

The query will give a response as to which path site is selected, for instance “NONE\n” or “FIL300\n”

Rx Forward/Reverse

Tactical Only. Set / Query the RF Board Rx Forward/Reverse Line

“RX:FORREV <Line_parameter>\n”
“RX: FORREV?\n”

Valid Line_parameter values are:

“SET”
“CLEAR”

Each Rx Duplexer has two outputs commonly called Tx and Rx and typically this is as viewed on a handset, so Tx is the Uplink/Reverse band and Rx the Downlink/Forward band.

The Line_parameter controls whether the Tx or Rx path is selected. Unfortunately “SET” can give Tx or Rx depending on the Duplexer/Filter fitted to a particular site.

The query will give a “SET\n” or “CLEAR\n” response depending on if the line is high or low.

Rx Input

Tactical Only. Set / Query Rx RF Input Type

“RX:INPUT <Input_parameter>\n”
“RX: INPUT?\n”

Valid Input _parameter values are:

“ANT”
“DAISY”

The Tactical units are expected to be used in a set, the “first” unit is connected to an antenna, the subsequent ones are fed from the RF output of the previous one, i.e. Daisy-Chained together. The first unit must have an LNA enabled, the subsequent units have a different amplifier enabled, presumably to allow for loses in the daisy-chaining link.

The query will give a “ANT\n” or “DAISY\n” response depending on which input mode is in force.

OCXO

“OXCO <ocxo_parameter>\n”
“OCXO?\n”

This command sets the OCXO Digipot which has a 10 bit setting range, i.e. 0 to 1023

OCXO parameter is the required setting as a text integer.

The query causes a response to be sent that contains the value in force, e.g.:

“0\n”
“1023\n”
“512\n”

LADON

Ladon Present Query

"LADON:PRES?\n"

If the command is successful, the following string is returned:

“Ladon status= OK Return=True\n”

Otherwise, the command returns a string of the form:

“ERR: Ladon status= CMD_ERROR”

Ladon Initialised Query

"LADON:INIT?\n"

If the command is successful, the following string is returned:

“Ladon status= OK Return=True\n”

Otherwise, the command returns a string of the form:

“ERR: Ladon status= CMD_ERROR”

Ladon PA Attenuation Query

"LADON:PATTN?\n"

If the command is successful, the following string is returned:

“Ladon status= OK Return=10dB\n”

Otherwise, the command returns a string of the form:

“ERR: Ladon status= CMD_ERROR”

Ladon Set PA Attenuation Command

"LADON:PATTN <attenuation in dB>\n"

The <attenuation in dB> is an integer in the range 0 .. 15 with 0 representing 15dB attenuation and 15 representing 0dB attenuation.

If the command is successful, the following string is returned:

“\n”

Otherwise, the command returns a string of the form:

“ERR: Ladon status= CMD_ERROR”

Ladon Forward Power Query

"LADON:FWDP?\n"

If the command is successful, the following string is returned:

“Ladon status= OK Return=10dB\n”

Otherwise, the command returns a string of the form:

“ERR: Ladon status= CMD_ERROR”

Ladon Set Tx Frequency Command

"LADON:TXFREQ <frequency in MHz>\n"

The <frequency in MHz> parameter is the frequency in MHz, e.g. “992”

If the command is successful, the following string is returned:

“\n”

Otherwise, the command returns a string of the form:

“ERR: Ladon status= CMD_ERROR”

Ladon Enable/Disable PA Command

"LADON:PENA <on/off parameter>\n"

The <on/off parameter> is one of “ON” or “OFF”.

If the command is successful, the following string is returned:

“\n”

Otherwise, the command returns a string of the form:

“ERR: Ladon status= CMD_ERROR”

Ladon Status and Error Query

"LADON:STATUS?\n"

A string of the following form is returned:

“Status Register=1234 Error Register=5678\n”

Ladon Set Tx Filter Band Command

"LADON:TXFBAND <filter band>\n"

The <filter band> parameter is one of:

"NONE"

"WIDEBAND"

"EGSM900_FWD"

"EGSM900_REV"

"GSM850_FWD"

"GSM850_REV"

"DCS1800_FWD"

"DCS1800_REV"

"PCS1900_FWD"

"PCS1900_REV"

"IDEN850_FWD"

"IDEN850_REV"

"UMTS1_FWD"

"UMTS1_REV"

If the command is successful, the following string is returned:

“\n”

Otherwise, the command returns a string of the form:

“ERR: Ladon status= CMD_ERROR”

Ladon Tx Filter Band Query

"LADON:TXFBAND? \n"

If the command is successful, the following string is returned:

“Ladon status=OK <list of filter bands, in the same form as that given in the set command>\n”

Otherwise, the command returns a string of the form:

“ERR: Ladon status= CMD_ERROR”

Ladon Set Rx Filter Band Command

"LADON:RXFBAND <filter band>\n"

The <filter band> parameter is one of:

"NONE"

"WIDEBAND"

"EGSM900_FWD"

"EGSM900_REV"

"GSM850_FWD"

"GSM850_REV"

"DCS1800_FWD"

"DCS1800_REV"

"PCS1900_FWD"

"PCS1900_REV"

"IDEN850_FWD"

"IDEN850_REV"

"UMTS1_FWD"

"UMTS1_REV"

If the command is successful, the following string is returned:

“\n”

Otherwise, the command returns a string of the form:

“ERR: Ladon status= CMD_ERROR”

Ladon Rx Filter Band Query

"LADON:RXFBAND?\n"

If the command is successful, the following string is returned:

“Ladon status=OK <list of filter bands, in the same form as that given in the set command>\n”

Otherwise, the command returns a string of the form:

“ERR: Ladon status= CMD_ERROR”

Ladon Enable/Disable Rx Common Amplifier Command

"LADON:RXCMNAMP <on/off parameter>\n"

The <on/off parameter> is one of “ON” or “OFF”.

If the command is successful, the following string is returned:

“\n”

Otherwise, the command returns a string of the form:

“ERR: Ladon status= CMD_ERROR”

Ladon Enable/Disable Rx Channel Amplifier Command

"LADON:RXCHANAMP <on/off parameter>\n"

The <on/off parameter> is one of “ON” or “OFF”.

If the command is successful, the following string is returned:

“\n”

Otherwise, the command returns a string of the form:

“ERR: Ladon status= CMD_ERROR”

Ladon Rx Gain Query

"LADON:RXGAIN? <receiver> <frequency in MHz>\n"

The <receiver> parameter is one of “RX1” or “RX2”

The <frequency in MHz> parameter is the frequency in MHz, e.g. “992”

If the command is successful, the following string is returned:

“\n”

Otherwise, the command returns a string of the form:

“ERR: Ladon status= CMD_ERROR”

Ladon Select Modulation Command

"LADON:MOD <modulation selector>\n"

The <modulation selector> is the 16bit value expected by the Ladon SelectModulation() function.

If the command is successful, the following string is returned:

“\n”

Otherwise, the command returns a string of the form:

“ERR: Ladon status= CMD_ERROR”

ID:RFSN?

“ID:RFSN?\n”

The query causes a response to be sent that contains the RF Board serial number currently in force, e.g.:

“1234567\n”

ID:DASN?

“ID:DASN?\n”

The query causes a response to be sent that contains the DA Board serial number currently in force, e.g.:

“1234567\n”

ID:DASN

“ID:DASN <serial number>\n”

The command sets the DA Board serial number to the <serial number> parameter. The following string is returned:

“\n”

ID:TRXSN?

"ID:TRXSN?\n"

The query causes a response to be sent that contains the Transceiver serial number currently in force, e.g.:

"1234567\n"

ID:TRXSN

"ID:TRXSN <serial number>\n"

The command sets the Transceiver serial number to the <serial number> parameter. The following string is returned:

"\n"

STATUS:RF:TEMP?

"STATUS:RF:TEMP?\n"

This command queries the RF Board temperature measurement.

The query causes a response to be sent that contains the last measured temperature in degrees Celsius, e.g.:

"37\n"
"-5\n"

STATUS:RF:POST

"STATUS:RF:POST\n"

This command runs the RF Board Power On Self Tests and returns the set of test results. The results are presented as a comma separated list of the form:

"Test: <test name> Status: [PASSED|FAILED] Info: <informational text>, … \n"

For example:

"Test: DAC_path_test Status: PASSED Info: OK, Test: ADC_path_test Status: FAILED Info: device_setup_error\n"

The query returns the set of test results from the last run (if any). If the tests have not been run the query responds with:

"Test: None Status: Unknown Info: No tests have been run\n"

STATUS:DA:TEMP?

“STATUS:DA:TEMP?\n”

This command queries the DA Board temperature measurement.

The query causes a response to be sent that contains the last measured temperature in degrees Celsius, in following format:

“XADC: T=47.49C\n”

STATUS:DSP

“STATUS:DSP\n”

Requests the status of the BIST SAP connection from the DSP.

It responds with “DSP Status: OK\n” if no problems are found, else an error string is returned.

ENG:OCXO

“ENG:OCXO <air interface> <evolved value>\n”

The command sets the OCXO evolved value for the given air interface. The supported air interfaces are “GSM”, “UMTS” and “LTE”. The evolved value is a decimal integer in the range 0..1023 or the string “RST” which commands the system to reset the evolved value for the air interface to “not set”. The following string is returned:

“\n”

ENG:OCXO?

“ENG:OCXO? <air interface>\n”

The query causes a response to be sent that contains the OCXO evolved value currently in force, e.g.:

“1023\n”

ENG:RXGAIN?

“ENG:RXGAIN? <frequency in Hz>\n”

The query causes a response to be sent that contains the calculated Receiver gain for the supplied frequency. The query makes use of other settings that are in force, i.e. Rx band, Rx LNA, Rx gain and Rx IF Attenuation, but as the gain can vary across a band, the frequency must be supplied. An example response is:

“236\n”

The returned gain is in tenths of a dBm so the example is equivalent to 23.6dBm.

ENG:LOOPPWR?

“ENG:LOOPPWR? \n”

The query causes a response to be sent that contains the calculated Transmit power and is valid when the RF board is in loopback mode. The query makes use of settings that are in force, i.e. Tx band and Rx IF Attenuation and is based on the system calibration data. An example response is:

“236\n”

The returned power is in tenths of a dBm so the example is equivalent to 23.6dBm.

ENG:SYNTH

```
“ENG:SYNTH:W RX <reg0> <reg1> <reg2> <reg3> <reg4> <reg5>\n”
“ENG:SYNTH:R RX\n”
“ENG:SYNTH:W TX <reg0> <reg1> <reg2> <reg3> <reg4> <reg5>\n”
“ENG:SYNTH:R TX\n”
“ENG:SYNTH:W UP <reg0> <reg1> <reg2> <reg3> <reg4> <reg5>\n”
“ENG:SYNTH:R UP\n”
```

Read or write the 6 32-bit synth register values for one of the 3 synths used in the Flight/Nesie2 Radio

ENG:TEMP?

“ENG:TEMP?\n”

Read the temperature of the RF board temperature sensor, Flight/Nesie2 only.

ENG:CPLD?

“ENG:CPLD?\n”

Read the contents of the set of RF board CPLD registers, Flight/Nesie2 only. We cannot read directly from the CPLD as it doesn’t support readback. What we see here is the contents of a set of shadow registers held on the CPLD I/F block in the FPGA. That should reflect the contents of the CPLD registers.

ENG:FWLD

“ENG:FWLD\n”
“ENG:FWLD?\n”

Load or query the “firmware”. “FWLD” generates a command to the nesie-driver to load the current DSP firmware. “FWLD?” queries the nesie-driver as to what firmware is loaded.

POWEROFF

“!!POWEROFF!!”

This commands causes the transceiver unit to power down.

A “poweroff” command is sent to the Linux OS.

APP:BIST_MODE…

“APP:BIST_MODE_ENABLE\n”
“APP:BIST_MODE_DISABLE\n”

Inform the Application that BIST Mode Is enabled or disabled. If the application has registered a GP Application Command handler function with the BIST Library functionality, then the handler will be called by the BIST library with command “4” and parameter “1” for enable or “0” for disable.

APP:PE…

“APP:PE_START <GSM Channel> <Power Level>\n”
“APP:PE_STOP\n”
“APP:PE_DELTA <Power Level Delta>\n”

GSM Application Phase Error Commands. We use these with a CMU200 to make GSM Phase Error Measurements. When started the GSM application will generate a GMSK modulated pseudo-random transmit data stream. It will also configure the transmitter and any Ladon present to transmit it the signal at the specified power level.

GSM Channel is an ARFCN

Power Level is an integer from 0 to 31. 31 is full power, thence down in 2 dB steps to PL of 1 (approximately 62 dB down from full power), PL of 0 is “Off”.

Power Level Delta is the delta in power levels, hence 3 means a 6 dB delta. The default is 0. When a non-zero delta is in force the application will toggle the transmit power between the power level set in the PE start command and that less the delta. The change in power is used to trigger the CMU200 to make a phase error measurement.

APP:3GCELL

“APP:3GCELL CREATE <freq> <attn.>\n”
“APP:3GCELL BLOCKERS <index> <freq> <gain>\n”
“APP:3GCELL START\n”
“APP:3GCELL STOP\n”

3G Application 3G Cell Commands.

We use Create to specify some of the parameters for a 3G grabber configuration

Freq is in 100 kHz units, i.e. MHz x 10, and should be an allowable 3G Channel centre frequency.

Attn is in cB, i.e. dB x 10, should be in the range 0 to 500.

When system will generate the following SCPI commands for the 3G application when it sees “APP:3GCELL CREATE <freq> <attn>\n”:

```
TX:CONF:MODE 3
TX:CONF:FREQ <freq>
TX:CONF:PSCHGAIN  2733
TX:CONF:SSCHGAIN  2733
TX:CONF:PCCPCHGAIN  2730
TX:CONF:SCCPCHGAIN  2730
TX:CONF:AICHGAIN  1368
TX:CONF:CPICHGAIN  5447
TX:CONF:ATTN <attn>
TRX:CONF:PRIMARYSCRAMBLINGCODE 17
```

The fixed channel gains, scrambling code etc. are deemed suitable for a representative grabber configuration.

“APP:3GCELL BLOCKERS <index> <freq> <gain>\n” is translated to:

```
TX:CONF:BLOCKINGCELL <index>,<freq>,<gain>
```

Index should be a value 0..3. The application may not support 4 independent blockers though.

Freq is the required blocker frequency in Hz, it needs to be a valid frequency for the 3G cell in use.

Gain is not currently used by the application, 0 should be used for the time being.

To disable an active blocker use “APP:3GCELL BLOCKERS <index> 0 0\n”

“APP:3GCELL START\n” and “APP:3GCELL STOP\n” are translated to “TRX:START” and “TRX:STOP” respectively. Between the start and the stop the configured cell should be transmitted. The configuration for both cell and blockers if any needs to be established before the start. If a change is made after the start the on-air configuration will not change until the cell is stopped and restarted.

Others [TBD]