KT-000-0202-00 Rev A.1 - Test Interface Board Commands

# Existing Commands

## Get Hardware Configuration Info ("$HCI")

Existing command still valid.

## Reset Hardware Configuration Info ("#RHCI")

Existing command still valid.

## Set Hardware Configuration Info ("#SHCI")

Existing command still valid.

## Get ADC Reading ("$ADC")

Existing command still valid except we now have two ADC devices to read from. I don’t know if it’s best to treat this as 16 logical channels or as two separate devices, line $ADC1 and $ADC2?

Maybe for temperature you could read from both devices and return the higher of the two?

## Get Board ID ("$BID")

This needs to change as some of these I/O lines have been repurposed. We now have just two ID bits.

## Set DDS Attenuator ("#DATT")

Existing command still valid.

## Set Tx Fine Attenuator ("#TFAT")

This needs to change as the attenuator device uses 7-bit control giving a range of 0-31.75 dB in 0.25 dB steps.

## Set Tx Course Attenuator ("#TCAT")

This needs to change as the attenuator device uses 1-bit control for either 0 dB or 20 dB setting (same as #DATT).

## Set Rx LNA Bypass ("#RLBY")

Existing command still valid, but… what the hardware actually does depends on the state of a new LNA bypass control “source” signal coming from the Transceiver (suggest we implement separate commands to control the two XCVR LNA controls – see further down). The truth table is:

|  |  |  |  |
| --- | --- | --- | --- |
| **XCVR\_CTRL\_F.LNA\_SRC** | **XCVR\_CTRL\_F.LNA\_BYP** | **RX\_CTRL.LNA\_BYP** | **LNA State** |
| 0 | 0 | 0 | LNA Path |
| 0 | 0 | 1 | LNA Bypass |
| 0 | 1 | 0 | LNA Path |
| 0 | 1 | 1 | LNA Bypass |
| 1 | 0 | 0 | LNA Path |
| 1 | 0 | 1 | LNA Path |
| 1 | 1 | 0 | LNA Bypass |
| 1 | 1 | 1 | LNA Bypass |

## Set Rx Preselector ("#RXPS")

Please can we change this command description to: **Set Rx Path (“#RXP”)**

Underlying command functionality needs to change to suit this truth table:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RX\_CTRL.PATH3** | **RX\_CTRL.PATH2** | **RX\_CTRL.PATH1** | **RX\_CTRL.PATH0** | **Selected Band** |
| **0** | **0** | **0** | **0** | RX0: 400-650 MHz |
| **0** | **0** | **0** | **1** | RX1: 550-1050 MHz |
| **0** | **0** | **1** | **0** | RX2: 950-1450 MHz |
| **0** | **0** | **1** | **1** | RX3: 1350-2250 MHz |
| **0** | **1** | **0** | **0** | RX4: 2150-3050 MHz |
| **0** | **1** | **0** | **1** | RX5: 2950-4650 MHz |
| **0** | **1** | **1** | **0** | RX6: 4550-6000 MHz |
| **0** | **1** | **1** | **1** | RX7: 5700-8000 MHz |
| **1** | **0** | **0** | **0** | OBS0: 400-6000 MHz |
| **1** | **0** | **0** | **1** | OBS1: 550-1050 MHz |
| **1** | **0** | **1** | **0** | OBS2: 950-1450 MHz |
| **1** | **0** | **1** | **1** | OBS3: 1350-2250 MHz |
| **1** | **1** | **0** | **0** | OBS4: 2150-3050 MHz |
| **1** | **1** | **0** | **1** | OBS5: 2950-4650 MHz |
| **1** | **1** | **1** | **0** | OBS6: 4550-6000 MHz |
| **1** | **1** | **1** | **1** | OBS7: 5700-8000 MHz |

Where RX\_CTRL.PATH3 is a repurposed I/O line on the NTM Interface connector. **Note: RX\_CTRL.PATH3 is 1V8 logic level at the NTM Interface!**

## Set Tx Path ("#TXP")

Underlying command functionality needs to change to suit this truth table:

|  |  |  |  |
| --- | --- | --- | --- |
| **TX\_CTRL.PATH2** | **TX\_CTRL.PATH1** | **TX\_CTRL.PATH0** | **Selected Band** |
| **1** | **1** | **1** | DDS0: 400-1500 MHz |
| **0** | **0** | **0** | DDS1: 1480-1880 MHz |
| **0** | **0** | **1** | DDS2: 1850-2250 MHz |
| **0** | **1** | **0** | DDS3: 2250-3000 MHz |
| **0** | **1** | **1** | DDS4: 2400-3400 MHz |
| **1** | **0** | **0** | DDS5: 3400-4600 MHz |
| **1** | **0** | **1** | DDS6: 4600-6000 MHz |
| **1** | **1** | **0** | DDS7: 5700-8000 MHz |

Note that TX\_CTRL.PATH3 is used in a new command to select one of two XCVR bands.

## Set Rx Enable ("#RXEN”)

Existing command still valid.

## Set Tx Enable ("#TXEN”)

Existing command still valid.

## Set Transceiver Reset ("#XRST”)

Existing command still valid.

## Get Transceiver Vendor ID ("$XVID”)

Existing command still valid.

## Get Transceiver GP Interrupt ("$GINT”)

Existing command still valid.

# New Commands

## Set Transceiver Tx Path ("#XTXP"?)

Uses the TX\_CTRL.PATH3 input.

The truth table is:

|  |  |
| --- | --- |
| **TX\_CTRL.PATH3** | **Selected Band** |
| **0** | XCVR0: 400-6000 MHz |
| **1** | XCVR1: 5700-8000 MHz |

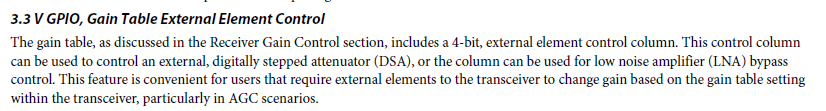
## Set Rx LNA Bypass Source ("#RLBS"?)

This involves configuring the Transceiver pin “GPIO\_3P3\_1” via SPI2 bus.

A ‘0’ means the external RX\_CTRL.LNA\_BYP signal has control of the LNA Bypass function. A ‘1’ means the external signal is ignored, and the Transceiver pin “GPIO\_3P3\_0” (XCVR\_CTRL\_F.LNA\_BYP) has control.

Please default this pin to ‘0’ as part of Test Jig initialisation.

Some additional information on configuring these I/O pins, that Rich managed to find:

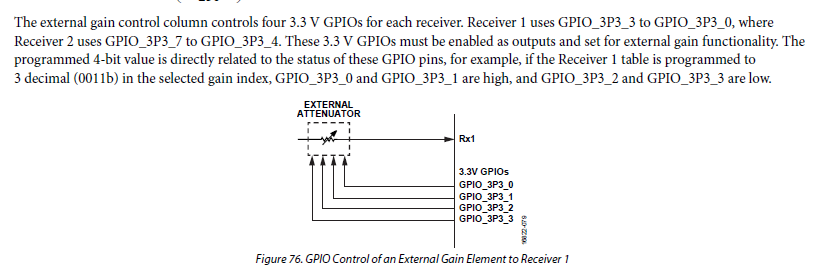




*We’re using Receiver 1 so these bits are GPIO\_3P3\_3:0. It’s referring to a table we can setup so I think this means we could use just the LSB to control LNA.*

*We don’t have spare IO to get this back to the FPGA, we don’t really need to see it in FPGA, we just want to enable LNA using either manual control from FPGA or using that bit out of the transceiver.*

*We can control the sense of the GPIO outputs as values are written into a table in the transceiver:*



## Set Transceiver Rx LNA Bypass ("#RLBX"?)

This involves configuring the Transceiver pin “GPIO\_3P3\_0” via SPI2 bus.

A ‘0’ means no bypass, non-zero to bypass the LNA. What the hardware actually does depends on the state of the XCVR\_CTRL\_F.LNA\_SRC signal (see previous command).

Please default this pin to ‘0’ as part of Test Jig initialisation.

The truth table is (as per *Set Rx LNA Bypass* command):

|  |  |  |  |
| --- | --- | --- | --- |
| **XCVR\_CTRL\_F.LNA\_SRC** | **XCVR\_CTRL\_F.LNA\_BYP** | **RX\_CTRL.LNA\_BYP** | **LNA State** |
| 0 | 0 | 0 | LNA Path |
| 0 | 0 | 1 | LNA Bypass |
| 0 | 1 | 0 | LNA Path |
| 0 | 1 | 1 | LNA Bypass |
| 1 | 0 | 0 | LNA Path |
| 1 | 0 | 1 | LNA Path |
| 1 | 1 | 0 | LNA Bypass |
| 1 | 1 | 1 | LNA Bypass |

## Get Synth Lock Detect ("$SYNLD"?)

Returns the state of the TX\_CTRL.SYNTH\_LD pin on the NTM Interface connector. **Note: this signal is 1V8 logic level at the NTM Interface!**

## Set Synth Output Frequency ("#SYNFQ"?)

Uses SPI1 bus to configure the ADF5356 registers. **Note: the SPI.nSYNTH\_CS signal is 1V8 logic level at the NTM Interface!**

Configures the Synth to output a fixed frequency. Valid range is 10800-12900 MHz, with 1 MHz step size.

Use the “ADF5356 Synth Tuning” sheet of Excel file “KT-000-0202-00 Rev A.1 - Frequency Planning.xlsx” to determine correct register values for any given output frequency.

As part of Test Jig initialisation, need to write all registers in descending order (Register 13 down to Register 0) as per the Excel file, using 10800 MHz as the initial output frequency.

When this Set Synth Output Frequency command is used, only need to write the registers below, and in the order shown (determining the appropriate register values as per the Excel file):

1. Write Register 13.
2. Write Register 10.
3. Write Register 2.
4. Write Register 1.
5. Wait for 160 us (the datasheet example given matches our settings, and the previous two register writes won’t take very long…)
6. Write Register 0.

Refer to the ADF5356 datasheet for more information.

## Set Synth Power Down ("#SYNPD"?)

True or false as parameter… By default the Synth should be in the powered up state following Test Jig initialisation, but this command may be useful for debugging spurious issues by powering down the synth.

Updates bit DB6 in Register 4 as appropriate.

## Write Synth Register ("#SYNRG"?)

Will be useful for debugging Synth issues as well as playing with settings. Write user-input 32-bit data to the Synth.