Software Specification

SC2444 Configuration and Control API

Number: SCT-SW1A61vc

Date: 2022-07-06
## Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Author</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>2022-Jul-07</td>
<td>K. Sumlak</td>
<td>Initial Release</td>
</tr>
</tbody>
</table>
Table of Contents

Revision History ........................................................................................................................................... 2

Table of Contents ...................................................................................................................................... 3

1. Overview ................................................................................................................................................. 8
   1.1 Protocols ........................................................................................................................................... 8
   1.2 Hardware .......................................................................................................................................... 8

2. Hardware Details .................................................................................................................................... 9
   2.1 Connections ................................................................................................................................... 9
      2.1.1 Expansion Port ....................................................................................................................... 9
         2.1.1.1 SPI Bus .............................................................................................................................. 9
         2.1.1.2 Serial ............................................................................................................................... 9
         2.1.1.3 Trigger ............................................................................................................................. 9
         2.1.1.4 Reference ......................................................................................................................... 9
      2.1.2 Control Port ............................................................................................................................. 10
         2.1.2.1 TX IQ Filter Bank ............................................................................................................ 10
         2.1.2.2 Serial ............................................................................................................................... 10
      2.1.3 Antenna Control Port ............................................................................................................. 10
   2.2 Configuration ..................................................................................................................................... 11

3. Console Protocol .................................................................................................................................... 12
   3.1 Command Mapping to Hardware ................................................................................................. 13
   3.2 Common Arguments ..................................................................................................................... 14
   3.3 Slave Responses ......................................................................................................................... 14
   3.4 Command Details ....................................................................................................................... 15
      3.4.1 System ..................................................................................................................................... 15
         3.4.1.1 SYS:ROLE ....................................................................................................................... 15
         3.4.1.2 SYS:ROLE? ..................................................................................................................... 15
         3.4.1.3 SYS:NCHAN ..................................................................................................................... 15
         3.4.1.4 SYS:NCHAN? ................................................................................................................... 16
         3.4.1.5 SYS:ID? ............................................................................................................................ 16
      3.4.2 Converter .................................................................................................................................. 17
         3.4.2.1 CV:MODE ......................................................................................................................... 17
         3.4.2.2 CV:MODE? ....................................................................................................................... 17
         3.4.2.3 CV:LO .............................................................................................................................. 18
         3.4.2.4 CV:LO? ............................................................................................................................ 18
         3.4.2.5 CV:VGAIN ......................................................................................................................... 19
<table>
<thead>
<tr>
<th>Section</th>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4.2.6</td>
<td>CV:VGAIN?</td>
<td>19</td>
</tr>
<tr>
<td>3.4.2.7</td>
<td>CV:CMV</td>
<td>20</td>
</tr>
<tr>
<td>3.4.2.8</td>
<td>CV:CMV?</td>
<td>20</td>
</tr>
<tr>
<td>3.4.2.9</td>
<td>CV:PHASE</td>
<td>21</td>
</tr>
<tr>
<td>3.4.2.10</td>
<td>CV:PHASE?</td>
<td>21</td>
</tr>
<tr>
<td>3.4.2.11</td>
<td>CV:MXOFF</td>
<td>22</td>
</tr>
<tr>
<td>3.4.2.12</td>
<td>CV:MXOFF?</td>
<td>22</td>
</tr>
<tr>
<td>3.4.2.13</td>
<td>CV:IFAMP</td>
<td>23</td>
</tr>
<tr>
<td>3.4.2.14</td>
<td>CV:IFAMP?</td>
<td>23</td>
</tr>
<tr>
<td>3.4.2.15</td>
<td>CV:BBAMPOFFSET</td>
<td>24</td>
</tr>
<tr>
<td>3.4.2.16</td>
<td>CV:BBAMPOFFSET?</td>
<td>24</td>
</tr>
<tr>
<td>3.4.2.17</td>
<td>CV:REGW</td>
<td>25</td>
</tr>
<tr>
<td>3.4.2.18</td>
<td>CV:REGR?</td>
<td>25</td>
</tr>
<tr>
<td>3.4.3</td>
<td>Synthesizer</td>
<td>26</td>
</tr>
<tr>
<td>3.4.3.1</td>
<td>SYN:REF</td>
<td>26</td>
</tr>
<tr>
<td>3.4.3.2</td>
<td>SYN:REF?</td>
<td>26</td>
</tr>
<tr>
<td>3.4.3.3</td>
<td>SYN:SPC</td>
<td>27</td>
</tr>
<tr>
<td>3.4.3.4</td>
<td>SYN:SPC?</td>
<td>27</td>
</tr>
<tr>
<td>3.4.3.5</td>
<td>SYN:MOD2</td>
<td>28</td>
</tr>
<tr>
<td>3.4.3.6</td>
<td>SYN:MOD2?</td>
<td>28</td>
</tr>
<tr>
<td>3.4.3.7</td>
<td>SYN:OUT</td>
<td>29</td>
</tr>
<tr>
<td>3.4.3.8</td>
<td>SYN:OUT?</td>
<td>29</td>
</tr>
<tr>
<td>3.4.3.9</td>
<td>SYN:PFDSET</td>
<td>30</td>
</tr>
<tr>
<td>3.4.3.10</td>
<td>SYN:PFDSET?</td>
<td>30</td>
</tr>
<tr>
<td>3.4.3.11</td>
<td>SYN:PFDREQ?</td>
<td>31</td>
</tr>
<tr>
<td>3.4.3.12</td>
<td>SYN:RFSET</td>
<td>31</td>
</tr>
<tr>
<td>3.4.3.13</td>
<td>SYN:RFSET?</td>
<td>32</td>
</tr>
<tr>
<td>3.4.3.14</td>
<td>SYN:POW</td>
<td>32</td>
</tr>
<tr>
<td>3.4.3.15</td>
<td>SYN:POW?</td>
<td>33</td>
</tr>
<tr>
<td>3.4.3.16</td>
<td>SYN:EN</td>
<td>33</td>
</tr>
<tr>
<td>3.4.3.17</td>
<td>SYN:EN?</td>
<td>34</td>
</tr>
<tr>
<td>3.4.3.18</td>
<td>SYN:APPLY</td>
<td>34</td>
</tr>
<tr>
<td>3.4.3.19</td>
<td>SYN:FREQUPDATE</td>
<td>35</td>
</tr>
<tr>
<td>3.4.3.20</td>
<td>SYN:FETCH</td>
<td>35</td>
</tr>
<tr>
<td>3.4.3.21</td>
<td>SYN:REGW</td>
<td>36</td>
</tr>
<tr>
<td>3.4.3.22</td>
<td>SYN:REGR</td>
<td>36</td>
</tr>
<tr>
<td>3.4.3.23</td>
<td>SYN:BREGW</td>
<td>37</td>
</tr>
</tbody>
</table>
3.4.10 SYN:BREG? ........................................................................................................................................37
3.4.4 Local Oscillator ..................................................................................................................................38
  3.4.4.1 LOCLK:FREQ ................................................................................................................................38
  3.4.4.2 LOCLK:FREQ? ..............................................................................................................................38
  3.4.4.3 LOCLK:SEL ..................................................................................................................................39
  3.4.4.4 LOCLK:SEL? ................................................................................................................................39
  3.4.4.5 LOCLK:REF ..................................................................................................................................40
  3.4.4.6 LOCLK:REF? ................................................................................................................................40
3.4.5 RF Path ..............................................................................................................................................41
  3.4.5.1 RFPATH:PATH ..............................................................................................................................41
  3.4.5.2 RFPATH:PATH? ..........................................................................................................................41
  3.4.5.3 RFPATH:TDD ..............................................................................................................................42
  3.4.5.4 RFPATH:TDD? ..........................................................................................................................42
3.4.6 IF Path ................................................................................................................................................43
  3.4.6.1 IFPATH:FLT ..................................................................................................................................43
  3.4.6.2 IFPATH:FLT? ...............................................................................................................................43
  3.4.6.3 IFPATH:ATTN ..............................................................................................................................44
  3.4.6.4 IFPATH:ATTN? ............................................................................................................................44
3.4.7 Baseband IQ Path ................................................................................................................................45
  3.4.7.1 BPATH:GAIN ..............................................................................................................................45
  3.4.7.2 BPATH:GAIN? ............................................................................................................................45
3.4.8 Baseband IQ Filter ...............................................................................................................................46
  3.4.8.1 BFLT:FLT ....................................................................................................................................46
  3.4.8.2 BFLT:FLT? ..................................................................................................................................46
  3.4.8.3 BFLT:GAIN ..................................................................................................................................47
  3.4.8.4 BFLT:GAIN? ...............................................................................................................................47
  3.4.8.5 BFLT:VGAIN ..............................................................................................................................48
  3.4.8.6 BFLT:VGAIN? ............................................................................................................................48
  3.4.8.7 BFLT:REGW ..............................................................................................................................49
  3.4.8.8 BFLT:REGR? ..............................................................................................................................49
3.4.9 External Baseband IQ Filter ..................................................................................................................50
  3.4.9.1 EBFLT:FLT ..................................................................................................................................50
  3.4.9.2 EBFLT:FLT? ...............................................................................................................................50
  3.4.9.3 EBFLT:REGW ..............................................................................................................................51
  3.4.9.4 EBFLT:REGR? ............................................................................................................................51
3.4.10 Antenna Array ....................................................................................................................................52
  3.4.10.1 ANT:SIG ....................................................................................................................................52
4. **Binary Protocol** ................................................................................................................. 62

4.1 Slave Timing ......................................................................................................................... 62

4.2 Protocol Sequence .................................................................................................................. 63

4.2.1 Slave Responses .................................................................................................................. 63

4.2.2 Write Transaction .............................................................................................................. 64

4.2.3 Read Transaction .............................................................................................................. 64

4.3 Transaction Data .................................................................................................................... 65

4.3.1 Command Byte .................................................................................................................. 65

4.3.2 Response Byte .................................................................................................................. 65

4.3.3 Data Bytes ......................................................................................................................... 65

4.4 Command Details .................................................................................................................. 66

4.4.1 0x00 - Null Operation ...................................................................................................... 66

4.4.2 0x01 - TDD IF Configuration .......................................................................................... 66
4.4.3 0x02 - TDD Baseband Configuration

4.4.4 0x03 - Active Path Select

4.4.5 0x04 - Hardware Control

4.4.5.1 Control ID

4.4.6 0x05 - Hardware Status

4.4.6.1 Status ID

4.4.7 0x10 - Converter Register Access

4.4.8 0x11 - Synthesizer Register Access

4.4.9 0x12 - Baseband IQ Filter Register Access

5. Appendix

5.1 Console Protocol Examples

5.1.1 Setup 1 – IF Mode FDD Loopback

5.1.1.1 Details

5.1.1.2 Sequence

5.1.2 Setup 2 – Baseband IQ TDD Conversion

5.1.2.1 Details

5.1.2.2 Sequence
1. Overview

This document describes the software API and hardware interfaces required to configure and control the SC2444 RadioThorium mmWave Up/Down Converter.

1.1 Protocols

Two protocol types are available to configure and control the SC2444:

- **Console Protocol** – The ASCII Protocol operates over the USB and Control Port UART interfaces. This protocol is intended as a simple user-friendly interface for interactive control of the hardware. See Section 3 for additional details.

- **Binary Protocol** – The Binary Protocol operates over the Expansion Control Port SPI Interface. This protocol is intended as an alternative suitable for interfacing with an external microcontroller or FPGA. Additionally, the SPI interface operates at higher speeds required for time-critical applications. See Section 4 for additional details.

1.2 Hardware

The Hardware interfaces available are detailed in Figure 1 and described in detail below.

![Figure 1 – SC2444 Hardware](image)

- **USB Control** – CDC USB device class that utilizes the Console Protocol (Section 3).
- **Control Port** – This port interfaces with the optional external TX IQ Filter Bank (part number SC4710) or provides a UART Serial Bus supporting the Console Protocol.
- **Expansion Port** – This port enables support for multi-unit configurations by vertically stacking up to four SC2444 units. This port utilizes the high-speed Binary Protocol over an SPI interface. Custom hardware containing an FPGA or microprocessor can access this port to implement application-specific control.
- **Antenna Control** – Provides control to other devices such as an Antenna Array.
2. Hardware Details

2.1 Connections

This section provides details related to the hardware connectors and the pin designations.

2.1.1 Expansion Port

The Expansion Port provides unit-to-unit connectivity when vertically stacking multiple SC2444 units. Additionally, an external FPGA or microprocessor can access this port to implement a custom device as a System Master. In this configuration, up to four SC2444 units are required to be configured as System Slaves. Figure 2 and Table 1 detail the pin functions and names.

![Expansion Port](image)

**Figure 2 - Expansion Port (Top Side Only)**

<table>
<thead>
<tr>
<th>Function</th>
<th>Pin(s)</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPI Bus</td>
<td>2, 3, 12, 14</td>
<td>SCK, CS#, MOSI, MISO</td>
</tr>
<tr>
<td>Serial</td>
<td>8, 9</td>
<td>RS485_P, RS485_N</td>
</tr>
<tr>
<td>Trigger</td>
<td>6, 16</td>
<td>TRIG_LVL, TRIG_TDD</td>
</tr>
<tr>
<td>Reference</td>
<td>18, 19</td>
<td>REF_IN_P, REF_IN_N</td>
</tr>
<tr>
<td>Ground</td>
<td>1, 3, 5, 7, 10, 11, 13, 15, 17, 20</td>
<td>GND</td>
</tr>
</tbody>
</table>

**Table 1 - Expansion Port Connections**

The connector part is [Hirose FX20 Series](https://example.com), FX20-20S-0.5SV10. The compatible mating connector for the Expansion Port is [Hirose FX20 Series](https://example.com), FX20-20P-0.5SV15.

2.1.1.1 SPI Bus

The SPI Bus supports the Binary Protocol as described in Section 4.

2.1.1.2 Serial

The Serial Bus supports the Console Protocol as described in Section 3. The Master device must drive this bus using a differential RS485 transceiver, such as the [LTC2854](https://example.com), and provide a 120 Ω termination.

See Table 2 for the required serial port settings.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baud Rate</td>
<td>115200</td>
</tr>
<tr>
<td>Parity</td>
<td>Even</td>
</tr>
<tr>
<td>Data Size</td>
<td>8 bits</td>
</tr>
<tr>
<td>Stop Bits</td>
<td>1</td>
</tr>
<tr>
<td>Flow Control</td>
<td>None</td>
</tr>
</tbody>
</table>

**Table 2 – Serial Port Settings**

2.1.1.3 Trigger

When the triggers are enabled, the external trigger signals controlled by the Master initiate the configuration changes. The TRIG_LVL signal latches in the IF Attenuation or Baseband Gain previously configured. The TRIG_TDD signal changes the TDD direction between the Upconversion or Downconversion Paths.

2.1.1.4 Reference

The Reference input signal required is a 10 MHz LVDS clock. This reference is distributed to the connected Slave devices and can be used to synchronize the internal synthesizers that generate the local oscillator signal.
2.1.2 **Control Port**

The Control Port interfaces with the optional external TX IQ Filter Bank (part number SC4710) or provides a UART Serial Bus supporting the Console Protocol. Figure 3 and Table 3 detail the pin functions and names.

The connector part is **Samtec TFML-107-01-S-D-RA**. A compatible mating cable is **Samtec SFSDT-07-28-G-08.00-DL-NDX**.

![Figure 3 – Control Port](image)

<table>
<thead>
<tr>
<th>Function</th>
<th>Pin(s)</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>SC4710</td>
<td>7, 9, 12</td>
<td>IO1, IO2, Power</td>
</tr>
<tr>
<td>Reserved</td>
<td>1, 2, 8, 10, 12</td>
<td>Reserved</td>
</tr>
<tr>
<td>Serial</td>
<td>3, 4</td>
<td>TX OUT, RX IN</td>
</tr>
<tr>
<td>Ground</td>
<td>5, 6, 11, 13, 14</td>
<td>GND</td>
</tr>
</tbody>
</table>

**Table 3 – Control Port Connections**

2.1.2.1 **TX IQ Filter Bank**

The SC4710 provides additional filtering for the baseband TX IQ path. See Section 3.4.9 for the Console Protocol commands used to configure the external filter bank.

2.1.2.2 **Serial**

The UART Serial Bus requires TTL 3.3V logic levels. The serial settings required are shown in Table 2.

2.1.3 **Antenna Control Port**

Provides control to an Antenna Array. Figure 4 and Table 4 detail the pin functions and names. The connector part is **Samtec FCS8-10-01-L—S-A-TR**. A compatible mating cable is **Samtec FCF8-10-01-L-08.00-S-1**.

![Figure 4 – Antenna Control Port](image)

<table>
<thead>
<tr>
<th>Function</th>
<th>Pin(s)</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Outputs</td>
<td>1, 2, 3, 4</td>
<td>SPARE, BEAM, TRSW, RESET</td>
</tr>
<tr>
<td>Power</td>
<td>5</td>
<td>POWER_IN</td>
</tr>
<tr>
<td>Ground</td>
<td>6</td>
<td>GND</td>
</tr>
<tr>
<td>SPI Bus</td>
<td>7, 8, 9, 10</td>
<td>CS#, MISO, MOSI, SCK</td>
</tr>
</tbody>
</table>

**Table 4 – Antenna Control Port Connections**

The Antenna Array Outputs and SPI bus are controlled using the Console Protocol commands described in Section 3.4.10. The Power Input is required to translate the IO voltage level of the Outputs and the SPI Bus signals to the voltage required by the Antenna Array assembly.
2.2 Configuration

The Configuration DIP switch determines how the SC2444 will function within a system with one or more units. The Switch positions are described from left to right in Table 5.

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Master/Slave</td>
<td>Selects the Master or Slave system role. When a device is configured as a Master, it can communicate to other Slave devices. Only one Master device can be connected to a system.</td>
</tr>
<tr>
<td></td>
<td>• Master Role = Down</td>
</tr>
<tr>
<td></td>
<td>• Slave Role = Up</td>
</tr>
<tr>
<td>Channel ID0/ID1</td>
<td>Selects the Channel ID required for the Console Protocol (Section 3). All devices connected must have a unique Channel ID selected.</td>
</tr>
<tr>
<td></td>
<td>• Channel ID 0 = ID1 (Down), ID0 (Down)</td>
</tr>
<tr>
<td></td>
<td>• Channel ID 1 = ID1 (Down), ID0 (Up)</td>
</tr>
<tr>
<td></td>
<td>• Channel ID 2 = ID1 (Up), ID0 (Down)</td>
</tr>
<tr>
<td></td>
<td>• Channel ID 3 = ID1 (Up), ID0 (Up)</td>
</tr>
<tr>
<td>BOOT0/1</td>
<td>Reserved. For normal operation, the switch positions must be as follows:</td>
</tr>
<tr>
<td></td>
<td>• BOOT0 = Up</td>
</tr>
<tr>
<td></td>
<td>• BOOT1 = Down</td>
</tr>
</tbody>
</table>

*Table 5 - Configuration DIP Switch*
3. Console Protocol

The console protocol is derived from but does not strictly adhere to SCPI/IEEE 488.2.

As per SCPI:

- ASCII based
- Command Syntax structure obeys general SCPI rules in terms of structure and organization.
- Query commands are terminated with a “?” Character

SCPI deviations:

- Input commands are terminated by a carriage return 0Dh only.
- For the console mode behavior, only a single command is accepted on a single line. There is no support for multiple commands separate by a semicolon on a single line.
- All commands will return their output on completion.
- For non-query commands (i.e., not terminated by a “?”), a status is returned ("OK" for success, “ERR” for error – possibly preceded with error information).
- For query commands, the output is immediately returned and then followed by a status ("OK" for success, “ERR” for error).
- A prompt is returned as the “>” character. New commands can be input once the prompt is received.
- Numerical arguments can be either in decimal or hexadecimal format. This will vary from command to command.
3.1 Command Mapping to Hardware

The following diagram details the high-level association between Console Protocol and the Hardware.

**Transmit Path**

**Receive Path**

**Local Oscillators**

*Figure 5 – Control Protocol Hardware Mapping*
3.2 Common Arguments

Many of the commands will require 2 common arguments.

1. **Channel ID** – This is the ID used to identify the specific UDCM and is assigned using the Configuration DIP Switch (Section 2.2). The valid range for this argument is 0 – 3.
2. **Path** – This identifies which of the Paths the command is directed to. The valid values for this argument are:
   - TX – transmit or upconverter
   - RX – receive or downconverter

3.3 Slave Responses

Commands that include a *Channel ID* identify the UDCM to which that command is to be forwarded. If the UDCM is not the local device, the actions taken will depend on the role of the UDCM as requested in the `SYS:ROLE` command (see section 3.4.1.1).

- If the UDCM is acting as a *SLAVE* device, the command will be ignored.
- If the UDCM is acting as a *MASTER* device, the command will be forwarded to the target device.

If the target device is present, the response will be returned with a prefix to identify its source (commands processed by the local device do not have the prefix). The prefix used has the following format \[ Channel ID \]. An example of a slave response is shown below.

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; IFPATH:FLT? 2 TX</td>
</tr>
<tr>
<td>[ 2 ] BYP</td>
</tr>
<tr>
<td>[ 2 ] OK</td>
</tr>
<tr>
<td>&gt; IFPATH:FLT? 3 TX</td>
</tr>
<tr>
<td>TIMEOUT</td>
</tr>
<tr>
<td>ERR</td>
</tr>
</tbody>
</table>

In the above example, the commands are issued by the UDCM Master (Channel-ID = 0). The Master attempts to forward the commands to the slave devices 2 and 3, however, only slave 2 responds (3 is unavailable or not connected).

**NOTES:**

1. The command status responses (*OK* or *ERR*) are returned as part of the slave response. The normal prompt “>” follows the slave response.
2. In the event, no response is received from the slave, an error will be reported by the local master. The normal prompt “>” follows the error response.
### 3.4 Command Details

#### 3.4.1 System

These commands set the system-level behavior of the UDCM.

##### 3.4.1.1 SYS:ROLE

**Description**

Sets the UDCM role as either a master or slave in the system structure. Only one master device can be present for a given configuration and is responsible for sending commands to the other slave UDCMs. A slave device is only capable of processing commands specific to its local hardware.

**Syntax**

SYS:ROLE <channel ID> <role>

**Arguments**

- channel ID: <0 - 3>
- role: <MASTER | SLAVE>
  - MASTER - operates in the master role
  - SLAVE - operations in the slave role

**Example**

> SYS:ROLE 0 MASTER

OK

##### 3.4.1.2 SYS:ROLE?

**Description**

Gets the UDCM role.

**Syntax**

SYS:ROLE? <channel ID>

**Arguments**

- channel ID: <0 - 3>

**Response**

- role: <MASTER | SLAVE>

**Example**

> SYS:ROLE? 0

MASTER

OK

##### 3.4.1.3 SYS:NCHAN

**Description**

Sets the number of channels in the system. This command is significant for a UDCM configured for the master role.

**Syntax**

SYS:NCHAN <channel ID> <number of channels>

**Arguments**

- channel ID: <0 - 3>
- number of channels: <1 - 4>

**Example**

> SYS:NCHAN 0 4

OK
### 3.4.1.4 **SYS:NCHAN?**

**Description**

Gets the number of channels in the system. This command is only significant for a UDCM configured for the master role. For a UDCM configured as a slave, the value of 1 is always returned.

**Syntax**

```plaintext
SYS:NCHAN? <channel ID>
```

**Arguments**

- channel ID: <0 - 3>

**Response**

- number of channels: <1 -4>

**Example**

```plaintext
> SYS:NCHAN? 0
4
OK
```

### 3.4.1.5 **SYS:ID?**

**Description**

Gets the assigned channel ID of the local system.

**Syntax**

```plaintext
SYS:ID?
```

**Arguments**

None

**Response**

- channel ID: <0 - 3>

**Example**

```plaintext
> SYS:ID?
0
OK
```
3.4.2 **Converter**

These commands are the instrument-specific commands to control the upconverter (ADMV1013) and downconverter (ADMV1014) IC devices.

The supported commands give access to the most typical features of the 2 devices. However, some advanced features do not have direct support (including envelope detector control and some additional power controls). If more precise control is required, the CV:REGW and CV:REGR commands in sections 3.4.2.13 and 3.4.2.18 to access any register in the devices.

### 3.4.2.1 CV:MODE

**Description**

Sets the up/down converter to operate in baseband IQ or IF mode.

**Syntax**

```
CV:MODE <channel ID> <path> <mode>
```

**Arguments**

- **channel ID**: `<0 - 3>`
- **path**: `<TX | RX>`
  - TX – transmit/upconverter
  - RX – receive/downconverter
- **mode**: `<IQ | IF>`
  - IQ – converter operates in baseband IQ mode
  - IF – converter operations in IF mode

**Example**

```
> CV:MODE 0 TX IQ
OK
> CV:MODE 0 RX IF
OK
```

### 3.4.2.2 CV:MODE?

**Description**

Gets the up/down converter operating mode.

**Syntax**

```
CV:MODE? <channel ID> <path>
```

**Arguments**

- **channel ID**: `<0 - 3>`
- **path**: `<TX | RX>`
  - TX – transmit/upconverter
  - RX – receive/downconverter

**Response**

- **mode**: `<IQ | IF>`

**Example**

```
> CV:MODE? 0 TX IQ
IQ
OK
> CV:MODE? 0 RX IF
IF
OK
```
### 3.4.2.3 CV:LO

**Description**
Sets the up/down converter LO input type and frequency filter bandwidth.

**Syntax**
```plaintext
CV:LO <channel ID> <path> <input> <filter bandwidth>
```

**Arguments**
- **channel ID:** `<0 - 3>`
- **path:** `<TX | RX>`
  - TX – transmit/upconverter
  - RX – receive/downconverter
- **input:** `<POS | NEG | DIFF>`
  - POS - converter LO input is in Single-Ended Mode, with Negative Side Disabled
  - (Not Supported) NEG - converter LO input is in Single-Ended Mode, with Positive Side Disabled
  - (Not Supported) DIFF - converter LO input is in Differential Mode
- **filter bandwidth:** `<0 – 3>`
  - 0: LO Frequency Bandwidth: 8.62 GHz to 10.25 GHz
  - 1: LO Frequency Bandwidth: 6.6 GHz to 9.2 GHz
  - 2: LO Frequency Bandwidth: 5.4 GHz to 8 GHz
  - 3: LO Frequency Bandwidth: 5.4 GHz to 7 GHz

**Example**
```plaintext
> CV:LO 0 TX POS 0
OK
> CV:LO 0 RX POS 2
OK
```

### 3.4.2.4 CV:LO?

**Description**
Gets the up/down converter LO input type and frequency filter bandwidth.

**Syntax**
```plaintext
CV:LO? <channel ID> <path>
```

**Arguments**
- **channel ID:** `<0 - 3>`
- **path:** `<TX | RX>`
  - TX – transmit/upconverter
  - RX – receive/downconverter

**Response**
- **input:** `<POS | NEG | DIFF>`
  - POS - converter LO input is in Single-Ended Mode, with Negative Side Disabled
  - (Not Supported) NEG - converter LO input is in Single-Ended Mode, with Positive Side Disabled
  - (Not Supported) DIFF - converter LO input is in Differential Mode
- **filter bandwidth:** `<0 – 3>`
  - 0: LO Frequency Bandwidth: 8.62 GHz to 10.25 GHz
  - 1: LO Frequency Bandwidth: 6.6 GHz to 9.2 GHz
  - 2: LO Frequency Bandwidth: 5.4 GHz to 8 GHz
  - 3: LO Frequency Bandwidth: 5.4 GHz to 7 GHz

**Example**
```plaintext
> CV:LO? 0 RX POS
POS 0
OK
```
### 3.4.2.5 **CV:VGAIN**

**Description**
Sets the control gain applied for the specified up/down converter using an external DAC to supply the analog control signal to the ADMV1013 or ADMV1014. See the respective datasheets for the voltage to gain relationships.

- For TX/upconverter (ADMV1013) Max Gain = 1.8 V, Min Gain = 0 V
- For RX/downconverter (ADMV1014) Max Gain = 0 V, Min Gain = 1.8 V

**Syntax**

```
CV:VGAIN <channel ID> <path> <gain control>
```

**Arguments**

- **channel ID:** <0 - 3>
- **path:** <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter
- **gain control:** <0 – 1.8 V>

**Example**

```
> CV:VGAIN 0 TX 1.2
OK
> CV:VGAIN 0 RX 0.7
OK
```

### 3.4.2.6 **CV:VGAIN?**

**Description**

Gets the gain control voltage applied to the specified up/down converter.

**Syntax**

```
CV:VGAIN? <channel ID> <path>
```

**Arguments**

- **channel ID:** <0 - 3>
- **path:** <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter

**Response**

- **gain control:** <0 – 1.8 V>

**Example**

```
> CV:VGAIN? 0 TX 1.2
OK
> CV:VGAIN? 0 RX 0.7
OK
```
### 3.4.2.7 CV:CMV

**Description**
Sets the common-mode voltage of the mixer when in IQ mode.

**Syntax**
CV:CMV <channel ID> <path> <voltage>

**Arguments**
- channel ID: <0 - 3>
- path: <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter
- voltage: <0 – 2600 mV>

**Example**

```
> CV:CMV 0 TX 1500
OK
> CV:CM 0 RX 1050
OK
```

### 3.4.2.8 CV:CMV?

**Description**
 Gets the common-mode voltage of the mixer when in IQ mode.

**Syntax**
CV:CMV <channel ID> <path>

**Arguments**
- channel ID: <0 - 3>
- path: <TX>
  - TX – transmit/upconverter
  - RX – receive/downconverter

**Response**
- voltage: <0 – 2600 mV>

**Example**

```
> CV:CMV? 0 TX
1500
OK
> CV:CMV? 0 RX
1050
OK
```
### 3.4.2.9 CV:PHASE

**Description**
Sets phase adjustments to calibrate the mixer image rejection for the up/down converter.

**Syntax**
CV:PHASE <channel ID> <path> <I phase> <Q phase>

**Arguments**
- **channel ID**: <0 - 3>
- **path**: <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter
- **I phase**: Register value LOAMP_PH_ADJ_I_FINE of the ADMV101[3/4] <0 - 127>
- **Q phase**: Register value LOAMP_PH_ADJ_Q_FINE of the ADMV101[3/4] <0 - 127>

**Example**
```plaintext
> CV:PHASE 0 TX 51 65
OK
> CV:PHASE 0 RX 72 83
OK
```

### 3.4.2.10 CV:PHASE?

**Description**
Gets the phase adjustments to calibrate the mixer image rejection for the up/down converter.

**Syntax**
CV:PHASE? <channel ID> <path>

**Arguments**
- **channel ID**: <0 - 3>
- **path**: <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter

**Response**
- **I phase**: Register value LOAMP_PH_ADJ_I_FINE of the ADMV101[3/4] <0 - 127>
- **Q phase**: Register value LOAMP_PH_ADJ_Q_FINE of the ADMV101[3/4] <0 - 127>

**Example**
```plaintext
> CV:PHASE? 0 TX 51 65
OK
> CV:PHASE? 0 RX 72 83
OK
```
### 3.4.2.11 CV:MXOFF

**Description**

Sets mixer offset adjustment to reduce LO feedthrough when in IF mode.

**Syntax**

```
CV:MXOFF <channel ID> <path> <I pos> <I neg> <Q pos> <Q neg>
```

**Arguments**

- **channel ID**: `<0 - 3>`
- **path**: `<TX>`
  - TX – transmit/upconverter
- **I pos**: Register value `MXER_OFF_ADJ_I_P` of the ADMV1013 `<0 - 127>`
- **I neg**: Register value `MXER_OFF_ADJ_I_N` of the ADMV1013 `<0 - 127>`
- **Q pos**: Register value `MXER_OFF_ADJ_Q_P` of the ADMV1013 `<0 - 127>`
- **Q neg**: Register value `MXER_OFF_ADJ_Q_N` of the ADMV1013 `<0 - 127>`

**Example**

```
> CV:MXOFF 0 TX 51 65 21 31
OK
> CV:MXOFF 0 RX 72 83 75 72
OK
```

### 3.4.2.12 CV:MXOFF?

**Description**

Gets sets mixer offset adjustment to reduce LO feedthrough when in IF mode.

**Syntax**

```
CV:MXOFF? <channel ID> <path>
```

**Arguments**

- **channel ID**: `<0 - 3>`
- **path**: `<TX>`
  - TX – transmit/upconverter

**Response**

- **I pos**: Register value `MXER_OFF_ADJ_I_P` of the ADMV1013 `<0 - 127>`
- **I neg**: Register value `MXER_OFF_ADJ_I_N` of the ADMV1013 `<0 - 127>`
- **Q pos**: Register value `MXER_OFF_ADJ_Q_P` of the ADMV1013 `<0 - 127>`
- **Q neg**: Register value `MXER_OFF_ADJ_Q_N` of the ADMV1013 `<0 - 127>`

**Example**

```
> CV:MXOFF? 0 TX 10 20 30 40
OK
> CV:MXOFF? 0 TX 100 95 85 75
OK
```
3.4.2.13 **CV:IFAMP**

**Description**
Sets the IF amplifier gain adjustments.

**Syntax**
CV:IFAMP <channel ID> <path> <I Coarse> <I Fine> <Q Coarse> <Q Fine>

**Arguments**
- channel ID: <0 - 3>
- path: <RX>
  - RX – receive/downconverter
- I coarse: Register value `IF_AMP_COARSE_GAIN_I` of the ADMV1014 <0 - 15>
- I fine: Register value `IF_AMP_FINE_GAIN_I` of the ADMV1014 <0 - 15>
- Q coarse: Register value `IF_AMP_COARSE_GAIN_Q` of the ADMV1014 <0 - 15>
- Q fine: Register value `IF_AMP_FINE_GAIN_Q` of the ADMV1014 <0 - 15>

**Example**

```plaintext
> CV:IFAMP 0 RX 5 7 10 12
OK
> CV:IFAMP 0 RX 11 13 2 1
OK
```

3.4.2.14 **CV:IFAMP?**

**Description**
Gets the IF amplifier gain adjustments.

**Syntax**
CV:IFAMP? <channel ID> <path>

**Arguments**
- channel ID: <0 - 3>
- path: <RX>
  - RX – receive/downconverter

**Response**
- I coarse: Register value `IF_AMP_COARSE_GAIN_I` of the ADMV1014 <0 - 15>
- I fine: Register value `IF_AMP_FINE_GAIN_I` of the ADMV1014 <0 - 15>
- Q coarse: Register value `IF_AMP_COARSE_GAIN_Q` of the ADMV1014 <0 - 15>
- Q fine: Register value `IF_AMP_FINE_GAIN_Q` of the ADMV1014 <0 - 15>

**Example**

```plaintext
> CV:IFAMP? 0 RX 5 7 10 12
OK
> CV:IFAMP? 0 RX 11 13 2 1
OK
```
### 3.4.2.15 CV:BBAMPOFFSET

<table>
<thead>
<tr>
<th>Description</th>
<th>Sets the baseband amplifier gain offset.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>CV:BBAMPOFFSET &lt;channel ID&gt; &lt;path&gt; &lt;I offset&gt; &lt;Q offset&gt;</td>
</tr>
</tbody>
</table>
| Arguments   | • channel ID: <0 - 3>  
• path: <RX>  
  o RX – receive/downconverter  
• I offset: Register value BB_AMP_OFFSET_I of the ADMV1014 -15 -15  
• Q offset: Register value IF_AMP_FINE_GAIN_I of the ADMV1014 -15 -15 |
| Example     | > CV:BBAMPOFFSET 0 RX -15 -15  
OK |

### 3.4.2.16 CV:BBAMPOFFSET?

<table>
<thead>
<tr>
<th>Description</th>
<th>Gets the baseband amplifier gain offset.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>CV:BBAMPOFFSET? &lt;channel ID&gt; &lt;path&gt;</td>
</tr>
</tbody>
</table>
| Arguments   | • channel ID: <0 - 3>  
• path: <RX>  
  o RX – receive/downconverter  
| Response    | • I offset: Register value BB_AMP_OFFSET_I of the ADMV1014 -15 -15  
• Q offset: Register value IF_AMP_FINE_GAIN_I of the ADMV1014 -15 -15 |
| Example     | > CV:BBAMPOFFSET? 0 RX -15 -15  
OK |
### 3.4.2.17 CV:REGW

**Description**

Directly writes the registers of the ADMV1013 (upconverter) or ADMV1014 (downconverter).

**Syntax**

CV:REGW <channel ID> <path> <register address> <register value>

**Arguments**

- channel ID: <0 - 3>
- path: <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter
- register address: <0x00 – 0x3F>
- register value: <0x0000 – 0xFFFF>

**Example**

```plaintext
> CV:REGW 0 TX 0x3 0x1d7
OK
> CV:REGW 0 RX 0x4 0x5700
OK
```

### 3.4.2.18 CV:REGR?

**Description**

Directly reads the registers of the ADMV1013 (upconverter) or ADMV1014 (downconverter).

**Syntax**

CV:REGR? <channel ID> <path> <register address>

**Arguments**

- channel ID: <0 - 3>
- path: <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter
- register address: <0x00 – 0x3F>

**Response**

- register value: <0x0000 – 0xFFFF>

**Example**

```plaintext
> CV:REGR? 0 TX 0x3 0x1d7
OK
> CV:REGR? 0 RX 0x4 0x5700
OK
```
3.4.3 Synthesizer

These commands control the LO synthesizer (ADF4372) for the upconverter path (TX) and downconverter path (RX). Some commands provide direct access to device-specific registers. Other commands are higher level and will manipulate the device-specific registers as necessary.

Due to the complexity and range of options in setting the ADF4372, the higher-level commands may not set the device-specific registers in a manner that is satisfactory for the application. The ADF4372 may provide many possible internal configurations to synthesize a specific output frequency from an input reference clock. The high-level commands will choose only one possible configuration.

If more precise control is required, the device-specific commands may be used to set some register values. If a specific register is not defined in this set of commands, the ”SYN:REGW” and ”SYN:REGR?” commands in sections 3.4.3.21 and 3.4.3.22 to access any register in the devices.

3.4.3.1 SYN:REF

<table>
<thead>
<tr>
<th>Description</th>
<th>Sets the input reference frequency for the LO synthesizer.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Note:</td>
<td>the hardware is optimized for 10 MHz.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Syntax</th>
<th>SYN:REF &lt;channel ID&gt; &lt;reference frequency&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arguments</td>
<td>channel ID: &lt;0 - 3&gt;</td>
</tr>
<tr>
<td></td>
<td>reference frequency: &lt;10000 – 600000 kHz&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
<th>&gt; SYN:REF 0 10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td></td>
</tr>
</tbody>
</table>

3.4.3.2 SYN:REF?

| Description | Gets the input reference frequency of the LO synthesizers. |

<table>
<thead>
<tr>
<th>Syntax</th>
<th>SYN:REF? &lt;channel ID&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arguments</td>
<td>channel ID: &lt;0 - 3&gt;</td>
</tr>
</tbody>
</table>

| Response    | reference frequency: <10000 – 600000 kHz>               |

<table>
<thead>
<tr>
<th>Example</th>
<th>&gt; SYN:REF? 0 125000</th>
</tr>
</thead>
<tbody>
<tr>
<td>OK</td>
<td></td>
</tr>
</tbody>
</table>
3.4.3.3  SYN:SPC

Description
Sets the channel spacing for the LO synthesizer assigned to the up/down converter. The combination of the channel spacing and Phase Detector frequency (as set by the SYN:PFDSET command) will determine the MOD2 parameter. The formula for this is defined as follows:

\[ MOD2 = \frac{F_{PFD}}{GCD(F_{PFD}, F_{CHSP})} \]

Where:

\( GCD(F_{PFD}, F_{CHSP}) \) is the greatest common divisor of the PFD frequency and the channel spacing frequency. \( F_{CHSP} \) is the desired channel spacing frequency. The MOD2 parameter affects the precise frequencies possible in fractional mode. The MOD2 parameter can be set directly using the SYN:MOD2 command and in the SYN:RFSET commands.

Syntax
SYN:SPC <channel ID> <path> <channel spacing>

Arguments
- channel ID: <0 - 3>
- path: <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter
- channel spacing: <1000 – 1000000 Hz>

Example
> SYN:SPC 0 TX 250000
   OK
> SYN:SPC 0 RX 60000
   OK

3.4.3.4  SYN:SPC?

Description
Gets the channel spacing for the LO synthesizer.

Syntax
SYN:SPC? <channel ID> <path>

Arguments
- channel ID: <0 - 3>
- path: <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter

Response
- channel spacing: <1000 – 1000000 Hz>

Example
> SYN:SPC? 0 TX 250000
   OK
> SYN:SPC? 0 RX 60000
   OK
### 3.4.3.5 SYN:MOD2

**Description**
Sets the MOD2 parameter for the LO synthesizer. The MOD2 parameter affects the precise frequencies possible in fractional mode.

**Syntax**

```
SYN:MOD2 <channel ID> <path> <MOD2>
```

**Arguments**
- channel ID: <0 - 3>
- path: <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter
- MOD2: <2 – 16383>

**Example**

```
> SYN:MOD2 0 TX 5000
OK
> SYN:MOD2 0 RX 600
OK
```

### 3.4.3.6 SYN:MOD2?

**Description**

Gets the MOD2 parameter for the LO synthesizer.

**Syntax**

```
SYN:MOD2? <channel ID> <path>
```

**Arguments**
- channel ID: <0 - 3>
- path: <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter

**Response**
- MOD2 value: <2 - 16383>

**Example**

```
> SYN:MOD2? 0 TX 5000
OK
> SYN:MOD2? 0 RX 600
OK
```
### 3.4.3.7 SYN:OUT

**Description**
SETS THE OUTPUT FREQUENCY FOR THE LO SYNTHESIZER. THE CALCULATIONS USED TO GENERATE THE OUTPUT FREQUENCY ARE BASED ON THE REFERENCE FREQUENCY AND CHANNEL SPACING ASSIGNED WITH THE SYN:REF, AND SYN:SPC OR SYN:MOD2 COMMANDS.

This command automatically alters the registers set by the SYN:RFSET commands.

**Syntax**
SYN:OUT <channel ID> <path> <output frequency>

**Arguments**
- channel ID: <0 - 3>
- path: <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter
- output frequency: <62500 – 8000000 kHz>

**Example**
> SYN:OUT 0 TX 2500000
OK
> SYN:OUT 0 RX 7500000
OK

### 3.4.3.8 SYN:OUT?

**Description**
GETS THE OUTPUT FREQUENCY OF THE LO SYNTHESIZER.

**Syntax**
SYN:OUT? <channel ID> <path>

**Arguments**
- channel ID: <0 - 3>
- path: <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter

**Response**
- output frequency: <62500 – 8000000 kHz>

**Example**
> SYN:OUT? 0 TX 2500000
OK
> SYN:OUT? 0 RX 7500000
OK
### 3.4.3.9  SYN:PFDSET

**Description**

Sets the registers that define the ADF4372 Phase Frequency Detector operating frequency. Note: the hardware is optimized for a 20 MHz PFD frequency which is typically achieved using a 10 MHz reference signal with the D bit set. The formula for this frequency is:

\[ F_{PFD} = F_{REF} \times \left( \frac{1 + D}{R \times (1 + T)} \right) \]

Where:
- \( F_{PFD} \) is the PFD frequency
- \( F_{REF} \) is the reference frequency input.
- D is the reference doubler bit.
- R is the reference division factor.
- T is the reference divide by 2 bit (0 or 1).

**Syntax**

SYN:PFDSET <channel ID> <path> <D> <R> <T>

**Arguments**

- channel ID: <0 - 3>
- path: <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter
- D: reference doubler bit <0 | 1>
- R: reference division factor <0 - 31>, where 0 = divide by 32
- T: is the reference divide by 2 bit <0 | 1>

**Example**

> SYN:PFDSET 0 TX 1 20 1
> OK

> SYN:PFDSET 0 RX 0 16 0
> OK

### 3.4.3.10  SYN:PFDSET?

**Description**

Gets the registers that define the Phase Frequency Detector operating frequency.

**Syntax**

SYN:PFDSET? <channel ID> <path>

**Arguments**

- channel ID: <0 - 3>
- path: <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter

**Response**

- D: reference doubler bit <0 | 1>
- R: reference division factor <0 - 31>, where 0 = divide by 32
- T: is the reference divide by 2 bit <0 | 1>

**Example**

> SYN:PFDSET? 0 TX 1 20 1
> OK

> SYN:PFDSET? 0 RX 0 16 0
> OK
3.4.3.11  **SYN:PFDFREQ?**

**Description**

Gets the Phase Frequency Detector operating frequency for the current settings.

**Syntax**

SYN:PFDFREQ? <channel ID> <path>

**Arguments**

- channel ID: <0 - 3>
- path: <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter

**Response**

- PFD frequency (in kHz)

**Example**

> SYN:PFDFREQ? 0 TX
> 100000
> OK
> SYN:PFDFREQ? 0 RX
> 125000
> OK

3.4.3.12  **SYN:RFSET**

**Description**

Sets the registers that define ADF4372 output frequency.

The formula for this frequency is:

\[ F_{OUT} = \left( \text{INT} + \frac{\text{FRAC1} + \text{FRAC2}}{\text{MOD2}} \right) \times \frac{F_{PFD}}{\text{RF Divider}} \]

Where:

- \( F_{PFD} \) is the PFD frequency
- \( F_{\text{REF}} \) is the reference frequency input.
- D is the reference doubler bit.
- R is the reference division factor.
- T is the reference divide by 2 bit (0 or 1).

**Syntax**

SYN:RFSET <channel ID> <path> <INT> <FRAC1> <FRAC2> <MOD2> <RF Divider>

**Arguments**

- channel ID: <0 - 3>
- path: <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter
- INT: integer division factor <20 – 65535>
- FRAC1: the numerator of the primary modulus <0 - 33554431>
- FRAC2: the numerator of the auxiliary modulus <0 – 16383>
- MOD2: auxiliary fractional modulus <2 - 16383>
- RF Divider: feedback divider <1 | 2 | 4 | 8 | 16 | 32 | 64>

**Example**

> SYN:RFSET 0 TX 260 26039637 512 1536 2
> OK
> SYN:RFSET 0 RX 260 13981013 1024 3072 2
> OK
### 3.4.3.13 SYN:RFSET?

**Description**

Gets registers that define ADF4372 output frequency.

**Syntax**

```
SYN:RFSET? <channel ID> <path>
```

**Arguments**

- channel ID: <0 - 3>
- path: <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter

**Response**

- INT: integer division factor <20 – 65535>
- FRAC1: the numerator of the primary modulus <0 - 33554431>
- FRAC2: the numerator of the auxiliary modulus <0 – 16383>
- MOD2: auxiliary fractional modulus <2 - 16383>
- RF Divider: feedback divider <1 | 2 | 4 | 8 | 16 | 32 | 64>

**Example**

```plaintext
> SYN:RFSET? 0 TX
260 26039637 512 1536 2
OK
> SYN:RFSET? 0 RX
260 13981013 1024 3072 2
OK
```

### 3.4.3.14 SYN:POW

**Description**

Sets the output power level for the LO synthesizer.

**Syntax**

```
SYN:POW <channel ID> <path> <power level>
```

**Arguments**

- channel ID: <0 - 3>
- path: <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter
- power level: <-4 | -1> dBm (+2 and +5 dBm settings are not supported)

**Example**

```plaintext
> SYN:POW 0 TX -1
OK
> SYN:POW 0 RX -1
OK
```
### 3.4.3.15 SYN:POW?

**Description**

Gets the output power level for the LO synthesizer.

**Syntax**

```
SYN:POW? <channel ID> <path>
```

**Arguments**

- **channel ID**: <0 - 3>
- **path**: <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter

**Response**

- power level: <-4 | -1> dBm

**Example**

```
> SYN:POW? 0 TX
-4
OK
> SYN:POW? 0 RX
-4
OK
```

### 3.4.3.16 SYN:EN

**Description**

Sets the output enable for the LO synthesizer.

**Syntax**

```
SYN:EN <channel ID> <path> <RF8 enable> <RF16 enable>
```

**Arguments**

- **channel ID**: <0 - 3>
- **path**: <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter
- **RF8 enable**: Primary RF output path enable <ON | OFF>
- **RF16 enable**: Doubled RF output path enable <ON | OFF>

**Example**

```
> SYN:EN 0 TX ON OFF
OK
> SYN:EN 0 RX OFF ON
OK
```
### 3.4.3.17 SYN:EN?

<table>
<thead>
<tr>
<th>Description</th>
<th>Gets the output enable for the LO synthesizer.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>SYN:EN? &lt;channel ID&gt; &lt;path&gt;</td>
</tr>
<tr>
<td>Arguments</td>
<td>• channel ID: &lt;0 - 3&gt;</td>
</tr>
<tr>
<td></td>
<td>• path: &lt;TX</td>
</tr>
<tr>
<td></td>
<td>o TX – transmit/upconverter</td>
</tr>
<tr>
<td></td>
<td>o RX – receive/downconverter</td>
</tr>
<tr>
<td>Response</td>
<td>• RF8 enable: Primary RF output path enable &lt;ON</td>
</tr>
<tr>
<td></td>
<td>• RF16 enable: Doubled RF output path enable &lt;ON</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; SYN:EN? 0 TX ON OFF OK</td>
</tr>
<tr>
<td></td>
<td>&gt; SYN:EN? 0 RX OFF ON OK</td>
</tr>
</tbody>
</table>

### 3.4.3.18 SYN:APPLY

<table>
<thead>
<tr>
<th>Description</th>
<th>Writes the parameters to the LO synthesizer by performing the ADF4372 Initialization Sequence.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>SYN:APPLY &lt;channel ID&gt; &lt;path&gt;</td>
</tr>
<tr>
<td>Arguments</td>
<td>• channel ID: &lt;0 - 3&gt;</td>
</tr>
<tr>
<td></td>
<td>• path: &lt;TX</td>
</tr>
<tr>
<td></td>
<td>o TX – transmit/upconverter</td>
</tr>
<tr>
<td></td>
<td>o RX – receive/downconverter</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; SYN:APPLY 0 TX OK</td>
</tr>
<tr>
<td></td>
<td>&gt; SYN:APPLY 0 RX OK</td>
</tr>
</tbody>
</table>
### 3.4.3.19 SYN:FREQUPDATE

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Writes a subset of parameters to the LO synthesizer by performing the <em>ADF4372 Frequency Update Sequence</em>.</td>
</tr>
<tr>
<td>Syntax</td>
</tr>
<tr>
<td>SYN:FREQUPDATE &lt;channel ID&gt; &lt;path&gt;</td>
</tr>
<tr>
<td>Arguments</td>
</tr>
<tr>
<td>• channel ID: &lt;0 - 3&gt;</td>
</tr>
<tr>
<td>• path: &lt;TX</td>
</tr>
<tr>
<td>o TX – transmit/upconverter</td>
</tr>
<tr>
<td>o RX – receive/downconverter</td>
</tr>
<tr>
<td>Example</td>
</tr>
<tr>
<td>&gt; SYN: FREQUPDATE 0 TX</td>
</tr>
<tr>
<td>OK</td>
</tr>
<tr>
<td>&gt; SYN: FREQUPDATE 0 RX</td>
</tr>
<tr>
<td>OK</td>
</tr>
</tbody>
</table>

### 3.4.3.20 SYN:FETCH

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reads all parameters from the LO synthesizer, updating the local register values.</td>
</tr>
<tr>
<td>Syntax</td>
</tr>
<tr>
<td>SYN:FETCH &lt;channel ID&gt; &lt;path&gt;</td>
</tr>
<tr>
<td>Arguments</td>
</tr>
<tr>
<td>• channel ID: &lt;0 - 3&gt;</td>
</tr>
<tr>
<td>• path: &lt;TX</td>
</tr>
<tr>
<td>o TX – transmit/upconverter</td>
</tr>
<tr>
<td>o RX – receive/downconverter</td>
</tr>
<tr>
<td>Example</td>
</tr>
<tr>
<td>&gt; SYN: FETCH 0 TX</td>
</tr>
<tr>
<td>OK</td>
</tr>
<tr>
<td>&gt; SYN: FETCH 0 RX</td>
</tr>
<tr>
<td>OK</td>
</tr>
</tbody>
</table>
### 3.4.3.21  SYN:REGW

**Description**

Directly writes the registers of the ADF4372 LO synthesizer.

**Syntax**

```
SYN:REGW <channel ID> <path> <register address> <register value 0> [... <register value n>]
```

**Arguments**

- **channel ID**: <0 - 3>
- **path**: <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter
- **register address**: <0x00 – 0x7FFF>
- **register value**: <0x00 – 0xFF>
  - Up to 8 total registers can be written, registers are addressed in ascending order.

**Example**

```plaintext
> SYN:REGW 0 TX 0x24 0x80 0x55
OK
> SYN:REGW 0 RX 0x26 0x32 0x33 0x22
OK
```

### 3.4.3.22  SYN:REGR?

**Description**

Directly reads the registers of the ADF4372 LO synthesizer.

**Syntax**

```
SYN:REGR? <channel ID> <path> <register address> <read count>
```

**Arguments**

- **channel ID**: <0 - 3>
- **path**: <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter
- **register address**: <0x00 – 0x7FFF>
- **read count**: <1 – 8> registers are addressed in ascending order.

**Response**

- **register value(s)**: <0x00 – 0xFF>

**Example**

```plaintext
> SYN:REGR? 0 TX 0x24 2
0x80 0x55
OK
> SYN:REGR? 0 RX 0x26 3
0x32 0x33 0x22
OK
```
### 3.4.3.23 SYN:BREGW

**Description**

Writes the buffered register values for the LO synthesizer. The buffered registers are not actively synchronized with the ADF4372 device. Use the `SYN:APPLY` command to apply the values of the buffered registers to the device.

**Syntax**

`SYN:BREGW <channel ID> <path> <register address> <register value 0> [... <register value n>]`

**Arguments**

- **channel ID**: `<0 - 3>`
- **path**: `<TX | RX>`
  - TX – transmit/upconverter
  - RX – receive/downconverter
- **register address**: `<0x00 – 0x7FFF>`
- **register value**: `<0x00 – 0xFF>`
  - Up to 8 total registers can be written, registers are addressed in ascending order.

**Example**

```
> SYN:BREGW 0 TX 0x24 0x80 0x55
OK
> SYN:BREGW 0 RX 0x26 0x32 0x33 0x22
OK
```

### 3.4.3.24 SYN:BREGR?

**Description**

Reads the buffered register values for the LO synthesizer. The buffered registers are not actively synchronized with the ADF4372 device. Use the `SYN:FETCH` command to load the values from the device into the buffered registers.

**Syntax**

`SYN:BREGR? <channel ID> <path> <register address> <read count>`

**Arguments**

- **channel ID**: `<0 - 3>`
- **path**: `<TX | RX>`
  - TX – transmit/upconverter
  - RX – receive/downconverter
- **register address**: `<0x00 – 0x7FFF>`
- **read count**: `<1 – 8>` registers are addressed in ascending order.

**Response**

- **register value(s)**: `<0x00 – 0xFF>`

**Example**

```
> SYN:BREGR? 0 TX 0x24 2
0x80 0x55
OK
> SYN:BREGR? 0 RX 0x26 3
0x32 0x33 0x22
OK
```
3.4.4 Local Oscillator

These commands configure the Local Oscillator (LO) source.

3.4.4.1 LOCLK:FREQ

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sets all configuration settings required to achieve the LO Frequency specified using the internal synthesizers.</td>
</tr>
</tbody>
</table>

This high-level command is equivalent to the following:

1. CV:LO <...> <filter bandwidth>
2. LOCLK:SEL <...> <source>
3. SYN:OUT <...> <output frequency>
4. SYN:EN <...> <RF8 enable> <RF16 enable>
5. SYN:APPLY

<table>
<thead>
<tr>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCLK:FREQ &lt;channel ID&gt; &lt;path&gt; &lt;frequency&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>• channel ID: &lt;0 - 3&gt;</td>
</tr>
<tr>
<td>• path: &lt;TX</td>
</tr>
<tr>
<td>o TX – transmit/upconverter</td>
</tr>
<tr>
<td>o RX – receive/downconverter</td>
</tr>
<tr>
<td>• frequency: &lt;21600000 – 41000000 kHz&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; LOCLK:FREQ 0 21600000</td>
</tr>
<tr>
<td>OK</td>
</tr>
</tbody>
</table>

3.4.4.2 LOCLK:FREQ?

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gets the LO Frequency when using the internal synthesizers.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCLK:FREQ? &lt;channel ID&gt; &lt;path&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>• channel ID: &lt;0 - 3&gt;</td>
</tr>
<tr>
<td>• path: &lt;TX</td>
</tr>
<tr>
<td>o TX – transmit/upconverter</td>
</tr>
<tr>
<td>o RX – receive/downconverter</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>• frequency: &lt;21600000 – 41000000 kHz&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; LOCLK:FREQ? 0 TX</td>
</tr>
<tr>
<td>21600000</td>
</tr>
<tr>
<td>OK</td>
</tr>
</tbody>
</table>
### 3.4.4.3 LOCLK:SEL

**Description**

Sets the local oscillator source of the converter.

**Syntax**

`LOCLK:SEL <channel ID> <path> <source>`

**Arguments**

- channel ID: `<0 - 3>`
- path: `<TX | RX>`
  - TX – transmit/upconverter
  - RX – receive/downconverter
- source: `<EXT | INT | INT2X>`
  - EXT: external LO connector selected
  - INT: output from internal synthesizer selected
  - INT2X: doubled output from internal synthesizer selected

**Example**

```bash
> LOCLK:SEL 0 TX INT2X
OK
> LOCLK:SEL 0 RX EXT
OK
```

### 3.4.4.4 LOCLK:SEL?

**Description**

Gets the local oscillator source selected.

**Syntax**

`LOCLK:SEL? <channel ID> <path>`

**Arguments**

- channel ID: `<0 - 3>`
- path: `<TX | RX>`
  - TX – transmit/upconverter
  - RX – receive/downconverter

**Response**

- source: `<EXT | INT | INT2X>`
  - EXT: external LO connector selected
  - INT: output from internal synthesizer selected
  - INT2X: doubled output from internal synthesizer selected

**Example**

```bash
> LOCLK:SEL? 0 TX INT2X
OK
> LOCLK:SEL? 0 RX EXT
OK
```
### 3.4.4.5 LOCLK:REF

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sets the reference clock source that is routed to the LO synthesizers.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCLK:REF &lt;channel ID&gt; &lt;source&gt; &lt;output&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>- channel ID: &lt;0 - 3&gt;</td>
</tr>
<tr>
<td>- clock source: &lt;EXT</td>
</tr>
<tr>
<td>- EXT: external reference source selected. Requires output is set to OFF.</td>
</tr>
<tr>
<td>- INT: internal reference selected</td>
</tr>
<tr>
<td>- MASTER: external Master reference selected. Requires SYS:ROLE is set to SLAVE.</td>
</tr>
<tr>
<td>- output: &lt;OFF</td>
</tr>
<tr>
<td>- OFF: output disabled; the reference connector is set as an input source.</td>
</tr>
<tr>
<td>- CMOS: output enabled; square wave output. Requires source is not set to EXT.</td>
</tr>
<tr>
<td>- SIN: output enable; sinusoid output. Requires source is not set to EXT.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; LOCLK:REF 0 INT CMOS OK</td>
</tr>
<tr>
<td>&gt; LOCLK:REF 0 EXT OFF OK</td>
</tr>
</tbody>
</table>

### 3.4.4.6 LOCLK:REF?

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gets the reference clock source that is routed to the LO synthesizers.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>LOCLK:REF? &lt;channel ID&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>- channel ID: &lt;0 - 3&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>- clock source: &lt;EXT</td>
</tr>
<tr>
<td>- EXT: external reference source selected</td>
</tr>
<tr>
<td>- INT: internal reference selected</td>
</tr>
<tr>
<td>- MASTER: external Master reference selected.</td>
</tr>
<tr>
<td>- output: &lt;OFF</td>
</tr>
<tr>
<td>- OFF: output disabled; the reference connector is set as an input source.</td>
</tr>
<tr>
<td>- CMOS: output enabled; square wave output.</td>
</tr>
<tr>
<td>- SIN: output enable; sinusoid output.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; LOCLK:REF? 0 INT CMOS OK</td>
</tr>
</tbody>
</table>
3.4.5 **RF Path**

These commands control the RF signal flow enabling FDD or TDD support.

### 3.4.5.1 RFPATH:PATH

**Description**
Sets the RF Path signal routing.

**Syntax**
RFPATH:PATH <channel ID> <path> <port>

**Arguments**
- channel ID: <0 - 3>
- path: <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter
- port: <FDD | TDD>
  - FDD: Both paths are routed to the FDD ports.
  - TDD: Both paths are routed to the TDD port, and the path specified is switched to the TDD port.
  Control the TDD port switch position independently using the RFPATH:TDD command.

**Example**

```
> RFPATH:PATH 0 RX FDD
OK
> RFPATH:PATH 0 TX TDD
OK
```

### 3.4.5.2 RFPATH:PATH?

**Description**
Gets the RF Path signal routing.

**Syntax**
RFPATH:PATH? <channel ID> <path>

**Arguments**
- channel ID: <0 - 3>
- path: <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter

**Response**
- port: <FDD | TDD>

**Example**

```
> RFPATH:PATH? 0 RX
FDD
OK
```
### 3.4.5.3 RFPATH:TDD

<table>
<thead>
<tr>
<th>Description</th>
<th>Sets the TDD port switch direction to the path specified.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>RFPATH:TDD &lt;channel ID&gt; &lt;path&gt;</td>
</tr>
<tr>
<td>Arguments</td>
<td>• channel ID: &lt;0 - 3&gt;</td>
</tr>
<tr>
<td></td>
<td>• path: &lt;TX</td>
</tr>
<tr>
<td></td>
<td>o TX – transmit/upconverter</td>
</tr>
<tr>
<td></td>
<td>o RX – receive/downconverter</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; RFPATH:TDD 0 RX</td>
</tr>
<tr>
<td></td>
<td>OK</td>
</tr>
</tbody>
</table>

### 3.4.5.4 RFPATH:TDD?

<table>
<thead>
<tr>
<th>Description</th>
<th>Gets the TDD port switch direction to the path specified.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>RFPATH:TDD? &lt;channel ID&gt;</td>
</tr>
<tr>
<td>Arguments</td>
<td>• channel ID: &lt;0 - 3&gt;</td>
</tr>
<tr>
<td>Response</td>
<td>• path: &lt;TX</td>
</tr>
<tr>
<td></td>
<td>o TX – transmit/upconverter</td>
</tr>
<tr>
<td></td>
<td>o RX – receive/downconverter</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; RFPATH:TDD? 0</td>
</tr>
<tr>
<td></td>
<td>TX</td>
</tr>
<tr>
<td></td>
<td>OK</td>
</tr>
</tbody>
</table>
3.4.6 **IF Path**

These commands control the selection of the passband filter and attenuators in the IF path.

### 3.4.6.1 **IFPATH:FLT**

**Description**

Sets the IF Filter selected in the IF Path.

**Syntax**

IFPATH:FLT <channel ID> <path> <filter>

**Arguments**

- channel ID: <0 - 3>
- path: <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter
- filter: <20 | 50 | 100 | 200 | 400 | 800 | 1400 | BYP>
  - 20: 20 MHz BW centered at 2017.5 MHz
  - 50: 50 MHz BW centered at 2345 MHz
  - 100: 100 MHz BW centered at 2350 MHz
  - 200: 200 MHz BW centered at 2593 MHz
  - 400: 400 MHz BW centered at 2300 MHz
  - 800: 800 MHz BW centered at 2400 MHz
  - 1400: 1400 MHz BW centered at 2500 MHz
  - BYP: Bypass path, no filtering

**Example**

```plaintext
> IFPATH:FLT 0 TX 50
OK
> IFPATH:FLT 0 RX 800
OK
```

### 3.4.6.2 **IFPATH:FLT?**

**Description**

Gets the IF Filter selected in the IF Path.

**Syntax**

IFPATH:FLT? <channel ID> <path>

**Arguments**

- channel ID: <0 - 3>
- path: <TX | RX>
  - TX – transmit/upconverter
  - RX – receive/downconverter

**Response**

- filter: <20 | 50 | 100 | 200 | 400 | 800 | 1400 | BYP>

**Example**

```plaintext
> IFPATH:FLT? 0 TX
50
OK
> IFPATH:FLT? 0 RX
800
OK
```
### 3.4.6.3 IFPATH:ATTN

**Description**

Sets the attenuation setting of the DSA(s) in the IF Path.

**Syntax**

```
IFPATH:ATTN <channel ID> <path> <setting 1> [<setting 2>]
```

**Arguments**

- channel ID: `<0 - 3>`
- path: `<TX | RX>`
  - TX – transmit/upconverter
  - RX – receive/downconverter
- setting1: `<0 – 31.5 dB>` in 0.5 dB steps
- setting2: `<0 – 31.5 dB>` in 0.5 dB steps *(RX Only)*

**Example**

```
> IFPATH:ATTN 0 TX 7.5
OK
> IFPATH:ATTN 0 RX 3 22.5
OK
```

### 3.4.6.4 IFPATH:ATTN?

**Description**

Gets the attenuation setting of the DSA(s) in the IF Path.

**Syntax**

```
IFPATH:ATTN? <channel ID> <path>
```

**Arguments**

- channel ID: `<0 - 3>`
- path: `<TX | RX>`
  - TX – transmit/upconverter
  - RX – receive/downconverter

**Response**

- setting1: `<0 – 31.5 dB>` in 0.5 dB steps
- setting2: `<0 – 31.5 dB>` in 0.5 dB steps *(RX Only)*

**Example**

```
> IFPATH:ATTN? 0 TX 7.5
OK
> IFPATH:ATTN? 0 RX 3 22.5
OK
```
3.4.7 **Baseband IQ Path**

These commands control the baseband IQ path variable gain amplifiers.

### 3.4.7.1 BPATH:GAIN

**Description**

Sets the gain of the variable gain amplifiers (VGA) in the baseband IQ path.

**Syntax**

```
BPATH:GAIN <channel ID> <path> <setting1> <setting2>
```

**Arguments**

- **channel ID**: <0 - 3>
- **path**: <RX> – receive/downconverter
- **setting1**: <-9 – 26 dB> in 1 dB steps
- **setting2**: <-9 – 26 dB> in 1 dB steps

**Example**

```
> BPATH:GAIN 0 RX 5 11
OK
```

### 3.4.7.2 BPATH:GAIN?

**Description**

Gets the gain of the variable gain amplifiers (VGA) in the baseband IQ path.

**Syntax**

```
BPATH:GAIN? <channel ID> <path>
```

**Arguments**

- **channel ID**: <0 - 3>
- **path**: <RX> – receive/downconverter

**Response**

- **setting1**: <-9 – 26 dB> in 1 dB steps
- **setting2**: <-9 – 26 dB> in 1 dB steps

**Example**

```
> BPATH:GAIN? 0 RX 5 11
OK
```
3.4.8 Baseband IQ Filter

These commands control the baseband IQ path filter and gain device (ADRF6520).

3.4.8.1 BFLT:FLT

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
</table>
Sets the filter selected of the ADRF6520 in the baseband IQ path.

<table>
<thead>
<tr>
<th>Syntax</th>
</tr>
</thead>
</table>
BFLT:FLT <channel ID> <path> <filter>

<table>
<thead>
<tr>
<th>Arguments</th>
</tr>
</thead>
</table>
• channel ID: <0 - 3>
• path: <RX>
  o RX – receive/downconverter
• filter: <36 | 72 | 144 | 288 | 432 | 576 | 720 | BYP>
  o 36: 36 MHz, 1dB corner
  o 72: 72 MHz, 1dB corner
  o 144: 144 MHz, 1dB corner
  o 288: 288 MHz, 1dB corner
  o 432: 432 MHz, 1dB corner
  o 576: 576 MHz, 1dB corner
  o 720: 720 MHz, 1dB corner
  o BYP: Bypass, no filter

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
</table>
> BFLT:FLT 0 RX 288
OK

3.4.8.2 BFLT:FLT?

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
</table>
Gets the filter selected of the ADRF6520 in the baseband IQ path.

<table>
<thead>
<tr>
<th>Syntax</th>
</tr>
</thead>
</table>
BFLT:FLT? <channel ID> <path>

<table>
<thead>
<tr>
<th>Arguments</th>
</tr>
</thead>
</table>
• channel ID: <0 - 3>
• path: <RX>
  o RX – receive/downconverter

<table>
<thead>
<tr>
<th>Response</th>
</tr>
</thead>
</table>
• filter: <36 | 72 | 144 | 288 | 432 | 576 | 720 | BYP>

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
</table>
> BFLT:FLT? 0 RX
288
OK
### 3.4.8.3 BFLT:GAIN

**Description**

Sets the gain, in dB, of the ADRF6520 in the baseband IQ path.

**Syntax**

BFLT:GAIN <channel ID> <path> <gain1> <gain2>

**Arguments**

- channel ID: <0 - 3>
- path: <RX>
  - RX – receive/downconverter
- gain1: <-3.5 – 26.5 dB>
- gain2: <-3.5 – 26.5 dB>

**Example**

```plaintext
> BFLT:GAIN 0 RX -1.5 -3
OK
```

### 3.4.8.4 BFLT:GAIN?

**Description**

Gets the gain, in dB, of the ADRF6520 in the baseband IQ path.

**Syntax**

BFLT:GAIN? <channel ID> <path>

**Arguments**

- channel ID: <0 - 3>
- path: <RX>
  - RX – receive/downconverter

**Response**

- gain1: <-3.5 – 26.5 dB>
- gain2: <-3.5 – 26.5 dB>

**Example**

```plaintext
> BFLT:GAIN? 0 RX
-1.50 -3.01
OK
```
### 3.4.8.5 BFLT:VGAIN

**Description**
Sets the external DAC voltage setting that controls the gain of the ADRF6520 in the baseband IQ path.

**Syntax**
BFLT:VGAIN <channel ID> <path> <gain1> <gain2>

**Arguments**
- channel ID: <0 - 3>
- path: <RX>
  - RX – receive/downconverter
- gain1: <0 – 1.5 V>
- gain2: <0 – 1.5 V>

**Example**
> BFLT:VGAIN 0 RX 1.2 1.3
OK

### 3.4.8.6 BFLT:VGAIN?

**Description**
Gets the external DAC voltage setting that controls the gain of the ADRF6520 in the baseband IQ path.

**Syntax**
BFLT:VGAIN? <channel ID> <path>

**Arguments**
- channel ID: <0 - 3>
- path: <RX>
  - RX – receive/downconverter

**Response**
- gain1: <0 – 1.5 V>
- gain2: <0 – 1.5 V>

**Example**
> BFLT:VGAIN? 0 RX 1.200 1.300
OK
### 3.4.8.7 BFLT:REGW

**Description**
Directly writes the registers of the ADRF6520 baseband IQ filter and gain device.

**Syntax**
BFLT:REGW <channel ID> <path> <register address> <register value>

**Arguments**
- channel ID: <0 - 3>
- path: <RX>
  - RX – receive/downconverter
- register address: <0x10>
- register value: <0x00 – 0xFF>

**Example**
```
> BFLT:REGW 0 RX 0x10 0x17
OK
```

### 3.4.8.8 BFLT:REGR?

**Description**
Directly reads the registers of the ADRF6520 baseband IQ filter and gain device.

**Syntax**
BFLT:REGR? <channel ID> <path> <register address>

**Arguments**
- channel ID: <0 - 3>
- path: <RX>
  - RX – receive/downconverter
- register address: <0x10>

**Response**
- register value: <0x00 – 0xFF>

**Example**
```
> BFLT:REGR? 0 RX 0x10
0x17
OK
```
### 3.4.9 External Baseband IQ Filter

These commands control the External Baseband Filter Bank, typically installed at the input of the TX Path.

#### 3.4.9.1 EBFLT:FLT

| Description |
|-------------|---|
| Sets the baseband filter selection. |

<table>
<thead>
<tr>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBFLT:FLT &lt;channel ID&gt; &lt;path&gt; &lt;filter&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>channel ID: &lt;0 - 3&gt;</td>
</tr>
<tr>
<td>path: &lt;TX&gt;</td>
</tr>
<tr>
<td>o TX – transmit/upconverter</td>
</tr>
<tr>
<td>filter: &lt;5</td>
</tr>
<tr>
<td>o 5: 5 MHz cutoff.</td>
</tr>
<tr>
<td>o 10: 10 MHz cutoff.</td>
</tr>
<tr>
<td>o 20: 20 MHz cutoff.</td>
</tr>
<tr>
<td>o 30: 30 MHz cutoff.</td>
</tr>
<tr>
<td>o 50: 50 MHz cutoff.</td>
</tr>
<tr>
<td>o 100: 100 MHz cutoff.</td>
</tr>
<tr>
<td>o 200: 200 MHz cutoff.</td>
</tr>
<tr>
<td>o BYP: Bypass, no filter.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; EBFLT:FLT 0 TX 30</td>
</tr>
<tr>
<td>OK</td>
</tr>
<tr>
<td>&gt; EBFLT:FLT 0 TX 200</td>
</tr>
<tr>
<td>OK</td>
</tr>
</tbody>
</table>

#### 3.4.9.2 EBFLT:FLT?

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gets the baseband filter selection.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>EBFLT:FLT? &lt;channel ID&gt; &lt;path&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>channel ID: &lt;0 - 3&gt;</td>
</tr>
<tr>
<td>path: &lt;TX&gt;</td>
</tr>
<tr>
<td>o TX – transmit/upconverter</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>filter: &lt;5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; EBFLT:FLT? 0 TX</td>
</tr>
<tr>
<td>30</td>
</tr>
<tr>
<td>OK</td>
</tr>
</tbody>
</table>
### 3.4.9.3 EBFLT:REGW

**Description**
Directly writes the register to control the External Baseband Filter.

**Syntax**
EBFLT:REGW <channel ID> <path> <register value>

**Arguments**
- channel ID: <0 - 3>
- path: <TX>
  - TX – transmit/upconverter
- register value: <0x00 – 0xFF>

**Example**
```
> EBFLT:REGW 0 TX 0x5
OK
```

### 3.4.9.4 EBFLT:REGR?

**Description**
Directly reads the register that controls the External Baseband Filter.

**Syntax**
EBFLT:REGR? <channel ID> <path>

**Arguments**
- channel ID: <0 - 3>
- path: <TX>
  - TX – transmit/upconverter

**Response**
- register value: <0x00 – 0xFF>

**Example**
```
> EBFLT:REGR? 0 TX
0x05
OK
```
3.4.10 **Antenna Array**

These commands control signals to an external antenna array.

3.4.10.1 **ANT:SIG**

<table>
<thead>
<tr>
<th>Description</th>
<th>Sets the external antenna array I/O signals.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>ANS:SIG &lt;channel ID&gt; &lt;signal n&gt; &lt;value n&gt; [ &lt;signal n&gt; &lt;value n&gt;]</td>
</tr>
<tr>
<td>Arguments</td>
<td>channel ID: &lt;0 - 3&gt;</td>
</tr>
<tr>
<td></td>
<td>signal n: &lt;RESET</td>
</tr>
<tr>
<td></td>
<td>o RESET: reset signal for the antenna array</td>
</tr>
<tr>
<td></td>
<td>o TRSW: transmit and receive switch control</td>
</tr>
<tr>
<td></td>
<td>o BEAM: beam control signal</td>
</tr>
<tr>
<td></td>
<td>o SPARE: unused spare signal (TBD)</td>
</tr>
<tr>
<td></td>
<td>value n: &lt;0</td>
</tr>
</tbody>
</table>

**Example**

> ANT:SIG 0 RESET 0 TRSW 1 BEAM 1 SPARE 0
OK
> ANT:SIG 0 TRSW 0
OK

3.4.10.2 **ANT:SIG?**

<table>
<thead>
<tr>
<th>Description</th>
<th>Gets the external antenna array I/O signals.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>ANS:SIG? &lt;channel ID&gt;</td>
</tr>
<tr>
<td>Arguments</td>
<td>channel ID: &lt;0 - 3&gt;</td>
</tr>
</tbody>
</table>

**Response**

| signal n: <RESET | TRSW | BEAM | SPARE> |
|               | o RESET: reset signal for the antenna array |
|               | o TRSW: transmit and receive switch control |
|               | o BEAM: beam control signal |
|               | o SPARE: unused spare signal (TBD) |
| value n: <0 | 1> |

**Example**

> ANT:SIG? 0
RESET = 0, TRSW = 0, BEAM = 1, SPARE = 0
OK
### 3.4.10.3 ANT:SPI

<table>
<thead>
<tr>
<th>Description</th>
<th>Directly writes and reads the external antenna array SPI Bus.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>ANT:SPI &lt;channel ID&gt; &lt;write data0&gt; [&lt;write data n&gt;]</td>
</tr>
<tr>
<td>Arguments</td>
<td>• channel ID: &lt;0 - 3&gt;</td>
</tr>
<tr>
<td></td>
<td>• write data n: &lt;0x00 – 0xFF&gt;</td>
</tr>
<tr>
<td></td>
<td>o write up to eight bytes</td>
</tr>
<tr>
<td>Response</td>
<td>• read data n: &lt;0x00 – 0xFF&gt;</td>
</tr>
<tr>
<td></td>
<td>o reads the equal number of bytes written</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; ANT:SPI 0 0x80 0x1f 0x3f 0x00 0x10 0x80</td>
</tr>
<tr>
<td></td>
<td>0x1a 0x2a 0x3a 0x22 0x33 0x44 OK</td>
</tr>
</tbody>
</table>

```
3.4.11  PA Bias Control

These commands control the AD7293 bias controller connected to the TX Power Amplifier (PA). The Bias Controller maintains the bias current setpoint of the PA in the Transmit (TX) path. Adjustments to the TX Bias Controller are not typically required.

3.4.11.1  BIAS:LOOP

Description
Enables the closed-loop bias control of the AD7293. This effectively turns on the Transmit Path Power Amplifier (PA).

Syntax
BIAS:LOOP <channel ID> <enable>

Arguments
- channel ID: <0 - 3>
- enable: <0 | 1>
  - 0: Disable
  - 1: Enable

Example
> BIAS:LOOP 0 1
OK

3.4.11.2  BIAS:DAC

Description
Sets the setpoint of the closed-loop bias control in DAC units.

Syntax
BIAS:DAC <channel ID> <setpoint>

Arguments
- channel ID: <0 - 3>
- setpoint: <0x000 – 0xFFF>

Example
> BIAS:DAC 0 500
OK

3.4.11.3  BIAS:DAC?

Description
Gets the setpoint of the closed-loop bias control in DAC units.

Syntax
BIAS:DAC? <channel ID>

Arguments
- channel ID: <0 - 3>

Response
- setpoint: <0x000 – 0xFFF>

Example
> BIAS:DAC? 0
DAC Level = 0x0500
OK
3.4.11.4  **BIAS:CUR**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sets the setpoint of the closed-loop bias control current in units of Amps.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIAS:CUR &lt;channel ID&gt; &lt;setpoint&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>• channel ID: &lt;0 - 3&gt;</td>
</tr>
<tr>
<td>• setpoint: &lt;0 – Current Limit (A)&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; BIAS:CUR 0 0.7</td>
</tr>
<tr>
<td>OK</td>
</tr>
</tbody>
</table>

3.4.11.5  **BIAS:CUR?**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gets the measured current of the closed-loop bias control.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIAS:CUR? &lt;channel ID&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>• channel ID: &lt;0 - 3&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>• reading: current (A)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; BIAS:CUR? 0</td>
</tr>
<tr>
<td>Current = 0.703125</td>
</tr>
<tr>
<td>OK</td>
</tr>
</tbody>
</table>

3.4.11.6  **BIAS:BVOLT?**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gets the measured closed-loop bias voltage.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIAS:BVOLT? &lt;channel ID&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>• channel ID: &lt;0 - 3&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>• reading: voltage (V)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; BIAS:BVOLT? 0</td>
</tr>
<tr>
<td>Bias Voltage = -0.63</td>
</tr>
<tr>
<td>OK</td>
</tr>
</tbody>
</table>
### 3.4.11.7 BIAS:PAVOLT?

**Description**
Gets the measured drain voltage of the PA connected to the bias controller.

**Syntax**
BIAS:PAVOLT? <channel ID>

**Arguments**
- channel ID: <0 - 3>

**Response**
- reading: voltage (V)

**Example**
```plaintext
> BIAS:PAVOLT? 0
PA Voltage = 4.97
OK
```

### 3.4.11.8 BIAS:TEMP?

**Description**
Gets the measured temperatures of the sensors connected to the bias controller.

**Syntax**
BIAS:TEMP? <channel ID>

**Arguments**
- channel ID: <0 - 3>

**Response**
- reading 0: temperature (°C)

**Example**
```plaintext
> BIAS:TEMP? 0
Sensor = 0, Temperature = 36.75
OK
```
3.4.11.9  **BIAS:WRITE**

**Description**
Directly writes the registers of the AD7293 bias controller.

**Syntax**
BIAS:WRITE <channel ID> <page> <address> <data0 value> [<data1 value>]

**Arguments**
- channel ID: <0 - 3>
- page: <0x00 – 0xFF>
- address: <0x00 – 0xFF>
- data0 value: <0x00 – 0xFF>
- data1 value: <0x00 – 0xFF>

**Example**
> BIAS:WRITE 0 0 33 5A B0
OK

3.4.11.10  **BIAS:READ?**

**Description**
Directly reads the registers of the AD7293 bias controller.

**Syntax**
BIAS:READ? <channel ID> <page> <address> <byte count>

**Arguments**
- channel ID: <0 - 3>
- page: <0x00 – 0xFF>
- address: <0x00 – 0xFF>
  - byte count: <1 - 2>

**Response**
- data0 value: <0x00 – 0xFF>
- data1 value: <0x00 – 0xFF>

**Example**
> BIAS:READ? 0 0 0x33 2
0x5a 0xb0
OK
3.4.12 SPI Direct Access

These commands provide direct access to the devices connected to the SPI bus.

3.4.12.1 SPI:CMD

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directly writes and reads the devices connected to the SPI Bus.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPI:CMD &lt;channel ID&gt; &lt;path&gt; &lt;device&gt; &lt;write data 0&gt; [&lt;write data n&gt;]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>• channel ID: &lt;0 - 3&gt;</td>
</tr>
<tr>
<td>• path: &lt;TX</td>
</tr>
<tr>
<td>o TX – transmit/upconverter</td>
</tr>
<tr>
<td>o RX – receive/downconverter</td>
</tr>
<tr>
<td>• device ID: &lt; CONV</td>
</tr>
<tr>
<td>o CONV: ADMV1013 (TX) or ADMV1014 (RX) converter.</td>
</tr>
<tr>
<td>o SYN: The synthesizer device is the ADF4372.</td>
</tr>
<tr>
<td>o FLT: The filter/VGA device is the ADRF6520</td>
</tr>
<tr>
<td>o DAC: The DAC to control gain on the ADRF6520</td>
</tr>
<tr>
<td>o ANT: External antenna array SPI device</td>
</tr>
<tr>
<td>• write data n: &lt;0x00 – 0xFF&gt;</td>
</tr>
<tr>
<td>o write up to four bytes</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>• read data n: &lt;0x00 – 0xFF&gt;</td>
</tr>
<tr>
<td>o reads the equal number of bytes written</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; SPI:CMD 0 TX CONV 0x80 0x1f 0x3f 0x1a 0x2a 0x3a OK</td>
</tr>
</tbody>
</table>
3.4.13  **Trigger Control**

3.4.13.1  **TRIG:EN**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sets the external trigger enable for the RFPATH:TDD selection and the IFPATH:ATTN (IF Mode) or BPATH:GAIN (IQ Mode) level control. When the trigger is enabled, the external trigger I/O or the TRIG:TRIG command initiates the configuration change.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIG:EN &lt;channel ID&gt; &lt;level control&gt; &lt;tdd&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>• channel ID: &lt;0 - 3&gt;</td>
</tr>
<tr>
<td>• level control: &lt;ON</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>• tdd: &lt;ON</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; TRIG:EN 0 ON ON</td>
</tr>
<tr>
<td>OK</td>
</tr>
</tbody>
</table>

3.4.13.2  **TRIG:EN?**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gets the external trigger enable for the RFPATH:TDD selection and the IFPATH:ATTN (IF Mode) or BPATH:GAIN (IQ Mode) level control.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRIG:EN? &lt;channel ID&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>• channel ID: &lt;0 - 3&gt;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>• level control: &lt;ON</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>• tdd: &lt;ON</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; TRIG:EN? 0 ON ON</td>
</tr>
<tr>
<td>OK</td>
</tr>
</tbody>
</table>
### 3.4.13.3  TRIG:TRIG

**Description**
Sets the external trigger state.

**Syntax**
TRIG:TRIG <channel ID> <level control> <tdd>

**Arguments**
- channel ID: <0 - 3>
- level control: <0 | 1>
  - 0: Trigger is deasserted.
  - 1: Trigger is asserted.
- tdd: <0 | 1>
  - 0: Trigger is deasserted.
  - 1: Trigger is asserted.

**Example**
> TRIG:TRIG 0 0 1
OK

### 3.4.13.4  TRIG:TRIG?

**Description**
Gets the external trigger state.

**Syntax**
TRIG:TRIG? <channel ID>

**Arguments**
- channel ID: <0 - 3>

**Response**
- level control: <0 | 1>
  - 0: Trigger is deasserted.
  - 1: Trigger is asserted.
- tdd: <0 | 1>
  - 0: Trigger is deasserted.
  - 1: Trigger is asserted.

**Example**
> TRIG:TRIG? 0
1 1
OK
3.4.14 Maintenance

3.4.14.1 *IDN?

<table>
<thead>
<tr>
<th>Description</th>
<th>Gets the hardware device information.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>IDN? &lt;channel ID&gt;</td>
</tr>
<tr>
<td>Arguments</td>
<td>- channel ID: &lt;0 - 3&gt;</td>
</tr>
<tr>
<td>Response</td>
<td>- manufacture: the device manufacture</td>
</tr>
<tr>
<td></td>
<td>- part number: the device part number</td>
</tr>
<tr>
<td></td>
<td>- serial number: the device serial number</td>
</tr>
<tr>
<td></td>
<td>- firmware version: the device firmware version</td>
</tr>
<tr>
<td></td>
<td>- hardware revision: the hardware revision</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; *IDN? 0</td>
</tr>
<tr>
<td></td>
<td>Signalcraft Technologies, SC2444, #H8D550001, 0.27, 1.0</td>
</tr>
<tr>
<td></td>
<td>OK</td>
</tr>
</tbody>
</table>

3.4.14.2 *RST

<table>
<thead>
<tr>
<th>Description</th>
<th>Performs a hardware reset, restarting the system and restoring default settings.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>RST &lt;channel ID&gt;</td>
</tr>
<tr>
<td>Arguments</td>
<td>- channel ID: &lt;0 - 3&gt;</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; *RST 0</td>
</tr>
</tbody>
</table>

3.4.14.3 MAINT:GETMANUF?

<table>
<thead>
<tr>
<th>Description</th>
<th>Gets the hardware manufacturing information.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syntax</td>
<td>MAINT:GETMANUF? &lt;channel ID&gt;</td>
</tr>
<tr>
<td>Arguments</td>
<td>- channel ID: &lt;0 - 3&gt;</td>
</tr>
<tr>
<td>Response</td>
<td>- serial number: the device serial number</td>
</tr>
<tr>
<td></td>
<td>- date: the date of manufacture using the format YYYY-MM-DD</td>
</tr>
<tr>
<td></td>
<td>- revision: the hardware revision</td>
</tr>
<tr>
<td>Example</td>
<td>&gt; MAINT:GETMANUF? 0</td>
</tr>
<tr>
<td></td>
<td>8D550001 2022-02-03 1.0</td>
</tr>
<tr>
<td></td>
<td>OK</td>
</tr>
</tbody>
</table>
4. Binary Protocol

The Binary Protocol operates over the Expansion Control Port SPI Interface. This protocol is intended as an alternative suitable for interfacing with an external microcontroller or FPGA. Additionally, the SPI interface operates at higher speeds required for time-critical applications. The UDC is referred to as the Slave Device whereas an external microcontroller or FPGA is the Master Device.

4.1 Slave Timing

The Master Device must observe the Slave Timing details listed below and illustrated in Figure 6.

<table>
<thead>
<tr>
<th>Number</th>
<th>Symbol</th>
<th>Description</th>
<th>Min</th>
<th>Max</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$F_{SCK}$</td>
<td>Frequency of operation</td>
<td>-</td>
<td>15</td>
<td>MHz</td>
</tr>
<tr>
<td>2</td>
<td>$T_{SCK}$</td>
<td>SCK period</td>
<td>$1 / F_{SCK}$</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>3</td>
<td>$T_{LEAD}$</td>
<td>Enable lead time</td>
<td>$T_{SCK} / 2$</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>4</td>
<td>$T_{HSCK}$</td>
<td>SCK high time</td>
<td>$(T_{SCK} / 2) - 5$</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>5</td>
<td>$T_{SETUP}$</td>
<td>Data setup time</td>
<td>2.7</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>6</td>
<td>$T_{HOLD}$</td>
<td>Data hold time</td>
<td>3.8</td>
<td>-</td>
<td>ns</td>
</tr>
<tr>
<td>7</td>
<td>$T_{VALID}$</td>
<td>Data valid after SCK edge (x4 slaves)</td>
<td>-</td>
<td>30.0</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data valid after SCK edge (x3 slaves)</td>
<td>-</td>
<td>25.6</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data valid after SCK edge (x2 slaves)</td>
<td>-</td>
<td>21.2</td>
<td>ns</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Data valid after SCK edge (x1 slave)</td>
<td>-</td>
<td>16.8</td>
<td>ns</td>
</tr>
<tr>
<td>8</td>
<td>$T_{ACCESS}$</td>
<td>Slave access time</td>
<td>-</td>
<td>$T_{LEAD} + T_{VALID}$</td>
<td>ns</td>
</tr>
</tbody>
</table>

![Binary Protocol Slave Timing](image)

Figure 6 - Binary Protocol Slave Timing
4.2 Protocol Sequence

The protocol sequence consists of four individual 32-bit transactions, for a total of 128-bits per sequence. Figure 7 illustrates the Binary Protocol Sequence. The SPI Chip Select (CS#) line must be asserted for the full duration of the sequence. Data is both sent to and received from all the slave device(s) within a single sequence.

4.2.1 Slave Responses

Up to four slave devices can be vertically connected to the Expansion Control Port SPI Interface. The order of the Response Transactions is dependent on the number and the physical order of slave devices connected. Figure 8 illustrates the position of each Response Transaction where Slave A represents the top of the vertical stack. The Command Transaction order is the same regardless of the number of slave devices connected.
4.2.2 Write Transaction

The Write Transaction consists of a Command and a Response. See Figure 9 for the Write Command Sequence.

![Figure 9 – Binary Protocol Write Command](image)

After the Write Command is processed by the slave device, the Write Response is returned on the next transaction. See Figure 10 for the Write Response Transaction.

![Figure 10 – Binary Protocol Write Response](image)

4.2.3 Read Transaction

The Read Transaction consists of a Command and a Response. The Read command specifies the Command ID along with optional Read Arguments as defined in Section 4.4. See Figure 11 for the Read Command Transaction.

![Figure 11 – Binary Protocol Read Command](image)

After issuing the Read Command to a slave device, the Read Response is returned on the next transaction. See Figure 12 for the Read Response Transaction.

![Figure 12 – Binary Protocol Read Response](image)
4.3 Transaction Data

The Command Transaction consists of the following:

<table>
<thead>
<tr>
<th>Bits</th>
<th>[31..24]</th>
<th>[23..0]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Command Byte</td>
<td>Write Data Bytes or Read Arguments</td>
</tr>
</tbody>
</table>

- Command Byte – Identifies the Operation (read or write) and the Command ID.
- Data Bytes – Write data bytes or Read Arguments.

The Response Transaction consists of the following:

<table>
<thead>
<tr>
<th>Bits</th>
<th>[31..24]</th>
<th>[23..0]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Response Byte</td>
<td>Read Data Bytes</td>
</tr>
</tbody>
</table>

- Response Byte – Indicates the previous transaction status and the Command ID.
- Data Bytes – Read data bytes.

4.3.1 Command Byte

The Command Byte is structured as follows:

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Operation 0: Write 1: Read</td>
<td>Command ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.3.2 Response Byte

The Response Byte is structured as follows:

<table>
<thead>
<tr>
<th>Bit</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Status 0: OK 1: Error</td>
<td>Command ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

4.3.3 Data Bytes

Read and Write operations contain three data bytes. The data bytes are sent onto the SPI bus with the Most Significant Byte (MSB) sent first. All data values are always aligned to the LSB of the data word.

<table>
<thead>
<tr>
<th>Bits</th>
<th>[31..24]</th>
<th>[23..16]</th>
<th>[15..8]</th>
<th>[7..0]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write</td>
<td>Command Byte</td>
<td>Data Byte 0 (MSB)</td>
<td>Data Byte 1</td>
<td>Data Byte 2 (LSB)</td>
</tr>
<tr>
<td>Read</td>
<td>Response Byte</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.4 Command Details

Below is a summary of the Command ID values.

<table>
<thead>
<tr>
<th>ID</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x00</td>
<td>Null Operation</td>
</tr>
<tr>
<td>0x01</td>
<td>TDD IF Configuration</td>
</tr>
<tr>
<td>0x02</td>
<td>TDD Baseband Configuration</td>
</tr>
<tr>
<td>0x03</td>
<td>Active Path Select</td>
</tr>
<tr>
<td>0x04</td>
<td>Hardware Control</td>
</tr>
<tr>
<td>0x05</td>
<td>Hardware Status</td>
</tr>
<tr>
<td>0x10</td>
<td>Converter Register Access</td>
</tr>
<tr>
<td>0x11</td>
<td>Synthesizer Register Access</td>
</tr>
<tr>
<td>0x12</td>
<td>Baseband IQ Filter Register Access</td>
</tr>
</tbody>
</table>

4.4.1 0x00 - Null Operation

The Null Operation does not perform any action and does not cause the slave device to change state. The Null Operation can be used to clock out any pending response data from a slave device without requiring an additional read or write operation.

The Null Operation can also be used to return the SPI Bus to a clean state when the synchronization between the Master and Slave devices is unknown. For example, if four slave devices are connected to the Bus, performing four Null Operations will ensure all slave devices are ready for new data.

4.4.2 0x01 - TDD IF Configuration

IF Path configuration settings that are required for the time-critical TDD operation.

<table>
<thead>
<tr>
<th>Bit(s)</th>
<th>Name</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>[23..21]</td>
<td>CFG_EN</td>
<td>Enable or disable the configuration setting specified.</td>
<td>W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0b = Disable (the value specified is ignored)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 1b = Enable</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Below are the enable bits for each group of settings.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 001b = RX_ATTN</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 010b = TX_ATTN</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 100b = TDD_SW</td>
<td></td>
</tr>
<tr>
<td>[20..19]</td>
<td>RESERVED</td>
<td>Reserved.</td>
<td>R</td>
</tr>
<tr>
<td>18</td>
<td>TDD_SW</td>
<td>Controls the position of the TDD RF Switch Position.</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0b = RX</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 1b = TX</td>
<td></td>
</tr>
<tr>
<td>[17..12]</td>
<td>TX_ATTN_0</td>
<td>IF Attenuation Control specified in 0.5 dB steps (0 to 63).</td>
<td>R/W</td>
</tr>
<tr>
<td>[11..6]</td>
<td>RX_ATTN_1</td>
<td>For example: (22.5 dB / 0.5 dB) = 45 steps = 0x2D</td>
<td>R/W</td>
</tr>
<tr>
<td>[5..0]</td>
<td>RX_ATTN_0</td>
<td></td>
<td>R/W</td>
</tr>
</tbody>
</table>
4.4.3 0x02 - TDD Baseband Configuration

Baseband Path configuration settings that are required for the time-critical TDD operation.

<table>
<thead>
<tr>
<th>Bit(s)</th>
<th>Name</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>[23..21]</td>
<td>CFG_EN</td>
<td>Enable or disable the configuration setting specified.</td>
<td>W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0b = Disable (the value specified is ignored)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 1b = Enable</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Below are the enable bits for each group of settings.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 001b = RX_GAIN</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 100b = TDD_SW</td>
<td></td>
</tr>
<tr>
<td>[20..19]</td>
<td>RESERVED</td>
<td>Reserved.</td>
<td>R</td>
</tr>
<tr>
<td>18</td>
<td>TDD_SW</td>
<td>Controls the position of the TDD RF Switch Position.</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0b = RX</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 1b = TX</td>
<td></td>
</tr>
<tr>
<td>[17..12]</td>
<td>RESERVED</td>
<td>Reserved.</td>
<td>R</td>
</tr>
<tr>
<td>[11..6]</td>
<td>RX_GAIN_1</td>
<td>Baseband Gain Control is specified in 1 dB steps (-9 to 26). The setting is</td>
<td>R/W</td>
</tr>
<tr>
<td>[5..0]</td>
<td>RX_GAIN_0</td>
<td>calculated using the following formula: (26 - Gain) = Setting</td>
<td></td>
</tr>
</tbody>
</table>

4.4.4 0x03 - Active Path Select

Sets the Active Path associated with the register values in the following sections.

<table>
<thead>
<tr>
<th>Bit(s)</th>
<th>Name</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>[23..1]</td>
<td>RESERVED</td>
<td>Reserved.</td>
<td>R</td>
</tr>
<tr>
<td>0</td>
<td>PATH</td>
<td>Selects the Active Path for the subsequent register access.</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0b = RX</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 1b = TX</td>
<td></td>
</tr>
</tbody>
</table>

4.4.5 0x04 - Hardware Control

Hardware Control for the Active Path Selected (Section 4.4.4).

<table>
<thead>
<tr>
<th>Bit(s)</th>
<th>Name</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>[23..16]</td>
<td>ID</td>
<td>Hardware Control ID</td>
<td>R/W</td>
</tr>
<tr>
<td>[15..0]</td>
<td>VALUE</td>
<td>The value associated with the Hardware Control ID</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Read the Hardware Control ID value specified using the following read arguments.

<table>
<thead>
<tr>
<th>Bit(s)</th>
<th>Name</th>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>[23..16]</td>
<td>ID</td>
<td>Hardware Control ID</td>
<td></td>
</tr>
<tr>
<td>[15..0]</td>
<td>RESERVED</td>
<td>Reserved.</td>
<td></td>
</tr>
</tbody>
</table>

4.4.5.1 Control ID

Below are the Hardware Control IDs and associated Values.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01</td>
<td>REF_SRC</td>
<td>Reference source selection for the internal synthesizers.</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x00 = Internal Oscillator</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x01 = External Reference (Requires REF_OUT = 0x00)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x02 = Master Reference</td>
<td></td>
</tr>
<tr>
<td>0x02</td>
<td>REF_OUT</td>
<td>Reference output enable.</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x00 = Disable</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x01 = Enable CMOS (Requires REF_SRC != 0x01)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x02 = Enable Sinusoid (Requires REF_SRC != 0x01)</td>
<td></td>
</tr>
<tr>
<td>ID</td>
<td>Name</td>
<td>Description</td>
<td>Access</td>
</tr>
<tr>
<td>-----</td>
<td>-----------</td>
<td>------------------------------------------------------------------------------</td>
<td>--------</td>
</tr>
<tr>
<td>0x03</td>
<td>LO_SRC</td>
<td>LO source selection for the converter of the Active Path.</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x00 = External</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x01 = Synthesizer (Primary)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x02 = Synthesizer (Doubled)</td>
<td></td>
</tr>
<tr>
<td>0x04</td>
<td>RF_PATH</td>
<td>RF Path Select for the TX and RX Paths.</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x00 = FDD</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x01 = TDD</td>
<td></td>
</tr>
<tr>
<td>0x05</td>
<td>RF_TDD</td>
<td>RF TDD Path Select.</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x00 = RX</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x01 = TX</td>
<td></td>
</tr>
<tr>
<td>0x06</td>
<td>EXT_TRIG</td>
<td>External Trigger Enable bits for the TDD Configuration Registers (Sections 4.4.2 and 4.4.3) When the trigger is enabled, the external trigger signal controlled by the Master initiates the configuration changes.</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 01b = RX_GAIN_0/1 or TX_ATTN_0 and RX_ATTN[0/1]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 10b = TDD_SW</td>
<td></td>
</tr>
<tr>
<td>0x07</td>
<td>IF_FILTER</td>
<td>IF Filter selection for the Active Path.</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x00 = 20 MHz BW, F_c = 2017.5 MHz</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x01 = 50 MHz BW, F_c = 2345 MHz</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x02 = 100 MHz BW, F_c = 2350 MHz</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x03 = 200 MHz BW, F_c = 2593 MHz</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x04 = 400 MHz BW, F_c = 2300 MHz</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x05 = 800 MHz BW, F_c = 2400 MHz</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x06 = 1400 MHz BW, F_c = 2500 MHz</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x07 = Bypass path, no filtering</td>
<td></td>
</tr>
<tr>
<td>0x08</td>
<td>BB_FILTER</td>
<td>Baseband Filter selection for the RX Path.</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x00 = 36 MHz, 1dB corner</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x01 = 72 MHz, 1dB corner</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x02 = 144 MHz, 1dB corner</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x03 = 288 MHz, 1dB corner</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x04 = 432 MHz, 1dB corner</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x05 = 576 MHz, 1dB corner</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x06 = 720 MHz, 1dB corner</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x07 = Bypass, no filter</td>
<td></td>
</tr>
<tr>
<td>0x09</td>
<td>EX_BB_FLT</td>
<td>External Baseband Filter selection for the TX Path.</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x00 = 5 MHz cutoff</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x01 = 10 MHz cutoff</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x02 = 20 MHz cutoff</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x03 = 30 MHz cutoff</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x04 = 50 MHz cutoff</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x05 = 100 MHz cutoff</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x06 = 200 MHz cutoff</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0x07 = Bypass, no filter</td>
<td></td>
</tr>
<tr>
<td>0x0A</td>
<td>ANT_OUT</td>
<td>Enable or Disable the Antenna Array outputs.</td>
<td>R/W</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0b = Disable (the output is low)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 1b = Enable (the output is high)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Below are the bits to control each output.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0001b = RESET</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0010b = TRSW</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0100b = BEAM</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 1000b = SPARE</td>
<td></td>
</tr>
<tr>
<td>0x0B</td>
<td>CV_GAINV</td>
<td>The DAC voltage is used to set the gain of the converter of the Active Path. The voltage is specified in mV. For example: 1800 mV = 0x0708</td>
<td>R/W</td>
</tr>
</tbody>
</table>

**Table notes:**
- **0x00, 0x01, 0x02:** These values select the source for the LO signal.
- **R/W:** Indicates Read/Write access.
- **F_c:** Center frequency of the IF filter.
- **R:** Indicates Read-only access.
- **W:** Indicates Write-only access.
<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x0C</td>
<td>BB_VGN1</td>
<td>The DAC voltage is used to control the gain of the ADRF6520 (VGN1 Pin) for the RX Path. The voltage is specified in mV. For example: 1500 mV = 0x05DC. A 10 msec delay is required after a read transaction and before issuing another Transaction.</td>
<td>R/W</td>
</tr>
<tr>
<td>0x0D</td>
<td>BB_VGN2</td>
<td>The DAC voltage is used to control the gain of the ADRF6520 (VGN2 Pin) for the RX Path. The voltage is specified in mV. For example: 1500 mV = 0x05DC. A 10 msec delay is required after a read transaction and before issuing another Transaction.</td>
<td>R/W</td>
</tr>
</tbody>
</table>
| 0x0E| PA_EN   | Enable or disable the closed-loop bias control of the PA for the TX Path. A 200 msec delay is required before issuing another Transaction.  
- 0x00 = Disable  
- 0x01 = Enable | W      |
| 0x0F| PA_CUR_SP| The PA Bias current setpoint for the TX Path. The current is specified in mA. For example: 700 mA = 0x02BC. A 30 msec delay is required before issuing another Transaction. | W      |
| 0xFF| HW_RST | Performs a hardware reset, restarting the system and restoring default settings.  
- 0xA57B = Reset Hardware | W      |
4.4.6 0x05 – Hardware Status

<table>
<thead>
<tr>
<th>Bit(s)</th>
<th>Name</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>[23..16]</td>
<td>ID</td>
<td>Hardware Status ID</td>
<td>R/W</td>
</tr>
<tr>
<td>[15..0]</td>
<td>VALUE</td>
<td>The value associated with the Hardware Control ID</td>
<td>R</td>
</tr>
</tbody>
</table>

Read the Hardware Status ID value specified using the following read arguments.

<table>
<thead>
<tr>
<th>Bit(s)</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[23..16]</td>
<td>ID</td>
<td>Hardware Status ID</td>
</tr>
<tr>
<td>[15..0]</td>
<td>RESERVED</td>
<td>Reserved.</td>
</tr>
</tbody>
</table>

4.4.6.1 Status ID

Below are the Hardware Status IDs.

<table>
<thead>
<tr>
<th>ID</th>
<th>Name</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x01</td>
<td>CH_ID</td>
<td>The Channel ID assigned to the device for the Console Protocol is described in Section 3.</td>
<td>R</td>
</tr>
<tr>
<td>0x02</td>
<td>SERIAL_1</td>
<td>The device serial number upper bytes (MSB). For example: the serial number 0xABC1234 = 0xABC.</td>
<td>R</td>
</tr>
<tr>
<td>0x03</td>
<td>SERIAL_2</td>
<td>The device serial number lower bytes (LSB). For example: the serial number 0xABC1234 = 0x1234.</td>
<td>R</td>
</tr>
<tr>
<td>0x04</td>
<td>HW_REV</td>
<td>The device hardware revision where the MSB is the major version and the LSB is the minor version. For example: the revision 1.2 = 0x0102</td>
<td>R</td>
</tr>
<tr>
<td>0x05</td>
<td>PA_CUR</td>
<td>The measured PA Bias current for the TX Path. The current is indicated in mA. For example: 700 mA = 0x02BC</td>
<td>R</td>
</tr>
</tbody>
</table>
4.4.7  0x10 - Converter Register Access

Direct access to the integrated converter (ADMV1013 or ADMV1014) for the Active Path Selected (Section 4.4.4).

<table>
<thead>
<tr>
<th>Bit(s)</th>
<th>Name</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>[23..16]</td>
<td>ADDRESS</td>
<td>Register Address</td>
<td>R/W</td>
</tr>
<tr>
<td>[15..0]</td>
<td>VALUE</td>
<td>Register Value</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Read the converter registers using the read arguments specified below.

<table>
<thead>
<tr>
<th>Bit(s)</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[23..16]</td>
<td>ADDRESS</td>
<td>Register Address</td>
</tr>
<tr>
<td>[7..0]</td>
<td>RESERVED</td>
<td>Reserved.</td>
</tr>
</tbody>
</table>

4.4.8  0x11 - Synthesizer Register Access

Direct access to the integrated synthesizer (ADF4372) for the Active Path Selected (Section 4.4.4).

<table>
<thead>
<tr>
<th>Bit(s)</th>
<th>Name</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>[23..8]</td>
<td>ADDRESS</td>
<td>Register Address</td>
<td>R/W</td>
</tr>
<tr>
<td>[7..0]</td>
<td>VALUE</td>
<td>Register Value</td>
<td>R/W</td>
</tr>
</tbody>
</table>

Read the synthesizer registers using the read arguments specified below.

<table>
<thead>
<tr>
<th>Bit(s)</th>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[23..8]</td>
<td>ADDRESS</td>
<td>Register Address</td>
</tr>
<tr>
<td>[15..0]</td>
<td>RESERVED</td>
<td>Reserved.</td>
</tr>
</tbody>
</table>

4.4.9  0x12 - Baseband IQ Filter Register Access

Direct access to the integrated Baseband IQ Filter (ADRF6520) for the Active Path Selected (Section 4.4.4).

<table>
<thead>
<tr>
<th>Bit(s)</th>
<th>Name</th>
<th>Description</th>
<th>Access</th>
</tr>
</thead>
<tbody>
<tr>
<td>[23..8]</td>
<td>RESERVED</td>
<td>Reserved.</td>
<td>R</td>
</tr>
<tr>
<td>[7..0]</td>
<td>VALUE</td>
<td>Register Value</td>
<td>R/W</td>
</tr>
</tbody>
</table>
5. Appendix

5.1 Console Protocol Examples

5.1.1 Setup 1 – IF Mode FDD Loopback

This example describes an IF loopback test connection using the FDD path.

<table>
<thead>
<tr>
<th>Important</th>
</tr>
</thead>
<tbody>
<tr>
<td>To avoid damage to the RX path, ensure at least 30 dB of external attenuation is present on the FDD path input.</td>
</tr>
</tbody>
</table>

5.1.1.1 Details

Reference:
- Configure the reference signal to the internal 10 MHz
  - Configure the Synthesizer PFD for 20 MHz (optimal)

Upconversion:
- IF input of 2.345 GHz with a bandwidth of 50 MHz
  - Configure IF input filter to 50 MHz
- RF output of 26.985 GHz (upper sideband)
- Configure the internal synthesizer for an LO of 24.650 GHz
  - Enables the RF8 output of the synthesizer
  - Switches the RF8 output to the Converter LO input
  - Configures the Synthesizer output to 6.160 GHz
    - The converter multiplies the LO by 4x
    - \( (\text{Frequency} / \text{PFD}) = \text{Integer Value} \) for best performance
  - Configures the Converter to use the LO Filter 5.4 GHz – 7 GHz

Downconversion:
- RF input of 26.985 GHz (upper sideband)
- IF output of 2.345 GHz with a bandwidth of 50 MHz
  - Configure IF output filter to 50 MHz
- Configure the internal synthesizer for an LO of 24.650 GHz
  - Enables the RF8 output of the synthesizer
  - Switches the RF8 output to the Converter LO input
  - Configures the Synthesizer output to 6.160 GHz
    - The converter multiplies the LO by 4x
    - \( (\text{Frequency} / \text{PFD}) = \text{Integer Value} \) for best performance
  - Configures the Converter to use the LO Filter 5.4 GHz – 7 GHz

Output:
- Select FDD mode.
5.1.1.2 Sequence

// Setup the system as a Master with 1 Channel
SYS:ROLE 0 MASTER
SYS:NCHAN 0 1

// Configure the System Reference to 10 MHz
SYS:REF 0 10000
LOCLK:REF 0 INT OFF

// Configure the Synthesizer PFD to 20 MHz
SYN:PFDSET 0 TX 1 1 0
SYN:PFDSET 0 RX 1 1 0

// Configure the LO Frequency
LOCLK:FREQ TX 24650000
LOCLK:FREQ RX 24650000

// Configure the Converter Mode
CV:MODE 0 TX IF
CV:MODE 0 RX IF

// Configure the IF Filters
IFPATH:FLT 0 TX 50
IFPATH:FLT 0 RX 50

// Configure the Converter Gain
// Assumes: TX input power <= -40 dBm
CV:VGAIN 0 TX 1.8
CV:VGAIN 0 RX 0.0

// Configure the IF Attenuation
// Assumes: TX input power <= -40 dBm
IFPATH:ATTN 0 TX 0
IFPATH:ATTN 0 RX 22 22

// Configure the RF Path
RFPATH:PATH 0 FDD
5.1.2 Setup 2 – Baseband IQ TDD Conversion

This example describes a Baseband configuration using the TDD path.

5.1.2.1 Details

Reference:
- Configure the reference signal to the internal 10 MHz
  - Configure the Synthesizer PFD for 20 MHz (optimal)

Upconversion:
- Baseband input with a bandwidth of 170 MHz
  - External Baseband filter connected
- RF output of 34.08 GHz
- Configure the internal synthesizer for an LO of 34.08 GHz
  - Enables the RF16 output of the synthesizer
  - Switches the RF16 output to the Converter LO input
  - Configures the Synthesizer output to 4.26 GHz
    - The synthesizer RF16 multiplies the output by 2x
    - The converter multiplies the LO by 4x
    - \( \frac{\text{Frequency}}{\text{PFD}} = \text{Integer Value} \) for best performance
  - Configures the Converter to use the LO Filter 6.6 GHz – 9.2 GHz

Downconversion:
- RF input of 40 GHz
- Baseband output with a bandwidth of 500 MHz
- Configure the internal synthesizer for an LO of 40 GHz
  - Enables the RF16 output of the synthesizer
  - Switches the RF16 output to the Converter LO input
  - Configures the Synthesizer output to 5 GHz
    - The synthesizer RF16 multiplies the output by 2x
    - The converter multiplies the LO by 4x
    - \( \frac{\text{Frequency}}{\text{PFD}} = \text{Integer Value} \) for best performance
  - Configures the Converter to use the LO Filter 8.62 GHz – 10.25 GHz

Output:
- Select TDD mode.
5.1.2.2 Sequence

// Setup the system as a Master with 1 Channel
SYS:ROLE 0 MASTER
SYS:NCHAN 0 1

// Configure the System Reference to 10 MHz
SYS:REF 0 10000
LOCLK:REF 0 INT OFF

// Configure the Synthesizer PFD to 20 MHz
SYN:PFDSET 0 TX 1 1 0
SYN:PFDSET 0 RX 1 1 0

// Configure the LO Frequency
LOCLK:FREQ TX 34080000
LOCLK:FREQ RX 40000000

// Configure the Converter Mode
CV:MODE 0 TX IQ
CV:MODE 0 RX IQ

// Configure the IQ Filters
// TX External IQ Filter with 200 MHz cutoff
EBFLT:FLT 0 TX 200
// RX Internal IQ Filter with 576 MHz cutoff
BFLT:FLT 0 RX 576

// Configure the Converter Gain
// Assumes: TX input power <= -40 dBm
CV:VGAIN 0 TX 1.8
CV:VGAIN 0 RX 0.0

// Configure the RX Gain
BFLT:GAIN 0 RX 25 10

// Configure the RX TDD Path (to receive)
// Configure the TX TDD Path (to transmit)
RFPATH:TDD 0 RX
RFPATH:TDD 0 TX