moglabs

a santec company

Agile RF Synthesizer & AOM driver

XRF421

Revision 1.8.4
Serials A09xxx and newer
Firmware v1.12.2

Limitation of Liability

Copyright

Contact

For further information, please contact:

MOG Laboratories P/L
49 University St
Carlton VIC 3053
AUSTRALIA
+61 3 9939 0677
info@moglabs.com
www.moglabs.com

Santec LIS Corporation
5823 Ohkusa-Nenjozaka, Komaki
Aichi 485-0802
JAPAN
+81 568 79 3535
www.santec.com

Preface

Acousto-optic modulators (AOMs) are an integral part of many laser-based experiments. They are used for frequency shifting, amplitude modulation, and laser frequency stabilisation. Many experiments require very simple control of the RF frequency and power, but others require sophisticated sequences. The MOGLabs XRF agile RF synthesizer provides such complexity with a user-friendly interface. The extraordinary capabilities of the XRF have not previously been available from any single supplier, let alone in a single unit. Two channels, with direct output of up to 4 W per channel. Wide frequency range of 20 to 400 MHz. Arbitrary frequency, amplitude and phase with high resolution. Analogue modulation of each channel, in frequency, amplitude, and/or phase, with 10 MHz bandwidth. Ergonomic front-panel controls, and ethernet/USB interface. Table-mode operation to define complex time-dependent waveform output. All in one box which connects directly to AC mains power and to your AOMs. As you delve into this manual you will uncover more and more capability, but the powerful FPGA at the heart of the XRF allows software improvements to add new features, so please check the MOGLabs website for updates, example code, and assistance.

We hope that you enjoy using the XRF, and please let us know if you have any suggestions for improvement in the XRF or in this document, so that we can make life in the lab better for all.

MOGLabs, Melbourne, Australia
www.moglabs.com

Safety Precautions

Safe and effective use of this product is very important. Please read the following safety information before attempting to operate. Also please note several specific and unusual cautionary notes before using the MOGLabs XRF, in addition to the safety precautions that are standard for any electronic equipment.

CAUTION To ensure correct cooling airflow, the unit should not be operated with cover removed.

WARNING High voltages are exposed internally, particularly around the mains power inlet and internal power supply unit. The unit should not be operated with cover removed.

NOTE The MOGLabs XRF is designed for use in scientific research laboratories. It should not be used for consumer or medical applications.

Protection Features

The MOGLabs XRF includes a number of features to protect you and your device.

Open/short circuit Each RF output should be connected to a 50 Ω load. The XRF will disable each high-power RF output if a short-circuit is detected.

Mains filter Protection against mains transients.

Temperature Several temperature sensors control the fan and will trigger a shutdown if the temperature exceeds a safe limit.

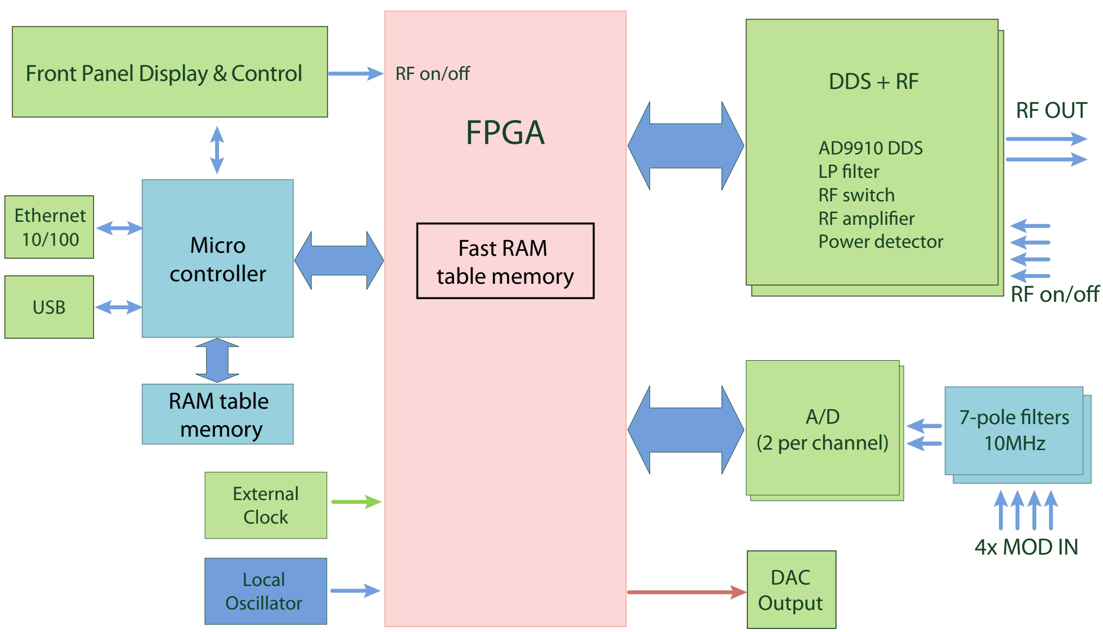
Contents

1. Introduction

The MOGLabs XRF consists of two independent AD9910 direct digital synthesizer (DDS) sources, each with 4 W amplifier. The frequency, amplitude and phase of each output is software-controlled via a microcontroller and FPGA (field programmable gate array).

This enables direct control of the frequency, amplitude and phase of the RF signals, which can be adjusted in real-time using the front-panel controls or via a scripting language over ethernet or USB. The RF parameters can be defined in a lookup table to enable complex sequences with very fast transitions.

The block diagram below shows the key components of the system.



The RF signal output from each DDS is low-pass filtered, pre-amplified, and then further amplified with a high-power output stage. The DDS chips are controlled by the FPGA. A microcontroller provides external interface with TCPIP and USB communications, and controls the front-panel display, rotary encoders (knobs) and push-buttons.

The device allows analogue modulation through two analogue-to-digital converters (ADC) with anti-aliasing filters. When modulation is enabled, the FPGA periodically reads the value of the modulation signal and uses that value to reprogram the DDS frequency, power and/or phase.

The XRF includes memory for storing complex waveform sequences, where each step in the sequence can include frequency, power, phase, time delay, and more complex definitions of ramps and other time-dependent functions. Complex capabilities can be accessed via either TCPIP or USB communications. See Chapter 3 for information on communications options and setup.

Once communications are established, the XRF can be controlled with simple text commands. The commands can be very basic, for example to define the frequency or power, or they can define complex dynamic sequences. Appendix C provides a summary of the available commands.

1.1 Operating modes

The XRF can be used at varying levels of complexity, as either a free-running RF source or to follow pre-determined instructions defined in a table. The modes of operation are outlined below, and the current operational mode of each channel can be individually set using the `MODE` command.

NSB: Basic mode

Default state on power-up. In this mode, each channel acts as a simple single-frequency RF source, with the DDS chips controlled directly by the FPGA. The frequency and power of the signal can be controlled via the front panel, using simple instructions over the computer interface (e.g. `FREQ` or `POW`), or using the modulation inputs. Basic mode is convenient for driving AOMs and other single-frequency devices, with the flexibility of modulation and PID control.

NSA: Advanced mode

Advanced mode provides direct user-control of each DDS through its internal registers via the `DDS` command. Direct programming of each DDS is complex and not necessary for most applications; it requires careful reference to the AD9910 datasheet and manual calculation of the hardware registers.

TSB: Simple table mode

In table mode, the RF parameters are automatically sequenced by the FPGA according to a table of values pre-loaded by the microcontroller and stepped through automatically. The table entries are defined by simple text commands from the host computer which define the RF frequency, amplitude, phase and any I/O behaviour, as detailed in chapter 8. The minimum duration of a TSB entry is 1 μs and each table can comprise up to 8191 instructions.

TPA: Advanced table mode

XRF models provide a more advanced table mode with greatly improved timing resolution and single parameter updates at 16 ns intervals. Smooth pulses can be generated with precise control of the envelope through piecewise-linear interpolation. Details on advanced table mode functionality are described in chapter 9.

1.2 Feature compatibility

The XRF provides a wide range of functionality, but not all features are compatible with each other. The following table summarises which features can be used in which modes.

	NSB	NSA	TSB	TPA
Front-panel controls	✓	✗	✗	✗
External modulation (AM/FM/PM)	✓	✗	✗	✗
PID control	✓	✗	✗	✗
Direct TTL on/off control	✓	✓	✗	✗
Direct DDS control	✗	✓	✗	✗
Autonomous execution	✗	✗	✓	✓
External TTL trigger	✗	✗	✓	✓
TTL output	✓	✓	✓	✓

Table 1.1: Summary of feature compatibility

1.3 RF on/off control

The RF output can be turned on and off via software control of the DDS generators, but for many applications that is too slow and the extinction ratio is inadequate. The XRF has additional hardware-based on/off control on the output of each DDS, using an RF switch before each amplifier.

Each hardware switch terminates the RF output into an internal 50 Ω load, and is controlled via a combination of software and hardware inputs (see §7.6). In this way, the RF can be controlled using the front panel, TTL signals, software commands, or table entries.

There is additional control of the DC supplies to the high-power RF amplifiers to further improve the extinction ratio. The response time is significantly longer than just switching the RF signal, but reduces the RF noise on the output.

2. Connections and controls

2.1 Front panel controls

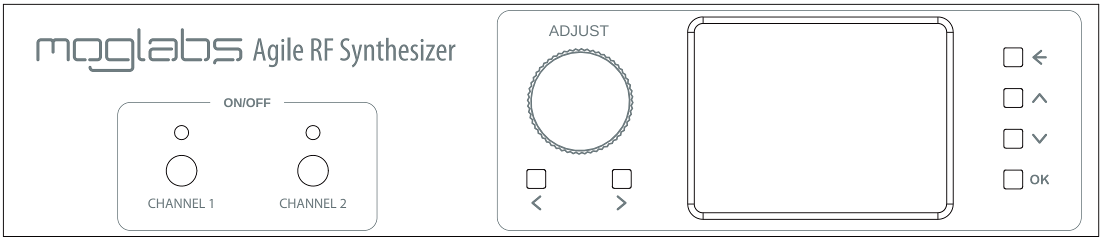


Figure 2.1: Front-panel layout of Rev9 XRF devices. Other revisions may have a different appearance.

The front-panel includes an interactive menu system for controlling the device. The buttons on the right-hand side of the display navigate through the menu structure, while the encoder wheel is used to edit values. The $\wedge$ and $\vee$ keys change between menu items, $\leftarrow$ exits to the previous menu, and OK enters the selected menu or activates the selected command.

The main menu (Figure 2.2) shows the current mode and status of each channel. In basic (NSB) mode, the current frequency and power of each channel is displayed, as well as whether modulation is currently enabled. Pressing the OK button with a channel selected will open the sub-menu to adjust settings for that channel (Figure 2.3).

The color of each menu item represents its purpose, as listed below.

White Static value, displayed for diagnostic purposes.

Yellow Adjustable value, modified using the encoder wheel.

Orange Currently selected channel.

Blue Submenu, entered with the OK button.

Green Command, executed by the OK button.

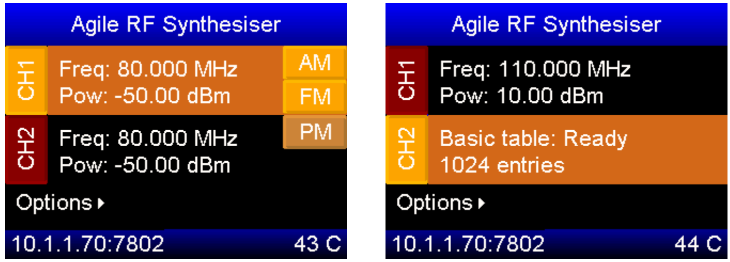


Figure 2.2: The main menu shows the current state of each channel. Left: both channels are in basic mode, with AM and FM enabled on CH1, and PM enabled on CH2. Right: CH2 is in basic table mode, with the number of entries in the table shown.

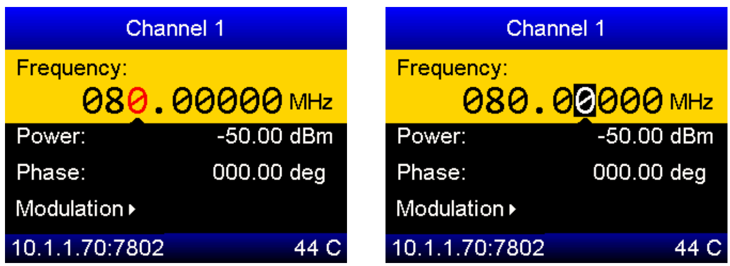


Figure 2.3: The basic parameters of each channel can be edited directly. Turning the encoder wheel modifies the selected digit of the current value (left) as indicated by the arrow. Pressing the encoder wheel changes to *digit select mode* (right), allowing the selected digit to be changed by turning the encoder.

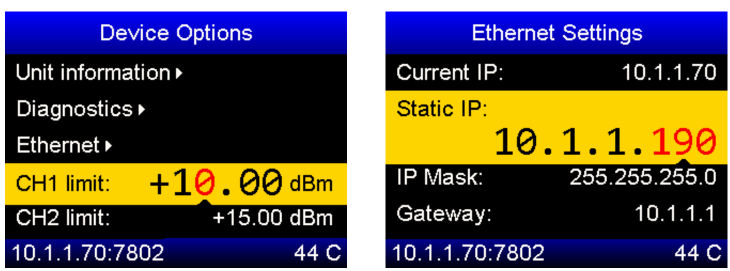


Figure 2.4: The options menu allows configuration of various settings, such as the maximum output power (left) and ethernet options (right).

When an editable (yellow) value is selected, turning the encoder wheel changes the value of the selected digit as identified by the arrow and red text. To change the digit of interest, either use the $<$ or $>$ buttons or press the encoder wheel to change to *digit selection mode*. In this mode, the currently selected digit is shown on a black background, and is changed by turning the encoder wheel. Pressing the encoder again returns to *value modification mode*.

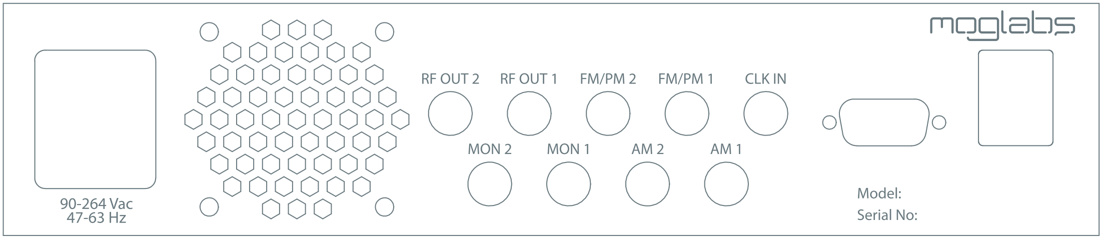
The options menu (Figure 2.4) allows the device configuration to be adjusted. In particular, the power limit applied to each channel should be set as per the desired application before use (see also the `LIM` command). The ethernet settings of the device can also be set using this interface, including whether to enable DHCP and the fall-back static IP address of the device. When in use, the network status is displayed on the display footer, and once connected displays the current IP address. Note that the *Restart ethernet* command must be used before changes in the ethernet menu will take effect.

Each channel of the device can be turned on or off using the push-buttons on the left of the front-panel. Each channel also has an associated multi-colour status LED indicator whose colour shows the current output state of the channel as follows.

Colour	DDS signal	Amplifiers
Off	✗	✗
Green	✓	✓
Yellow	✓	✗
Blue	✗	✓
Purple	Debug mode	
Red	Error state	

The overall brightness of the display can be set with the *contrast* value in the Options menu. The display also includes a *sleep* timer that dims the display if it hasn't received input in a given period of time. This feature can be disabled by setting the sleep timer value to 0.

2.2 Rear panel controls and connections



IEC power in The XRF is compatible with all standard AC power systems, from 90 to 264 V and 47 to 63 Hz.

Fan The XRF has three temperature-controlled fans directing air flow over the RF power amplifiers and the FPGA, exhausting through the rear vent. Ensure that the vent does not become blocked.

RF OUT SMA connectors for the primary RF outputs. Should be connected to a 50Ω load. Must not be short-circuited.

MON SMA connectors for *monitor* RF outputs, which are −27 dBc copies of the main output (when terminated into 50 Ω).

FM/PM/AM SMA analog inputs, nominally for frequency, phase and amplitude modulation (see chapter 5). These inputs can also be used for laser noise-eater or frequency stabilisation applications (see chapter 6).

CLK IN The XRF can be synchronised to a high-performance external clock input via this SMA connector and software commands (see §C.7). The input is 50 Ω terminated, and the provided reference should be between +3 dBm and +10 dBm, and preferably square-wave.

DE15 The DE15 connector provides basic I/O functionality (§7.1). There are TTL inputs for quickly suppressing the RF output, and TTL outputs for controlling experimental devices such as shutters. The XSMA breakout board is available to provide convenient SMA connectors for each I/O channel (§7.3).

RJ45/USB-A Ethernet (TCP/IP 10/100 Mb/s) and USB-HS type-A communications jacks.

2.3 Internal DIP switches

Four DIP switches are provided to assist in diagnosis and recovery of the XRF units. They should be left in default configuration for regular operation.

WARNING There is potential for exposure to high voltages inside the XRF. Take care around the power supply and ensure that objects, particularly electrically conducting objects, do not enter the unit.

CAUTION The cover should be replaced before powering on to ensure proper airflow and cooling.

	OFF	ON
1	Normal operation	Firmware update mode
2	Disable FPGA	Normal operation
3	Use factory settings	Normal operation
4	Normal operation	Factory reset

DIP 1 Default OFF. If switched ON, the unit will start in firmware upload mode (see §B.1).

DIP 2 Default ON. Switch OFF to disable the FPGA for diagnostic purposes.

DIP 3 Default ON. Switch OFF to use default device and network settings.

DIP 4 Default OFF. Switch ON and reboot to restore the unit to factory version and configuration.

3. Communications

The XRF can be connected to a computer by USB or ethernet (TCPIP). The software package `mogrf` (chapter 4) provides interactive functionality, or communications can be integrated into existing control software. Examples of controlling the XRF in several languages are provided in Appendix D.

3.1 Protocol

Communication follows a query/response protocol, where the user sends an ASCII string to the unit, and the unit sends an ASCII response back. The list of possible commands is detailed in Appendix C. All messages are CRLF-terminated, requiring that any communications must end with a carriage return ('\r' = ASCII 0x0D) and new-line ('\n' = ASCII 0x0A). Most terminal applications and drivers provide the ability to automatically append these characters when configured appropriately.

Statements are either *commands* or *queries*. A command is a statement that causes some action to occur, and the unit will respond with either `OK` or `ERR` depending on whether the command succeeded or not. For example,

```
> FREQ,1,80MHz
< OK: CH1 freq now 80.00000009 MHz (0x147AE148)
> FREQ,1,10MHz
< ERR: Frequency 10.00 MHz out of range
```

The response describes the outcome of the command, such as the achieved frequency taking into account discretisation by the DDS.

Queries are statements that return a value, which respond with the value in physical units first where applicable, or an error message beginning with "`ERR`". For example,

```
> FREQ,1
< 80.00000009 MHz (0x147AE148)
> FREQ,3
< ERR: Invalid channel, 3
```

In the above example, the frequency query provides a value first in MHz as well as the internal DDS setting (called the *frequency tuning word*) as hexadecimal in brackets.

It is strongly recommended that all software should wait for this response and check whether it indicates an error before continuing. The `python` and `LabVIEW` bindings provided by MOGLabs take care of buffering and error checking automatically.

The `mogcmd` application, which is available from the MOGLabs website as a standalone application or as part of the `mogrf` package, provides a convenient interface for sending commands and receiving responses (Figure 3.1).

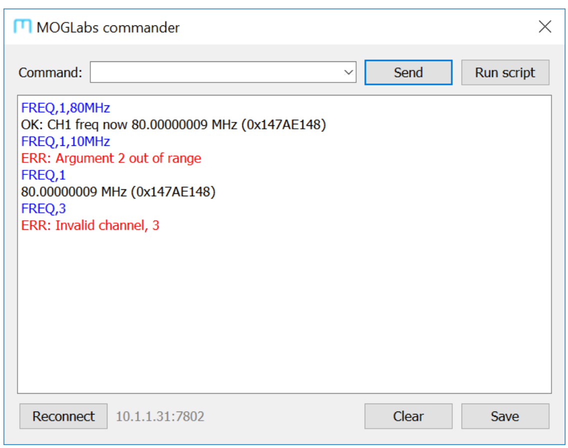


Figure 3.1: The `mogcmd` application, showing successful and unsuccessful commands and queries.

3.2 TCP/IP

The XRF can be accessed over ethernet via the IPv4 protocol. When ethernet is connected, the XRF will attempt to obtain an IP address by DHCP. If DHCP fails, an internally defined address will be used. In both cases, the address will be shown on the device display (for example, `10.1.1.190:7802`), showing the address and port number for communicating with the device.

3.2.1 Changing IP address

Depending on your network settings, you may need to manually set the IP address. This is most easily done via the front-panel menu system as described below. Once configured, these settings are stored in the non-volatile memory of the unit and will be recalled in future. However, automatic address acquisition via DHCP is a simpler solution where available.

1. From the main menu, open Options > Ethernet Settings.
2. Select *Static IP* and use the encoder wheel to set the IP address of the device as required. Note that pressing the encoder wheel changes between octets of the address.
3. Select *Gateway* and set the gateway address as required.
4. Select *DHCP* and set to OFF by turning the encoder anti-clockwise.
5. Select *Restart ethernet* and press the OK button.
6. The new IP address will be displayed in the display footer.

In some situations it may be necessary to power-cycle the device to propagate ethernet changes.

3.3 USB

The XRF can be directly connected to a host computer using a USB cable (type A-male). The device will appear as a Virtual COM port – a fast serial port that behaves like an RS232 connection.

The required STM32 Virtual COM Port Driver (VCP) device driver is available from the MOGLabs website for the Windows™ operating system. After installation, the XRF will appear as a new COM port on the machine.

To determine the port number of the device, go to Device Manager (Start, then type `Device Manager` into the Search box). You should see a list of devices including *Ports* (Figure 3.2).

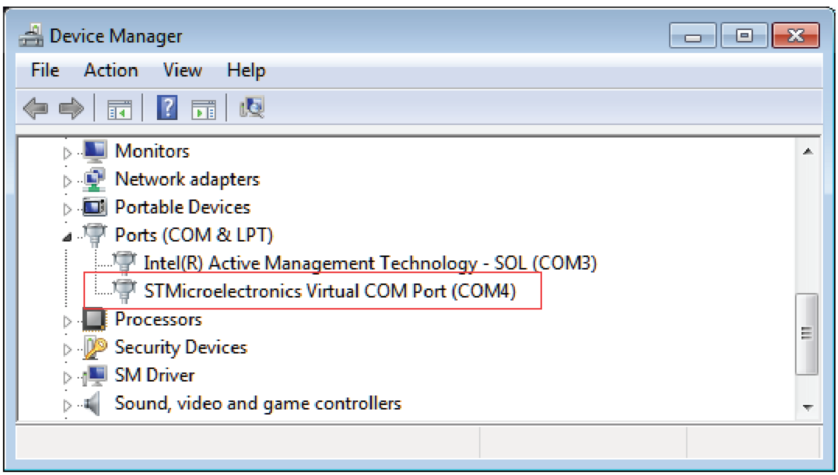


Figure 3.2: Screenshot of Device Manager, showing that the XRF can be communicated with using COM4. The port number might change when plugging into a different USB port, or after applying a firmware update.

The device can be identified as a COM port with the following name,

`STMicroelectronics Virtual COM Port (COMxx)`

where `xx` is a number (typically between 4 and 15). In the example above, the device was installed as COM4.

Note that if the port appears in Device Manager with a different name, then the driver was not successfully installed. If this occurs, disconnect the device from the host computer, reinstall the VCP driver, then reconnect the USB cable.

The `mogrf` host software (§4) automatically enumerates the available COM ports when started, making device identification simpler.

4. MOGRF host software

The mogrf software package provides a simple user interface to the basic behaviour of XRF devices, with the ability to issue commands, run scripts, control tables, and apply firmware updates.

Please note: It may be necessary to install a firmware update (see Appendix B) to use the software described in this section.

4.1 Device discovery

Upon starting the application, a device discoverer (Figure 4.1) is initiated. This program scans the COM ports of the host computer looking for an XRF device, and then scans the local network subnet. Starting the application is then as simple as selecting the device and clicking *Connect*. If your device is not listed, recheck your connection and network settings.

If the network and/or firewall blocks device discovery, enter the IP address of the unit in the *Device address* box directly.

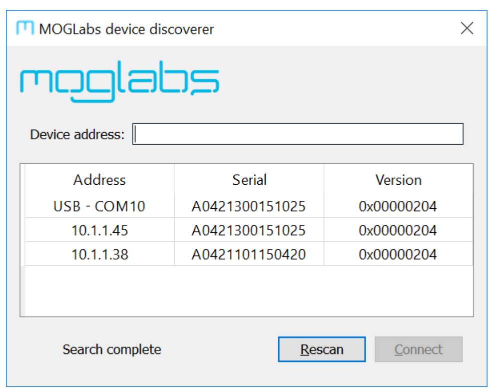


Figure 4.1: Example of the *Device discoverer* window, showing that one USB device and two networked devices were detected.

4.2 Device commander

The *Device commander* is an interactive terminal for issuing commands and queries to your XRF device and displaying the result (Figure 4.2). The accepted commands and their functions are listed in Appendix C. Type statements into the *Command* box and execute them by pressing the ENTER key or clicking Send. The window contains a history of recently executed commands.

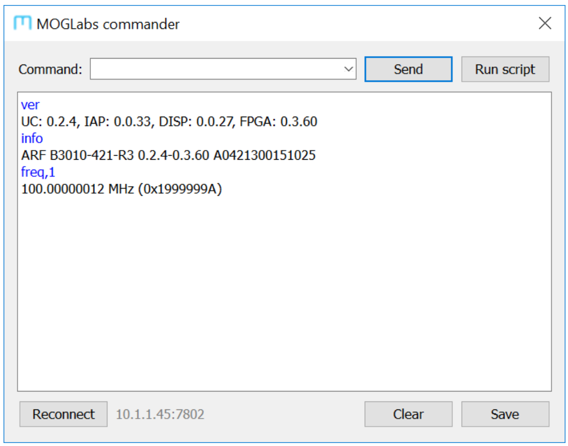


Figure 4.2: The *Device commander* window, which permits the execution of individual instructions or of text files containing scripts.

Scripts are ASCII text files where each line corresponds to a command to be executed (see Appendix D). Clicking *Run script* triggers stepwise execution of such a script, where the success of each statement is checked before executing the subsequent line. If an error occurs, execution of the script is aborted and an error message is displayed.

If the device is restarted or the connection is lost, clicking *Reconnect* will attempt to reestablish communication.

4.3 MOGRF main window

The main window of `mogrf` is shown below. The two channels are displayed side-by-side, with information and controls that depend on the current operational mode of each channel.

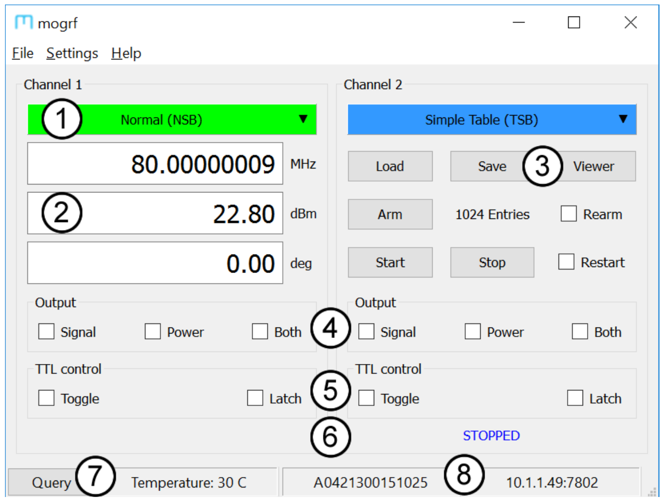


Figure 4.3: The main window of `mogrf`, showing Channel 1 in normal (NSB) mode and Channel 2 in simple table (TSB) mode.

The main features of the application are as follows:

1. Current operational mode of the channel. Click to change mode by selecting from a list. Note that *advanced table mode* will only appear on XRF units.
2. Current frequency, amplitude and phase in NSB mode. Changing the value immediately updates the output.
3. Controls are provided for specific table-mode functionality. Tables can be exchanged with internal FLASH memory, or uploaded/downloaded from the host machine in binary or CSV format (§8.5). Table execution can be started or stopped, auto-restart configured (§8.6), and a graphical viewer is provided for table visualisation in TSB mode (§4.4).

4. Channel output can be controlled by enabling only the RF switch (signal), RF amplifiers (power) or both (421-series only).
5. Options to enable external TTL control of the channel output using the OFF input on the DE15 connector (see §7.6).
6. Current channel status. Includes whether any modulation options are enabled (both frequency and amplitude modulation are enabled in this example) and the current execution status in table mode.
7. Click *Query* to manually update the displayed status information. Useful for reflecting changes caused by device commands or front-panel input.
8. The status bar contains diagnostic information about the unit and connection.

4.3.1 File menu

Device command Starts the *Device commander* (§4.2) for interactive execution of instructions to control the device.

Upload firmware Starts the firmware update application to upload and install updates on the device. The procedure for applying firmware updates is described in detail in Appendix B. It is strongly recommended to make a backup of device settings (Settings→ Download settings) before commencing an update.

Upload table Upload a previously downloaded binary table to FLASH memory, which can be subsequently loaded into either channel. Note that binary compatibility between firmware revisions is not guaranteed, and it is recommended that all tables be generated and stored in ASCII (human-readable) form.

4.3.2 Settings menu

Ethernet Allows configuration of network connection settings (IP address, mask, gateway and port). Particularly useful for configuring the network settings over USB. Note that changing the *Static IP* only has an effect if DHCP is disabled, or if DHCP name resolution fails.

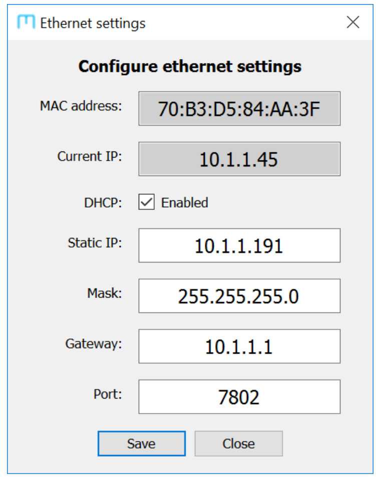


Figure 4.4: Ethernet configuration interface.

Note that changing the ethernet settings will require the application to be restarted, and may also require the device to be rebooted. The port should be unchanged at 7802 to ensure that the `mogrf` suite of programs can continue to communicate with the device.

Synchronisation Configures the channel synchronisation feature, detailed in §8.8.

Download settings Downloads configuration and calibration data from the device and stores it in a file for backup purposes.

Upload settings Restore previously downloaded settings to the unit.

4.3.3 Modulation

The XRF supports a wide variety of modulation options (see chapter 5) which can be controlled using this window. Individual modulation types can be enabled/disabled using the checkboxes on the left, and their associated gains adjusted using the sliders (Figure 4.5).

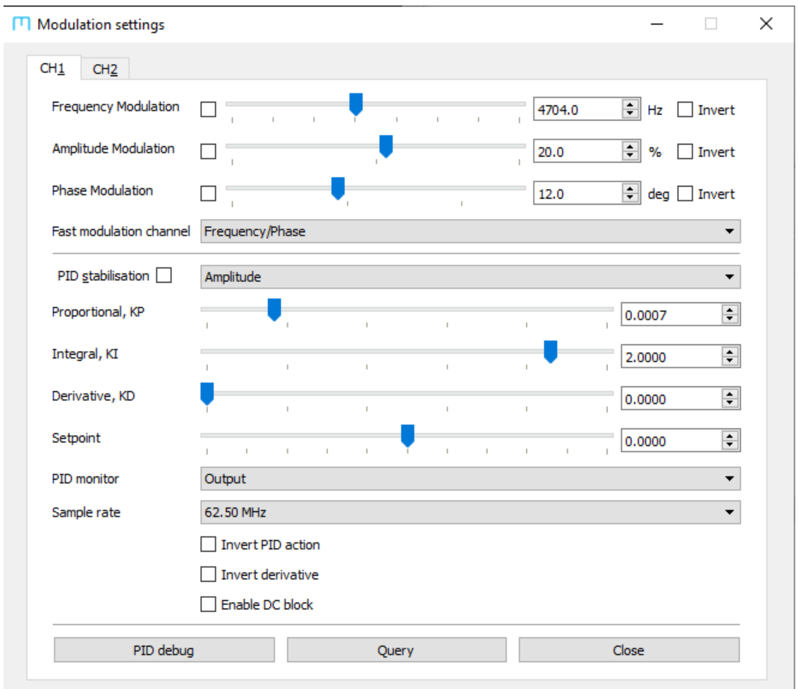


Figure 4.5: Use the Modulation dialog in the `mogrf` application to change the modulation settings for each channel

4.3.4 PID debug

The `mogrf` software contains a diagnostic capture viewer that displays the error signal input and servo output signals from the PID controller, which can be beneficial in checking the operation of the PID loop and optimising the gain coefficients.

In particular, it is recommended for performing the following checks:

- Adjust the offset voltage to give zero input at the lock point.
- Confirm the PID action polarity by checking the error signal converges *towards zero* when the controller is engaged.
- Adjust the VGA gain to increase noise suppression ADC.
- Detect oscillation when the gain is too high.

The indicator LED shows green when the PID is locked and red when the controller has saturated. Clicking this indicator toggles whether the PID is engaged. Clicking the *Single* button momentarily disables and re-enables the PID, allowing the initial transient to be observed.

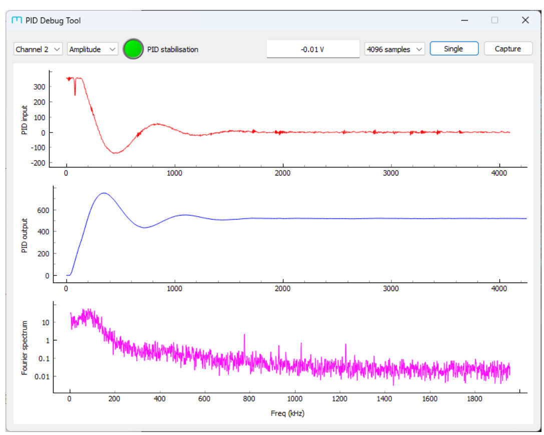


Figure 4.6: Demonstration of the PID debug capture when engaging the lock showing the error signal rapidly converging to zero. The overshoot implies the gain is slightly too high.

4.3.5 Help

Diagnostics Queries the unit for diagnostic information, which may be useful in assessing issues with the functionality of the XRF (Figure 4.7). When encountering a problem with the device, please run the diagnostics and click *save results*, then send the resulting text file and a description of the problem to MOGLabs for analysis.

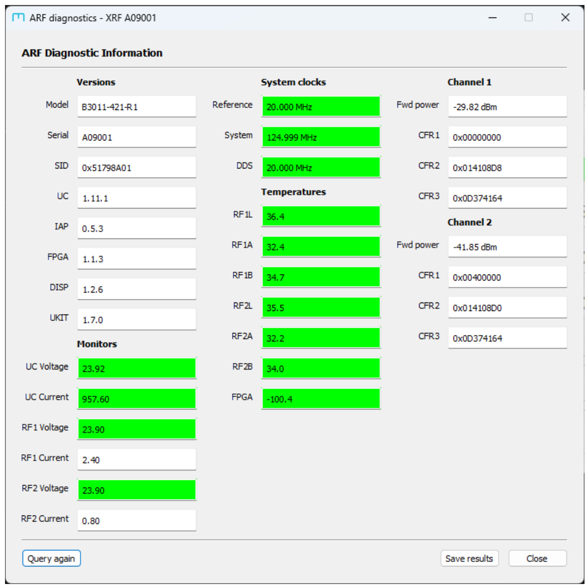


Figure 4.7: Diagnostic information about the connected XRF unit, which should be sent to MOGLabs for analysis if there is a problem with the device.

About Displays version information about the `mogrf` toolkit and connected XRF device, for support purposes.

4.4 Table viewer

In simple table mode (TSB mode), `mogrf` provides a viewer for inspecting both the table instructions currently loaded into each channel, as well as the instructions stored in FLASH memory (Figure 4.8). This is beneficial for cataloguing the sequences in memory, as well as for debugging sequences which have been generated by scripts and uploaded to the device.

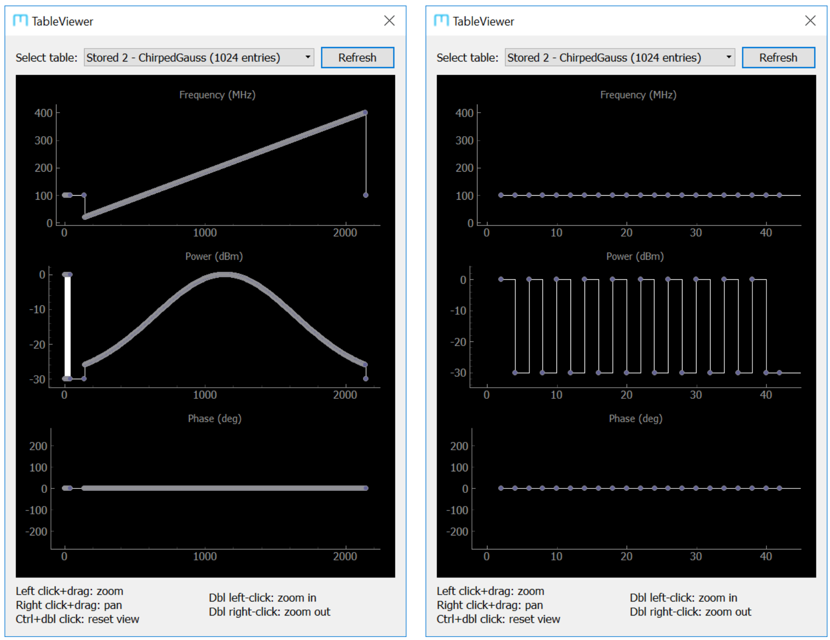


Figure 4.8: *Table viewer* showing how the frequency, power and phase of a table stored in FLASH memory change across the sequence (left). The example shown is a chirped Gaussian pulse, with a number of rapid on/off pulses at the beginning. Mouse controls allow zooming in on areas of interest, such as the rapid pulses at the start of the sequence (right).

At present, the table viewer is only available for simple tables, but may be extended in future to provide visualisation of advanced tables.

4.5 External I/O settings

The XRF provides extensive digital I/O capability through the `EXTIO` command and configuration window (Figure 4.9). It displays the instantaneous input and output state of the I/O pins on both the DE15 connector (§7.1) and the high-speed banks (§7.2).

This is provided to assist diagnosing the I/O state, and to ensure that any settings are correct for the desired application. In particular, note that associated pins must be set to AUTO control to be used in combination with table mode.

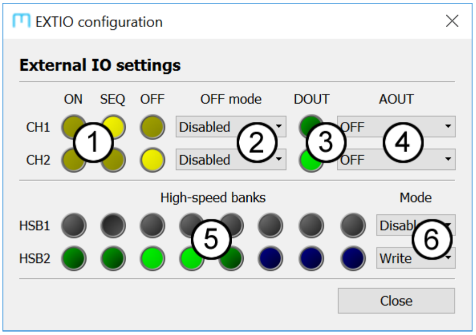


Figure 4.9: External I/O configuration window, showing the current state of inputs (yellow), outputs (green), table-mode outputs (blue) and disabled outputs (black). Left-clicking on an output changes its state, and right-clicking brings up a menu of options.

Features of the EXTIO configuration window are:

1. Current state of the input pins of the DE15 connector.

2. Configure the CHx-OFF input on the DE15 connector as an interlock (*Latch* mode) or for direct control of the RF switch (*Toggle* mode), see §7.6.

3. Current state of the DE15 digital output (shutter) pin, which can be set manually or placed in AUTO mode to use in table mode.

4. Analog monitoring signal currently output on the analog output pin of the DE15 connector (not supported on all units).

5. Current state of the two high-speed banks. The banks are disabled (black) on boot, and must be set to either read (yellow) or write (green) mode on a per-bank level. Once in write mode, individual output pins can be set as AUTO (blue) for use in table mode.

6. Overall I/O mode to apply to the associated high-speed digital bank.

5. External modulation

The XRF supports external modulation of the RF in NSB mode through the modulation input SMA connectors on the back-panel. Frequency, amplitude and phase modulation of the RF are supported, and dual-modulation is possible for simultaneous FM/AM or FM/PM.

WARNING: The modulation inputs are nominally ±4V, and can be permanently damaged by applying higher voltages. Ensure that modulation is disabled when disconnecting the back-panel SMA connectors, as floating inputs can cause unexpected results. Input termination is recommended when not in use.

5.1 Operational principle

Modulation is performed by digitising the analogue input signal, which is then multiplied by the modulation gain and added to the internal control value associated with the particular modulation mode (*frequency tuning word* for frequency, "*amplitude scale factor*" for power or *phase offset word* for phase). Limits are applied to the value to ensure that the power is always limited to the value set with the `LIMIT` command.

The ADC operates at 62.5 MS/s with 12-bit resolution (±4 V range), anti-aliased with 7^{th}-order filters for a measured 3-dB bandwidth of 10 MHz. Simultaneous modulation of two parameters is possible, although one mode will have a reduced bandwidth (see §5.3).

Modulation is enabled/disabled with the `MDN` command. For example, to enable AM on channel 1, use the command `MDN,1,AMPL,ON`. Modulation is not available in table mode.

5.2 Modulation gain

The modulation depth is controlled by the *gain* and is set using the `GAIN` command. Each modulation mode (amplitude, frequency or phase) has a separate modulation gain for individual control, and can be negative to indicate that the modulation action is inverted. The gain can be specified either with physical units, or using an integer representation.

5.2.1 Physical units

The recommended way to specify modulation gain is using physical units, which corresponds to the modulation achieved at +1V input. Frequency can be specified in MHz, kHz or Hz, phase in degrees or radians, and amplitude in percentage.

For example, when +1V is applied to the associated modulation input, `GAIN,1,FREQ,10MHz` will shift the output frequency by 10 MHz and `GAIN,1,AMPL,-100%` will bring the amplitude to zero.

Note changing the output power changes the amplitude and hence the modulation depth as a percentage, so the amplitude gain command must be issued again to keep the percentage depth the same.

Also, when performing frequency modulation, the gain should be chosen so that the output frequency stays between 20 MHz and 400 MHz to prevent unexpected behaviour.

5.2.2 Hexadecimal representation

The gain can also be specified as a signed 32-bit integer in hexadecimal, with a negative value indicating that the modulation action is inverted[1]. The range of gain values is shown in the table below.

[1] Negative values are internally represented using two's complement.

	FM	AM	PM
Max gain	0x3FFF8000	0x3FFF	0xFFFF
Step size	0.23 Hz	0.006% Max	0.0055°
Max modulation	250 MHz	100% Max	360°

Table 5.1: Gain ranges for different modulation modes at full scale (4 V).

Taking these values, correcting for the full-scale voltage and ignoring discretisation and saturation, an applied voltage will have the following effective modulation:

$$\text{Frequency:} \quad df = (0.23\ \text{Hz/V})(G_f V_{\text{in}}/4)$$

$$\text{Phase:} \quad d\phi = (0.0055°/\text{V})(G_\phi V_{\text{in}}/4)$$

$$\text{Amplitude (*):} \quad dA \approx \begin{cases} (3.5\ \text{V/V}) G_a V_{\text{in}} & \text{for 421-models} \\ (0.65\ \text{V/V}) G_a V_{\text{in}} & \text{for 021-models} \end{cases}$$

(*): This expression assumes a 50 Ω load, and depends on the individual unit power calibration. The actual output amplitude respects the maximum power limit set by the `LIMIT` command. The available modulation depth depends on the difference between the current output power (`POW`) and the predefined limit (`LIMIT`).

Similarly the gain required to achieve a desired modulation depth at 1 V input can be estimated as:

$$G_f = \frac{1073709056}{62.5\ \text{MHz}} df, \qquad G_\phi = \frac{65536}{90°} d\phi, \quad \text{and} \quad G_A = 4A_0 M_A,$$

where M_A is the amplitude modulation depth and A_0 is the initial amplitude (returned by the `POW` command as a hexadecimal number).

Examples: when using a ±1 V modulation input,

- To change the frequency by ±1 MHz, set the gain to
  $1073709056 \times (1/62.5) = 17179345.$
- To modulate the phase by ±45°, set the gain to
  $65536 \times (45/90) = 32768.$

- To amplitude modulate by ±25% in an XRF at an average RF power of +30 dBm, use the POW command to determine that the amplitude is 0x2000 = 8192, and set the gain to

  $$4 \times 8192 \times 0.25 = 8192.$$

5.3 Dual modulation: fast and slow modes

The XRF is capable of dual modulation, where the RF is either simultaneously FM and AM or PM and AM modulated. However, due to DDS interface limitations, only one parameter can be modulated at full speed using the parallel bus. The other parameter is modulated on the serial bus at 1 MHz.

The FMSPEED command allows selection of which modulation parameter has higher bandwidth as shown in the table below.

Command	FM/PM bandwidth	AM bandwidth
FMSPEED,1,FAST	10 MHz	1 MHz
FMSPEED,1,SLOW	1 MHz	10 MHz

Table 5.2: Effect of using FMSPEED to control modulation on Channel 1.

The signal processing chain causes a propagation delay between the modulation input and the RF output of approximately 500 ns in fast (parallel) mode, and $< 3\ \mu s$ in slow (serial) mode. Furthermore, inducing a step change with slow modulation (e.g. using AM to switch the output) may appear to have jitter of up to 500 ns depending on the delay between the change in the modulation input and the subsequent DDS update.

Simultaneous FM and PM is not currently supported, as phase modulation shares the FM/PM modulation input.

5.4 Examples

5.4.1 Simple linear ramps

Listing 5.1 shows simultaneous linear ramping of amplitude and frequency (Figure 5.1). A linear ramp is connected to both the FM/PM1 and AM1 modulation inputs in parallel. Subsequently increasing the AM gain results in clamping the amplitude to respect the limit set by the `LIMIT` command (Figure 5.2).

```
# configure the channel
FREQ,1,60MHz
POW,1,0dBm
LIM,1,10dBm
# connect linear ramp to FREQ1 and AMP1 mod in
MDN,1,AMPL,ON,4000
MDN,1,FREQ,ON,30000
FMSPEED,1,SLOW
```

Listing 5.1: Simultaneous AM and FM

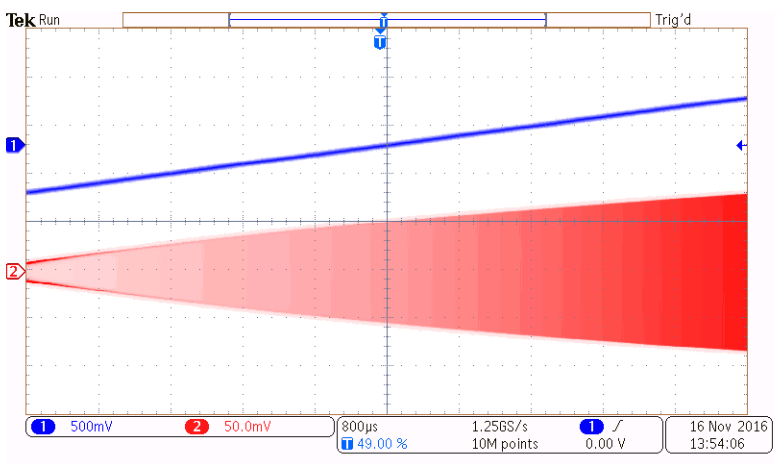


Figure 5.1: Demonstration of simultaneous AM/FM modulated RF (red) when driven by a linear ramp (blue).

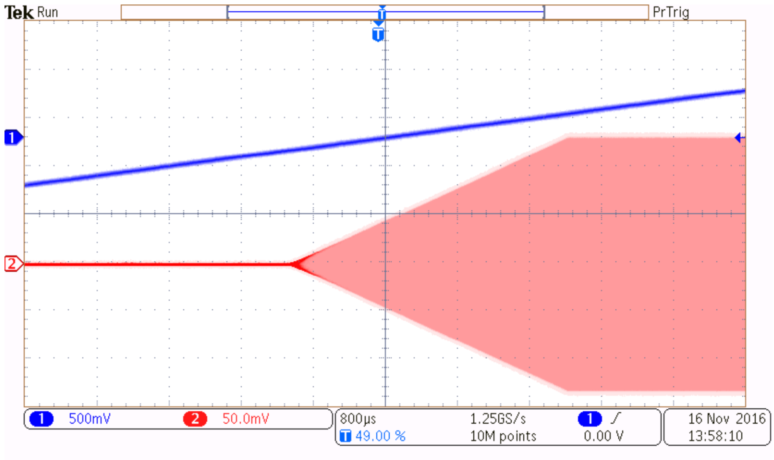


Figure 5.2: Demonstration of high gain amplitude modulation showing clipping at zero and the power limit set by the `LIMIT` command.

5.4.2 *Comparison of fast and slow modulation*

Figure 5.3 compares the result of different `FMSPEED` settings when applying a 1 V_{pp} sine wave at 100 kHz to the AM-input when both AM and FM are enabled simultaneously.

In this scenario, only one parameter can be applied to the fast parallel bus for maximum modulation rate. When `FMSPEED` is set to FAST, the amplitude is modulated via the slow serial interface, and the envelope displays stepwise discretisation.

Alternatively, when `FMSPEED` is SLOW, amplitude modulation uses the fast parallel interface and the resulting envelope is smoother at the expense of stepwise changes in frequency.

Although smooth changes in all parameters is desirable, depending on the application stepwise changes in one parameter may be acceptable provided smooth changes can be achieved in the other.

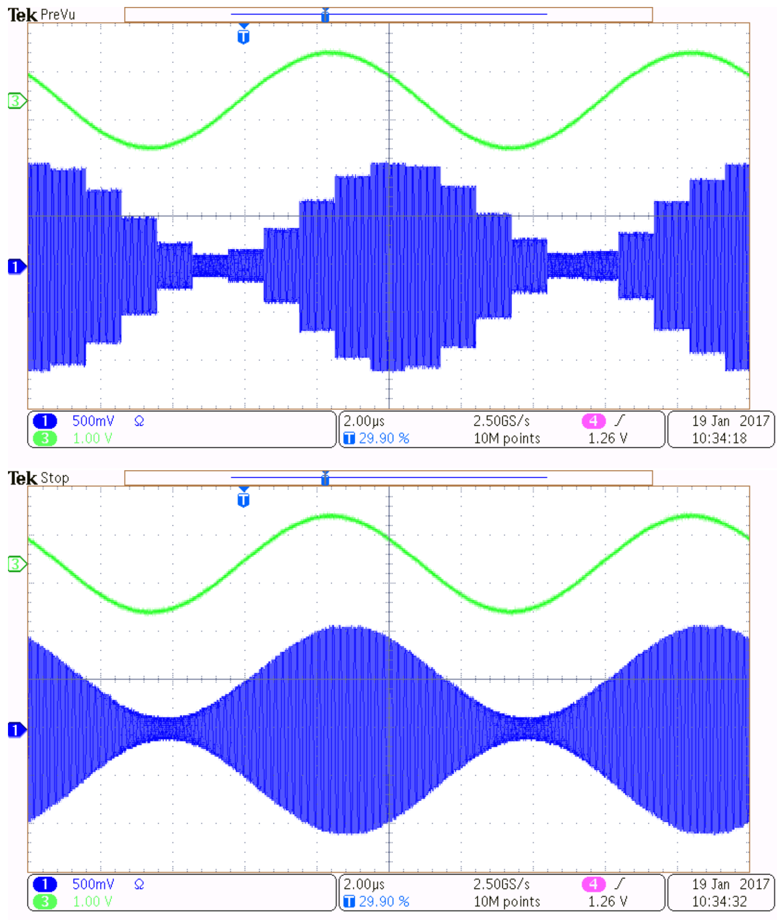


Figure 5.3: Comparison of the output waveform (blue) with `FMSPEED,FAST` (top) and `FMSPEED,SLOW` (bottom) when performing amplitude modulation with a 100 kHz sine-wave input (green).

5.4.3 Phase modulation

The two channels of the XRF can be used to perform phase modulation experiments whereby CH2 is used to demodulate CH1. Using phase modulation mode with a 1 Vpp modulation input with gain `0x7fff` gives $\pm\pi$ phase modulation (Figure 5.4).

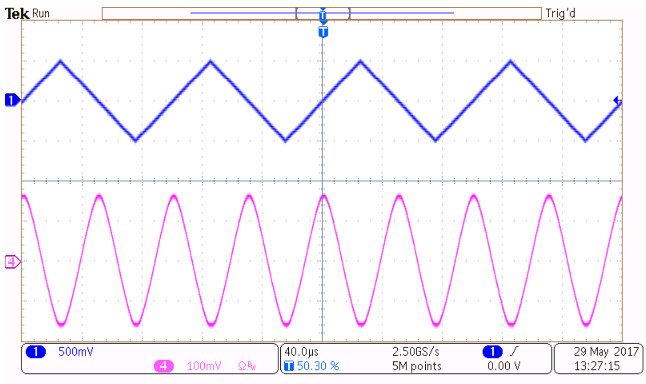


Figure 5.4: A 1 Vpp, 10 kHz triangle wave (blue) is used for phase modulation of CH1 and demodulated with an unmodulated CH2 at the same frequency (magenta).

Note that it may be necessary to adjust the phase of one of the channels using the PHAS command to compensate for the (frequency-dependent) differential phase delay in the signal paths and give a demodulated signal centred at zero. Similarly the SYNC feature should be enabled to ensure the two channels are phase-locked to each other.

6. PID stabilisation

In addition to external modulation, the XRF in NSB mode also implements PID control loops which can be used in conjunction with an AOM to perform intensity or frequency stabilisation of a laser. Each channel has an independent PID controller, which acts to drive an "error signal" provided to the modulation input towards zero.

Engaging the PID controller in AM mode adjusts the instantaneous RF output power, which could be used to compensate for the amplifier frequency response when performing wide-band frequency ramps, or to reduce the technical noise of a laser beam propagating through the AOM ("noise eating").

A standalone application note describing intensity stabilisation in more detail, including instructions for optimisation, is available from the MOGLabs website at `http://moglabs.com/appnotes`

6.1 Signal conditioning

Starting with serial numbers A09xxx, the XRF contains integrated signal conditioning that enables a DC-coupled photodiode signal to be used directly for locking the PID servo, removing the requirement for external signal conditioning circuits. This includes configurable analog offset (setpoint subtraction) and a variable gain amplifier (VGA) to improve the dynamic range of the input signal. The error signal $e(t)$ processed by the PID is simply

$$e(t) = G(V_{\text{in}} - V_{\text{sp}}),$$

where the offset V_{sp} is set in volts using the `PID,OFFSET` command (± 3.5 V range), and the gain G in decibels[1] using the `PID,VGA` command (-8 to $+24$ dB).

[1] Note that when the VGA is enabled (non-unity gain), the error signal will saturate at a lower voltage than when the VGA is disabled (i.e. set to 0 dB gain).

6.2 PID control loop

The XRF implements the feedback control via a standard PID (proportional integral differential) function:

$$u(t) = Gk_p e(t) + Gk_i \int_0^t e(\tau) d\tau + Gk_d \frac{de}{dt},$$

where $e(t)$ is the input error signal, $u(t)$ is the feedback response, and G is the overall modulation gain. The gain constants k_p, k_i, k_d are floating-point values in the range $[0, 1)$ which correspond to proportional, integral and differential terms respectively. Typical values are $k_p = 0.03 - 0.8$, $k_i = 0.01 - 0.15$ and $k_d = 0$.

When optimising a PID control loop, it should be kept in mind that the achievable loop bandwidth is limited by the **propagation delay** of the signal processing chain, which is distinct from the modulation bandwidth. This includes the impulse response of the AOM itself, as well as the photodetector, signal-processing electronics, analog-to-digital converters, and the DDS.

6.3 Dual modulation with PID

It is possible to perform PID simultaneously with another form of modulation enabled. For example, PID can be used to compensate for the frequency response of RF components or the AOM when performing wide-band frequency modulation, as shown in Listing 6.1.

```
# enable FM on channel 1
MDN,1,FREQ,ON
GAIN,1,FREQ,0x3FFFF
# set PID gains and enable
PID,GAIN,1,P,0.1
PID,GAIN,1,I,0.01
PID,GAIN,1,D,0
PID,ENABLE,1,AMPL
# set FM to SLOW mode, allowing PID to be FAST
FMSPEED,1,SLOW
```

Listing 6.1: Simultaneous FM and PID intensity stabilisation example.

Limitations of the DDS interface mean that only one of PID and external modulation can be performed at full bandwidth (§5.3). Most applications will benefit from using PID in "fast" mode, but some applications such as compensating for thermal drift in AOM diffraction efficiency do not require high bandwidth and can be operated in "slow" mode.

6.4 Noise-eater implementation

A common application for PID controllers is optical noise eating, which suppresses technical noise arising from power fluctuations in a laser beam. Figure 6.1 shows a typical configuration, where the intensity of the undiffracted (zero-order) beam is stabilised as seen in Figure 6.2.

In this configuration, the AOM acts as a high-speed variable optical attenuator, diffracting some of the light into the unused first-order output. The transmitted optical power is measured with a photodetector, and the XRF controls the RF power in proportion to the measured optical power. If the measured power is too high, the RF power is increased and more light is diverted into the diffracted output. This allows fluctuations in intensity to be suppressed, at the expense of reducing the transmitted power slightly (typically 90% transmission is achieved).

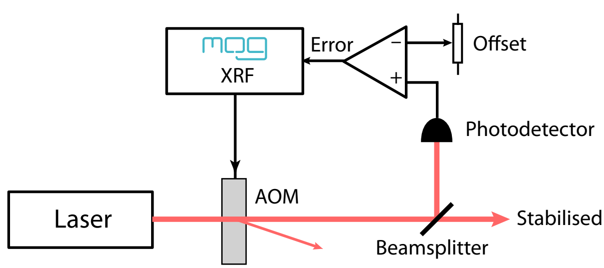


Figure 6.1: Typical setup for optical power noise eater.

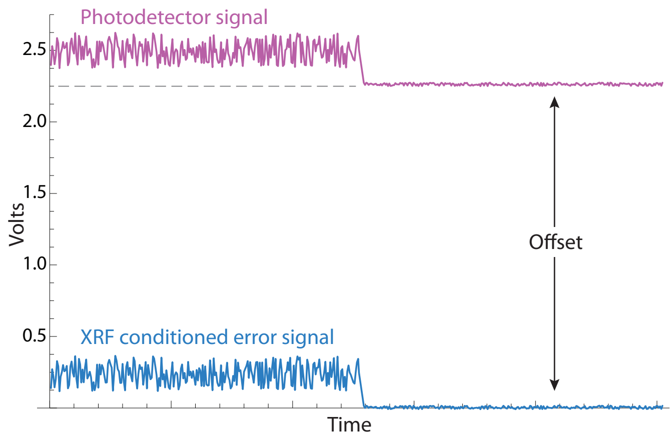


Figure 6.2: Photodetector signal (magneta) and conditioned error signal (blue) before and after activating noise-eater feedback (simulated).

6.5 Example

The following example configures PID on Channel 1 for a particular set of gain constants. Note that the **gain constants must be tuned for each individual implementation**. It is recommended to optimise the gain constants using the *Modulation settings* window in `mogrf` while monitoring the measured noise on a spectrum analyser (see §4.3.4).

```
# Setup channel 1 for PID noise—eater feedback
# set sample rate
PID,RATE,1,15.625
# set controller gains
PID,GAIN,1,P,0.5
PID,GAIN,1,I,0.05
PID,GAIN,1,D,0
# set overall gain (i.e. max controller output)
GAIN,1,AMPL,0x100
# set frequency and power
FREQ,1,80MHz
POW,1,20dBm
# activate PID in AMPLITUDE mode
PID,ENABLE,1,AMPL
```

7. Digital I/O

TTL digital inputs and outputs (0–5 V) are provided on the XRF through the DE15 connector on the rear panel, and the high-speed bus (HSB). The inputs can be used as triggers and the outputs can be controlled manually or using by table mode entries (§8.3).

Note: Digital inputs are pulled *high*, meaning that a disconnected input pin is equivalent to supplying a TTL high to that input.

7.1 DE15 connector

The DE15 connector on the rear panel (Figure 7.1) provides digital inputs and outputs for monitoring and synchronisation purposes.

CHx-DOUT `Pin 4 (Ch1), Pin 9 (Ch2)`
TTL outputs that can be controlled manually or from table mode, for example to activate a mechanical optical shutter or trigger another device. This output has a rise time of 3 us.

Note: Pin 9 is internally disconnected in some commercial DE15 cables. Please check continuity on pin 9 continuity when using CH2-DOUT.

CHx-SEQ `Pin 3 (Ch1), Pin 5 (Ch2)`
Pin used for hardware triggering in table mode. When the table is armed and a falling edge is received on the associated input, the table begins executing.

CHx-ON `Pin 2 (Ch1), Pin 6 (Ch2)`
Driving this pin LOW in NSB mode instructs the FPGA to switch **ON both the RF signal and amplifiers** (if present). Has no effect if the output is already enabled. For applications that require the amplifiers to stay powered on, the CHx-OFF pin should be used instead.

Pin	Signal	Type
1	CH1 OFF	TTL in
2	CH1 ON	TTL in
3	CH1 SEQ	TTL in
4	CH1 DOUT	TTL out
5	CH2 SEQ	TTL in
6	CH2 ON	TTL in
7	CH2 OFF	TTL in
8	GND	0 V
9	CH2 DOUT	TTL out
10	GND	0 V
11	N/A	
12	GND	0 V
13	N/A	
14	GND	0 V
15	GND	0 V

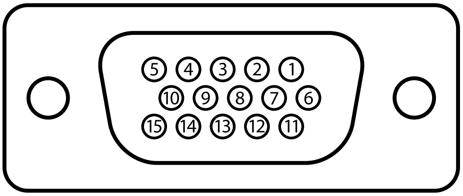


Figure 7.1: Pinout of high-density 15-pin female DE-style rear panel IO connector.

CHx-OFF `Pin 1 (Ch1), Pin 7 (Ch2)`

This input bypasses the FPGA and directly turns **the RF switch OFF unless a TTL LOW is provided**. Bypassing the FPGA provides an extremely fast method for generating externally-controlled pulses, as further discussed in §7.6.

This pin is **disabled by default** and must be enabled using the `EXTIO,ENABLE` command. If this feature is enabled and the pin is disconnected, **the RF output will not turn on**.

7.2 High-speed digital

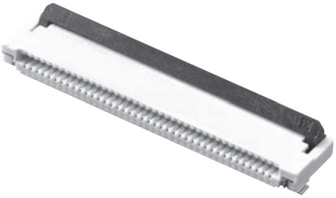
The FPGA also provides 16 high-speed digital I/O lines for use with table mode (Figure 7.2). Internal connector P1 accepts a 30-way, 0.50 mm pitch ribbon cable that can be inserted through a slot in the left-hand side of the case. The connector is an Omron XF2M-3015-1A, with example matching FFC ribbons Molex 0982660326 (150 mm length) or 0152660329 (200 mm length). Each line includes a 10 Ω series resistor and capability to sink and source 24 mA.

The high-speed lines can be configured as inputs or outputs using the `EXTIO,MODE` command, which configures groups of lines called *banks*. For example, to configure pins D1-D8 (bank 1) as inputs and pins D9-D16 (bank 2) as outputs, use the commands:

```
EXTIO,MODE,1,HSB,READ
EXTIO,MODE,2,HSB,WRITE
```

If finer control is required, the digital lines can also be configured as sub-banks of 4 lines. For example to set pins D1-D4 (sub-bank 1A) as inputs and pins D5-D8 (sub-bank 1B) as outputs, use the command:

```
EXTIO,MODE,1,HSB,READ,WRITE
```



Pin	Signal
1	3.3 V
2	3.3 V
3	3.3 V
4	GND
5	A0
6	A1
7	A2
8	A3
9	GND
10	GND

Pin	Signal
11	A4
12	A5
13	A6
14	A7
15	GND
16	GND
17	B0
18	B1
19	B2
20	B3

Pin	Signal
21	GND
22	GND
23	B4
24	B5
25	B6
26	B7
27	GND
28	GND
29	GND
30	GND

Figure 7.2: High-speed digital IO connector (internal). Note that the FFC cable can be inserted upside-down, reversing the pin ordering.

7.3 XSMA breakout board

The XSMA breakout board (Figure 7.3) is an optional additional component that provides SMA connectors for each of the digital I/O lines of both the DE15 connector (§7.1) and the high-speed bus (§7.2). The pins of the high-speed bus have matched track-lengths, to ensure consistent propagation delay for applications using advanced table mode.

The flat-flex cable (FFC) carrying the high-speed digital I/O signals can be inserted in either orientation, with contacts facing up or down. Each pin of the high-speed bus has two labels, corresponding to the purpose of the pin given the cable orientation.

If the orientation is the same within the XRF and the XSMA board, the second set of labels apply, whereas if the cable is crossed-over,

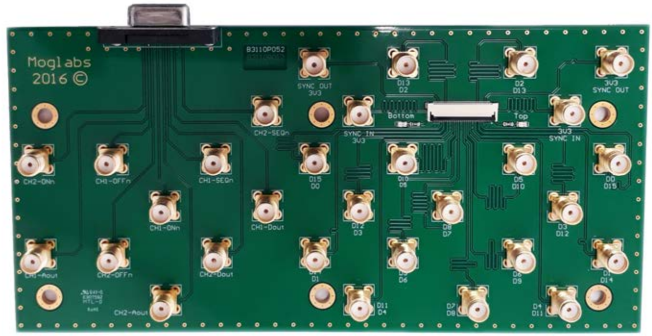


Figure 7.3: XSMA digital I/O breakout board, providing SMA connectors for the DE15 connector (left) and high-speed digital connector (right).

the first set apply. Second-generation breakout boards include LED indicators labelled *Top* and *Bottom* that identify which set of labels to use.

The board dimensions are 172x70mm (first generation) or 172x85mm (second generation).

7.4 Configuration

The `EXTIO` command is used to control the behaviour of digital I/O. Outputs can be set with `EXTIO,WRITE`, and queried with `EXTIO,READ` when set to MANUAL control, or commanded by table mode entries when set to AUTOMATIC control.

The table below shows the functionality available on the different pins. Pins in the high-speed bus can be addressed individually (HSn) or collectively as a whole bank (HSBANK) in case multiple outputs need to be changed simultaneously. HSB is short-hand for HSBANK.

Function	OFF	ON/SEQ	DOUT	HSB	HSn
Enable	✓	–	–	✓	–
Disable	✓	–	–	✓	–
Reset	✓	–	✓	✓	–
Mode	✓	–	–	✓	–
Control	–	–	✓	✓	✓
Write	–	–	✓	✓	✓
Read	✓	✓	✓	✓	✓
Counter	✓	–	–	–	✓

The different `EXTIO` commands are summarised below.

EXTIO,ENABLE `EXTIO,ENABLE,ch,pin`
Enable the functionality of the specified `pin` on the given channel `ch`. If `pin` is HSB, the entire bank of pins is enabled.

EXTIO,DISABLE `EXTIO,DISABLE,ch,pin`
Disable the functionality of the specified `pin`.

EXTIO,RESET `EXTIO,RESET,ch,pin`
Resets the functionality of the specified `pin` to its default state.

EXTIO,MODE `EXTIO,MODE,ch,pin,[mode]`
Change the mode of the specified `pin`. If `pin` is HSB, then `mode` is either READ or WRITE. If `pin` is OFF, then `mode` is either LATCH or TOGGLE. If `pin` is disabled, it is enabled first.

The sub-banks can be controlled using a second mode argument: the first `mode` argument applies to lines 1–4 of the bank, and the second applies to lines 5–8.

EXTIO,CONTROL `EXTIO,CONTROL,ch,pin,[mode]`
Sets the control of the specified `pin`. The parameter `mode` is either MAN[UAL] or AUTO[MATIC]. Pins must be set to AUTO mode to access them in table mode. If `pin` is HSB and the (sub)bank is not in write mode, the (sub)bank is changed to write mode first.

EXTIO,WRITE `EXTIO,WRITE,ch,pin,value`
Write the specified `value` to the output `pin`. If `pin` is HSB, then `value` is an 8-bit number, whose bits correspond to the values to set on the pins of that bank. Otherwise `value` can be one of ON, OFF, 1 or 0. If the pin is not set to MANUAL control, it is changed to manual control first.

EXTIO,READ `EXTIO,READ,ch,pin`
Reads the specified `pin` and returns its current state. If `pin` is HSB, then the returned value is an 8-bit hexadecimal number where each bit corresponds to the state of the corresponding line.

EXTIO,COUNTER `EXTIO,COUNTER,ch,pin,cmd`
The FPGA provides independent digital counters which can be activated on any pin configured as an input. The counters can be individually started, stopped or queried (see §7.7).

7.5 TTL switching

A versatile feature of the XRF is the ability to switch the RF in response to an external input such as a tactile switch or a TTL trigger for device synchronisation.

Each channels has two TTL inputs on the DE15 connector, labelled CHx-OFF and CHx-ON (see 7.1), that control whether the output is enabled or disabled as described below. Both of the inputs are active LOW and have no effect if pulled HIGH.

Each input has a debouncer circuit for use with tactile switches that can be enabled or disabled using the `DEBOUNCE` command. Note that activating the debouncer will introduce a small delay to changes in the input.

CHx-ON When the output is disabled, pulling this pin LOW will switch **both** the RF signal and amplifiers ON (NSB mode only). Has no effect when disconnected.

CHx-OFF When enabled (below), the RF signal will be disabled **unless** this pin is pulled LOW. If disconnected or pulled HIGH, no output is produced. Does not affect the RF amplifiers.

This CHx-OFF functionality disables the RF output **unless** the required input is provided, so it must be explicitly enabled using software commands. There are two modes of operation for this input as described below.

`EXTIO,MODE,1,OFF,TOGGLE`
: Sets CHx-OFF to TOGGLE mode: the RF is off whenever the input is HIGH and the output is enabled whenever the input is LOW. Can be used for generating rapid externally-controlled pulses.

`EXTIO,MODE,1,OFF,LATCH`
: Sets CHx-OFF to LATCH mode: if the OFF input goes HIGH, the output will be disabled and remain disabled. The output can then only be re-enabled by taking the input LOW and **then switching on the output** via software or the front-panel. This functionality can be used as part of an interlock system.

`EXTIO,ENABLE,1,OFF`
: Enable the CHx-OFF behaviour, as previously configured by the `EXTIO,MODE` command.

`EXTIO,DISABLE,1,OFF`
: Disable the CHx-OFF input, regular operation using the `ON` and `OFF` commands.

Please note that this setting *is* stored persistently and needs to be manually disabled when no longer desired.

7.6 Pulse generation

Note: Ensure that the debouncer is disabled using the DEBOUNCE command when generating pulses with TTL inputs.

There are several approaches to generating pulses using an XRF in NSB mode (normal operation): using the TTL inputs on the DE15 connector, or using amplitude modulation (AM). Alternatively, pulses can also be generated in a preprogrammed sequence using table mode (chapter 8).

Method	Transition	Time
RF switch	↓	25 ns
RF switch	↑	30 ns
RF amplifier	↓	2 s
RF amplifier	↑	2 s
AM (fast)		500 ns
AM (slow)		<3 us
DE15-ON		2 ms
DE15-OFF		40 ns

Table 7.1: Typical on/off time delays for switching hardware components, and for different methods of pulse generation (debouncer disabled). The time given for amplifier transitions includes time for the output to stabilise, which may vary between hardware revisions.

Note that the RF amplifiers are susceptible to thermal transients when powering on and off, causing the output power to fluctuate. Switching the amplifiers (e.g. using CHx-ON) may result in RF output within several milliseconds, but the output power may take several seconds to stabilise. Switching the RF only (e.g. using CHx-OFF) is recommended to avoid transients.

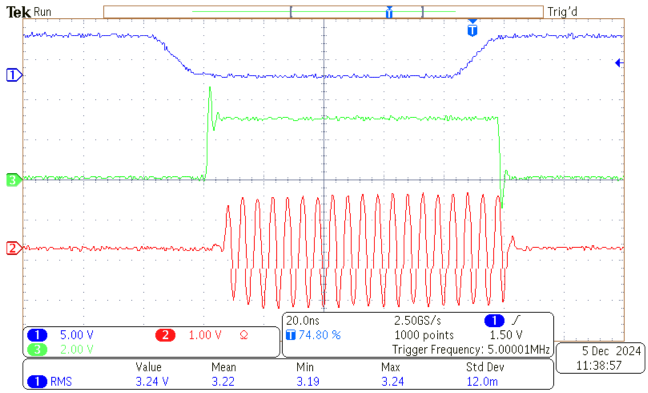


Figure 7.4: Modulation of RF output using the CHx-OFF input. Blue is TTL signal at the source; green is the TTL signal at the RF switch (internal to XRF); red is the RF signal.

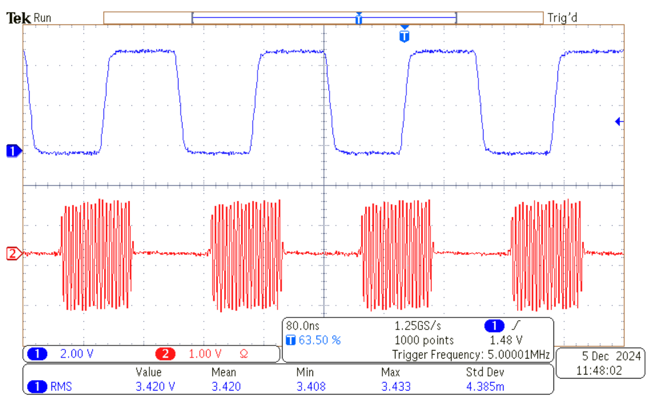


Figure 7.5: Example of pulse generation using the CHx-OFF input. Red is TTL signal; blue is the RF signal.

7.7 Counters

Fast digital counters can be accessed for each digital input pin. XRF devices can use these counters in advanced table mode (§9.4); XRF devices can only use them manually in scripts. To use a counter, the associated pin must be in READ mode and the counter function activated. The maximum edge-detection rate on the high-speed bus is 50 MHz, and level-detection (HIGH and LOW modes) accumulates 125 counts per microsecond.

The syntax to control counters is `EXTIO,COUNTER,ch,pin,command`, where `command` is one of the following:

READ, V[ALUE] Return the counter value as a 32-bit number

RESET, C[LEAR] Reset the counter value to zero

E[NABLE] Activate counter and begin accumulating count

D[ISABLE] Deactivate counter but hold count value

H[IGH] Count while input is HIGH, enables counter

L[OW] Count while input is LOW, enables counter

R[ISING] Count rising edges, enables counter

F[ALLING] Count falling edges, enables counter

B[OTH] Count both rising and falling edges, enables counter

The following NSB-mode example sets up a rising edge counter on HSB3 and counts for approximately 100 ms.[1]

```
EXTIO,MODE,2,HSB,READ
EXTIO,COUNTER,2,HS3,RISING
EXTIO,COUNTER,2,HS3,RESET
SLEEP,100 # wait approximately 100ms
EXTIO,COUNTER,2,HS3,READ # returns counts recorded
```

[1] Advanced table mode should be used for more accurate measurement (§9.4).

7.8 Examples

The following examples demonstrate how to configure and use the external I/O pins. Note that pins must be set to MANUAL using the `EXTIO,CTRL` command to be used for READ and WRITE.

These commands may be useful in executing scripts or diagnosing experiments. For any application where timing is important, table mode should be used.

`EXTIO,CTRL,1,HSB,MAN`
: Set HSB1 to MANUAL mode, for use with READ and WRITE

`EXTIO,WRITE,1,DOUT,1`
: Sets the CH1-DOut pin (DE15) to HIGH

`EXTIO,READ,2,OFF`
: Reads the current state of the CH2-OFF pin (DE15)

`EXTIO,MODE,1,HSB,WRITE`
: Set the entire first high-speed bank into write mode

`EXTIO,WRITE,1,HS7,ON`
: Sets port 7 of HSB1 to TTL HIGH

`EXTIO,WRITE,1,HSB,0x7`
: Simultaneously writes all pins in HSB1. Sets pins 0–2 HIGH and pins 3–7 LOW

`EXTIO,MODE,2,HSB,READ`
: Sets the entire second high-speed bank into read mode

`EXTIO,READ,2,HSB`
: Simultaneously read all 8-inputs of the second HSB, and return the result as an 8-bit number

`EXTIO,READ,2,HS3`
: Read only port 3 of HSB2 (returns `ON` or `OFF`)

8. Simple table mode

Table mode performs sequential execution of up to 8191 instructions with precise timing. This enables generation of complicated pulse sequences, custom envelope shapes, and automated control of experiment sequences through digital I/O.

There are two versions of table mode: *simple* table mode (TSB mode) which utilises the DDS serial interface, and *advanced* table mode (TPA mode) that utilises the parallel interface. Advanced table mode has increased functionality and improved timing resolution, as described in chapter 9, and is only available in XRF devices.

8.1 Operational principle

A table is defined as a number of entries that describe the frequency, amplitude and phase of the rf output at each step, as well as any desired I/O. These are preloaded by the FPGA into a DDS*profile*, so that when the sequence is executed the parameters are updated instantaneously. However, this makes table mode incompatible with both external modulation and PID control.

The speed of the serial interface limits the rate at which new instructions can be loaded into the DDS, so the duration of each table entry is discretised at 1 μs.[1]

Once the sequence has been defined using the `TABLE,ENTRY` commands, it is readied for execution using the `TABLE,ARM` command. The table is checked for errors, and will fail if an incompatibility is detected. For example, to use digital output, the associated pin must be configured for AUTO control (§7.4).

Once the table is armed, execution is started by either a hardware TTL trigger on the SEQ input or using the `TABLE,START` command. The

[1] Advanced table mode (XRF) is capable of 16 ns steps using the parallel bus.

phase-accumulator of the DDS is then reset and the table executes autonomously under FPGA control. This provides a very high degree of reproducibility in terms of both timing of instructions and output of the DDS generators, as the DDS phase accumulator is reset for every execution.

The table can be automatically restarted after completion by enabling the `TABLE,RESTART` option, and execution can be stopped mid-sequence using the `TABLE,STOP` command.

Each channel has its own independent table, and there are *slots* for four distinct tables in non-volatile memory. Commonly-used tables can then be stored on the device for later use. However, it should be noted that stored tables may become inoperable after a firmware upgrade and tables should be archived in human-readable form.

When a table is armed, the RF is switched on (including the amplifiers for 421-models), and upon completion of the table the final RF state remains ongoing. If it is required that the output be disabled when the table is complete, the final entry should set the power to `0x0` (zero amplitude, not 0 dBm).

8.2 Defining table entries

Table entries can be defined or queried using the `TABLE,ENTRY` command. Once the entries have been set, the `TABLE,ENTRIES` command should be used to set the length of the table. Alternately, the `TABLE,APPEND` command can be used to add an entry to the end of the table and update the count automatically.

It is recommended to always explicitly empty the table with the `TABLE,CLEAR` command before defining the entries.

TABLE,ENTRY `TABLE,ENTRY,ch,num,[freq,pow,phas,dur,flags]`
Configure the specified table entry. If only `ch` and `num` are given, the current entry of the table is returned.

ch The channel to edit (1 or 2)

num The entry to edit (1 to 8191)

freq Frequency to output during this step

pow Output power during this step

phas Phase of the RF for this step

dur Duration of this step (discretised at 1 us)

flags A comma-separated list of flags, comprised of the following:

- **OFF** Switch off the RF signal for this step, disabling the output. Must be repeated in subsequent steps for the signal to remain off.
- **TRIG** Repeat the current instruction until a hardware trigger is received, optionally specifying trigger source and edge (§8.4.1). More advanced behaviour is available with the `TABLE,LOOP` command instead (§8.4).
- **IOxy** Perform a digital ouput action on pin *x* as specified by *y* (§8.3.1). Only one I/O operation can be performed in a given table entry.
- **IOSETx** Write multiple HSB digital outputs simultaneously (§8.3.2) in conjunction with an IOMASK flag.
- **IOMASKy** Specifies which outputs should be controlled by IOSET (§8.3.2).

The maximum duration for simple table mode is normally $2^{20} - 1\,\mu$s, 1.048 s. For multiple output mode (§8.3.2), the limit is $2^{16} - 1\,\mu$s, 65.535 ms. For advanced table mode, the maximum duration is $2^{32}-1$ times 16 ns, about 68 s.

The following commands are also provided to simplify editing tables in memory. They take the same parameters as the `TABLE,ENTRY` command but automatically update the table entry count as relevant.

TABLE,APPEND `TABLE,APPEND,ch,freq,pow,phas,dur,flags`
Add the specified table entry to the end of the table and increment the entry counter.

TABLE,INSERT `TABLE,INSERT,ch,num,freq,pow,phas,dur,flags`
Insert the specified entry into the table at the specified index, and shift all other entries down.

TABLE,DELETE `TABLE,DELETE,ch,num`
Remove only the specified entry from the table.

TABLE,CLEAR `TABLE,CLEAR,ch`
Deletes the entire table from memory and resets any state variables.

Listing 8.1 shows basic operation of table mode using the `TABLE,ENTRY` command to define the instructions. Entries are loaded in individually then the total number is set with `TABLE,ENTRIES`. The table is armed and executed using `TABLE,START`, resulting in typical output shown in Figure 8.1.

The last instruction is held after the table completes, so it is good practice to include a final instruction that sets the output power to 0x0 if it is desired for the RF to be off after execution finishes.

```
# Example of table output
MODE,1,TSB
# Begin table
TABLE,ENTRY,1,1,100MHz,-10dBm,0,100us
TABLE,ENTRY,1,2,100MHz,0dBm,0,100us
TABLE,ENTRY,1,3,80MHz,-5dBm,0,100us
TABLE,ENTRY,1,4,80MHz,-15.0dBm,0,100us
TABLE,ENTRY,1,5,100MHz,-2.0dBm,0,100us
TABLE,ENTRY,1,6,100MHz,0x0C00,0,100us
TABLE,ENTRY,1,7,100MHz,0x0200,0,100us
TABLE,ENTRY,1,8,100MHz,0x0,0,100us
TABLE,ENTRIES,1,8
TABLE,START,1
```

Listing 8.1: Simple table mode demonstration.

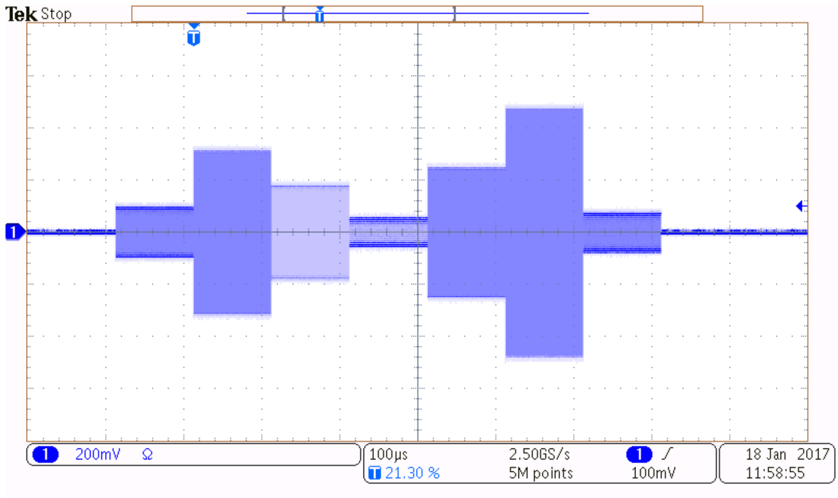


Figure 8.1: Demonstration of RF output generated by Listing 8.1.

8.3 Digital I/O

Each entry in the table can control a single digital I/O pin (§8.3.1), or write multiple pins simultaneously (§8.3.2). However, pins must be correctly configured using the `EXTIO` to be used in table mode: inputs must be set to READ mode, and outputs must be set to WRITE mode with AUTO control.

8.3.1 Digital output

Digital output can be controlled from the table on any pin which has been configured for AUTO control using the `EXTIO,CTRL` command (§7.4). The output is configured using a flag of the form `IOxy`, where `x` is the pin to use and `y` the function to perform.

The pin can specified as:

D The digital output (DOUT) associated with this channel on the DE15 connector

0-7 The specified pin of the high-speed bank with the same number as this channel (A for CH1, B for CH2)

A0-A7 The specified pin of high-speed bank A (irrespective of channel)

B0-B7 The specified pin of high-speed bank B (irrespective of channel)

Both channels are capable of accessing the same I/O pins using the second notation. It is up to the user to ensure that this does not cause conflicts, particularly when loading a table from memory.

Available functions are listed below. Only the first letter is significant when specifying the command.

L[OW] Set the pin to output digital LOW

H[IGH] Set the pin to output digital HIGH

T[OGGLE] Switch the output level of specified pin

P[ULSE] Output a short pulse ($\sim 500\,\text{ns}$) on specified pin

Examples of I/O flags:

`IO3H` Set pin 3 of associated high-speed bank to output HIGH

`IODT` Toggle the output of the associated DOUT pin on the DE15 connector

`IOA2P` Output a short pulse on pin 2 of high-speed bank A

`IOB1L` Set pin 1 of high-speed bank B to output digital LOW

The example below shows how to toggle an output in the high-speed bank, resulting in the output shown in Figure 8.2.

```
MODE,1,TSB
# set HSB—A to output
EXTIO,MODE,1,HSB,WRITE
# set HSB pin A1 to AUTO mode
EXTIO,CONTROL,1,HS1,AUTO
# define table —— same frequency, five different amplitudes
TABLE,ENTRY,1,1,100MHz,0x200,0,2
# next entry sets pin 1 high
```

```
TABLE,ENTRY,1,2,100MHz,0x600,0,2,IO1H
# 2us later, set pin 1 low
TABLE,ENTRY,1,3,100MHz,0x1000,0,2,IO1L
TABLE,ENTRY,1,4,100MHz,0x1400,0,2
# create 500ns pulse
TABLE,ENTRY,1,5,100MHz,0x2000,0,2,IO1P
TABLE,ENTRY,1,6,100MHz,0x200,0,2
# toggle pin 1 (from low to high)
TABLE,ENTRY,1,7,100MHz,0x600,0,2,IO1T
TABLE,ENTRY,1,8,100MHz,0x1000,0,2
# toggle pin 1 again (i.e. from high to low)
TABLE,ENTRY,1,9,100MHz,0x1400,0,2,IO1T
TABLE,ENTRY,1,10,100MHz,0x2000,0,2
TABLE,ENTRIES,1,10
```

Listing 8.2: Simple example showing digital output in table mode.

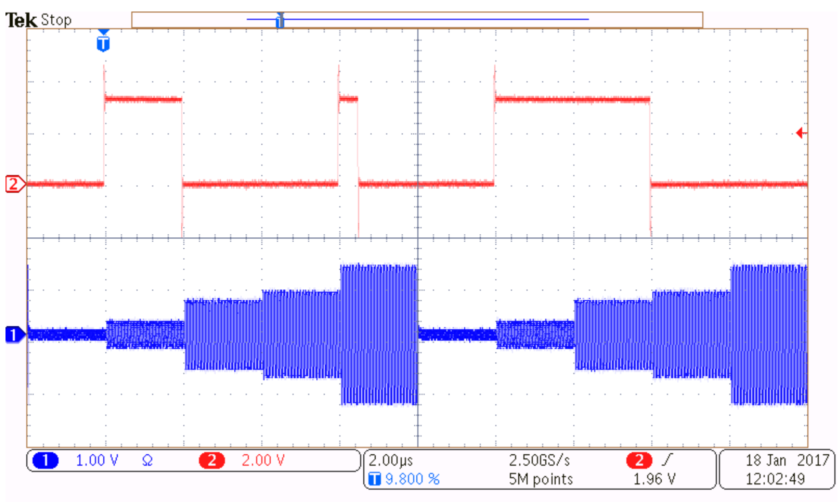


Figure 8.2: Example of table mode, showing changing RF output (blue, lower trace) and synchronised TTL output (red, upper trace) generated by the example in Listing 8.2.

8.3.2 Simultaneous digital output

As of firmware v1.3.0, multiple digital outputs can be set simultaneously in a single table instruction using the IOSET and IOMASK flags. Both instructions take a 16-bit number, where each bit corresponds to a pin of the high-speed output banks. If a bit is set in the IOMASK value, then the corresponding value of IOSET is written to that pin, overwriting any previous value.

This allows all pins of both high-speed banks to be changed in each table entry, or only a subset of the pins. Table 8.1 demonstrates an example that simultaneously writes to 5 pins of bank A and 4 pins of bank B.

	Bank B								Bank A							
Bit	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
IOSET	0	0	1	0	1	1	1	1	1	0	0	1	0	0	1	1
IOMASK	0	1	0	0	1	1	0	1	1	1	1	0	1	0	1	0
Outcome	-	0	-	-	1	1	-	1	1	0	0	-	0	-	1	-

Table 8.1: Example of simultaneous output using IOSET0x2F93 and IOMASK0x4DEA. Only the 9 pins corresponding to bits set in IOMASK are affected by the command, with "-" denoting no change.

If IOMASK is not specified, the value `0xFFFF` is assumed and all HSB outputs are written at once. Care must be taken when running two tables simultaneously to ensure that the same pin isn't being written to simultaneously by both tables, which can result in undefined behaviour.

Where only a few outputs need to be combined, it is possible to chain together multiple IOxH and IOxL flags on the same table entry to avoid needing to construct the mask word. For example, the following set of output flags could be used,

`IOA3H, IOA4L, IOB1H`,

which is equivalent to

`IOSET0x0208, IOMASK0x0218`,

but has improved clarity.

For reference, the flag `IOMASK0x00FF` will only write to bank A, and `IOMASK0xFF00` will only write to bank B. It is recommended to encode values in hexadecimal in this way to improve readability.

Note: Multiple digital output mode cannot be used in combination with `LOOP` or `TRIG`.

8.4 Loops and triggers

Table mode supports the use of loops, to simplify the generation of pulse sequences or to wait for a hardware trigger to synchronise execution with external devices. The general syntax for defining a loop is described below. A simplified option for external triggering (the `TRIG` flag) is described in §8.4.1.

TABLE,LOOP `TABLE,LOOP,ch,source,dest,condition`
Set the table to jump from the `source` entry to the `dest` entry until the `condition` is satisfied. If `source` and `dest` are the same, or if `dest` is 0, then the table effectively holds the instruction.

`source` and `dest` can be negative numbers, which are then taken as offsets. If `source` is negative, it is taken as an offset from the end of the table (requires `TABLE,ENTRIES` to be set). If `dest` is negative, it is taken as the offset from the `source`. This is particularly convenient when using the `TABLE,APPEND` command which automatically updates `TABLE,ENTRIES`.

The `condition` can be an integer in the range $[1, 4095]$, corresponding to the number of times to execute the loop, or a hardware descriptor flag of the form `IOxy`, indicating to repeat until the digital input pin `x` exhibits behaviour `y`, as described below.

The pin syntax is described in §8.3.1, with the addition that "D" refers to the associated trigger input (CHx-SEQ) of the DE15 connector. The behaviour `y` is one of the following options:

H[IGH] Terminate loop on logic level HIGH *at* the loop instruction

L[OW]	Terminate loop on logic level LOW *at* the loop instruction
F[ALLING]	Terminate loop *after* falling edge occurs
R[ISING]	Terminate loop *after* rising edge occurs

> **Note:** When using the TTL inputs as a trigger in table mode, it is strongly recommended to ensure that the input debouncer is disabled using the `DEBOUNCE` command.

Loops are subject to the following restrictions:

- Nested loops are not supported.
- The `TABLE,LOOP` command can only be used after the `source` entry has been defined.
- The *first* entry and *last three* entries of the table cannot contain a `LOOP` or `TRIG`.
- Loop conditions are checked at the *start* of the instruction, so if the condition is met *during* the instruction, it will take effect on the next repetition.
- All loops will execute *at least once*, even if the `condition` is met when the loop is first entered.
- In simple table mode there must be a minimum of 4 table entries between consecutive `LOOP` instructions.

The example below shows how to define a loop with a fixed number of iterations. The loop counter is 4, so instructions 1–3 are executed a total of 5 times (Figure 8.3). Since the final three instructions cannot contain a loop, several dummy instruction are added to the end which also turn off the RF.

```
TABLE,CLEAR,1
TABLE,APPEND,1,100MHz,0dBm,0,1us
TABLE,APPEND,1,100MHz,-5dBm,0,4us
TABLE,APPEND,1,100MHz,-10dBm,0,2us
TABLE,LOOP,1,3,1,4
TABLE,APPEND,1,100MHz,-30dBm,0,1us,OFF
```

```
TABLE,APPEND,1,100MHz,-30dBm,0,1us,OFF
TABLE,APPEND,1,100MHz,-30dBm,0,1us,OFF
```

Listing 8.3: Demonstration of a simple loop

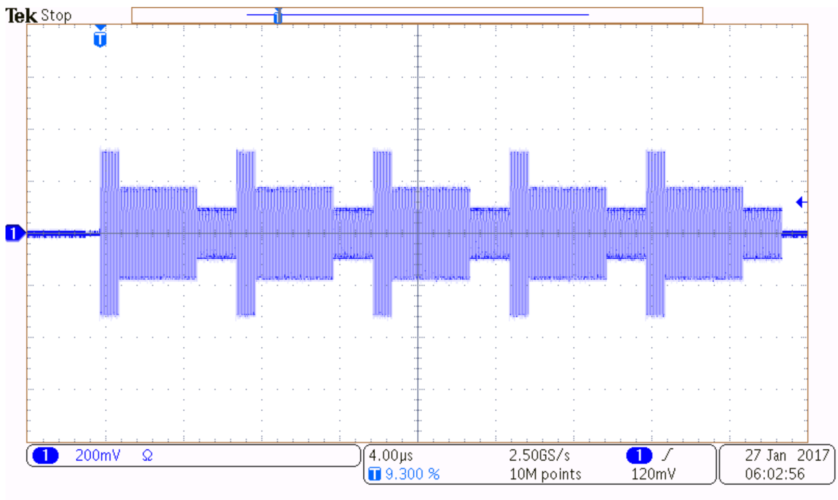


Figure 8.3: Demonstration of a simple loop.

An alternate way of specifying the `LOOP` instruction in the above example is `TABLE,LOOP,1,-1,-2,4`, since the loop should activate after the most recently added entry (source -1), and the destination (entry 1) is 2 instructions earlier.

8.4.1 External trigger flag

An alternate way to wait for an external trigger is to use the `TRIG` flag on a table entry. This alleviates the need for a separate call to the `TABLE,LOOP` command. The flag is of the form `TRIGxy` where `x` is the pin (e.g. `D` or `A0`) and `y` the condition. If neither `x` nor `y` are specified, the default is `TRIGDF`.

Similarly to the general loop, the trigger behaviour is to **repeat the current instruction until a trigger is received**. If the trigger condition is met *during* the instruction period, the duration of the current instruction is completed first. Hence the duration of a TRIG instruction should be *as small as possible* to ensure rapid response.

The example below shows how to use the TRIG flag, with typical result shown in Figure 8.4.

```
MODE,1,TSB # set table mode
EXTIO,CTRL,1,HSB,AUTO # ready HSB1 for output
TABLE,CLEAR,1
TABLE,APPEND,1,100,5,0,10
TABLE,APPEND,1,100,10,0,1,IO1T,TRIG # wait for trigger
TABLE,APPEND,1,100,-5,0,10
TABLE,APPEND,1,100,-30,0,3,IO1L
TABLE,ARM,1 # ready table for execution
```

Listing 8.4: Demonstration of a table with a trigger command

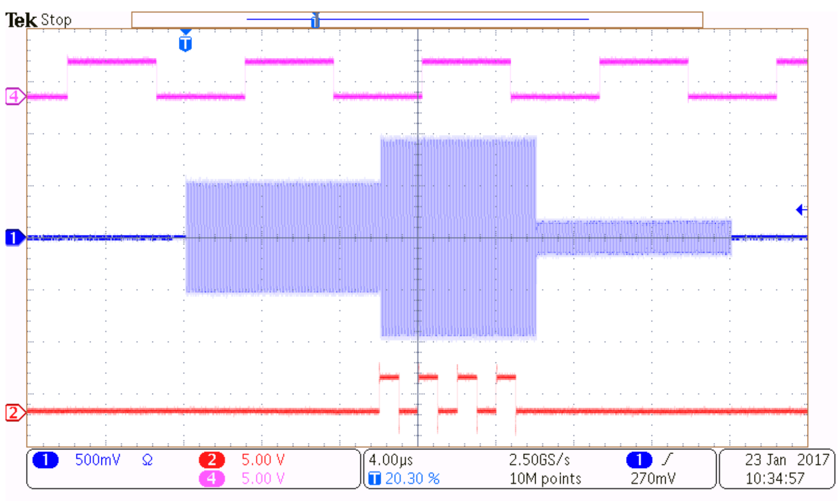


Figure 8.4: Waiting for a trigger causes the entry to be repeated until a falling edge in the trigger (magenta) is detected. The digital output (red) toggles several times, showing the repetition.

Once a table is armed with the TABLE,ARM command it will automatically begin executing upon a falling external trigger on the DE15 input. Starting the table execution from an external trigger therefore does not require the TRIG flag on the first table entry. If a trigger is not received but an instruction is waiting for a trigger, the TABLE,STOP or OFF command should be used to abort operation.

8.5 Upload and download

The host software mogrf provides convenient functionality that enables upload and download of tables in both binary and CSV format. This simplifies handling of tables, and allows simple generation of sequences from scripting languages like matlab or python, and the human-readable structure simplifies trouble-shooting.

Binary format enables direct upload of table data into non-volatile memory and is provided for fast back-up purposes. **The internal binary representation of tables is likely to change between minor firmware revisions** due to the implementation of new functionality that makes old binary tables incompatible with new firmware. It is strongly recommended that all tables be stored in ASCII (human-readable) format for long-term storage. An example of this format is shown below.

```
100 MHz, -5 dBm,  0 deg, 10us
150 MHz,  0 dBm, 90 deg, 2us, IODH
 80 MHz,  5 dBm,  0 deg, 1us, TRIG
100 MHz,  0 dBm,  0 deg, 5us
```

Listing 8.5: Example of table mode CSV file for use with mogrf.

A distinction should be made between *CSV tables* and *scripts*. In this context, CSV tables contain only the table instructions directly, whereas scripts may contain arbitrary sequences of command for the device. Listing 8.5 demonstrates the format of a CSV table. The intention is that such files are more easily generated by scripting languages (such as matlab or python).

```
MODE,1,TSB
TABLE,CLEAR,1
TABLE,ENTRIES,1,4
```

```
TABLE,ENTRY,1,1,100MHz,-5dBm,0deg,10us
TABLE,ENTRY,1,2,150MHz,0dBm,90deg,2us,IODH
TABLE,ENTRY,1,3,80MHz,5dBm,0deg,1us,TRIG
TABLE,ENTRY,1,4,100MHz,0dBm,0deg,5us
```

Listing 8.6: Script equivalent to Listing 8.5.

8.6 Re-arm and restart

The FPGA can be instructed to automatically re-arm the table after a successful execution using the `TABLE,REARM` command. This automatically prepares the table for execution from either hardware or software trigger once execution has finished.

Furthermore, the table can be repeated continuously by enabling the `TABLE,RESTART` option. This will cause the table to immediately re-execute after it has been rearmed, although a hardware or software trigger must be provided to begin the first execution.

Typically, the time between a table completing and subsequently restarting under FPGA control is $\sim 5\mu s$, depending on the length of the table and any synchronisation features being used. Enabling the `NORESET` option will prevent resetting the DDS phase accumulator between executions and reduces the delay. Note that unless the `NORESET` option is enabled, the output will be switched off during the phase accumulator reset ($\sim 2\mu s$).

If this delay is unacceptable, an alternative is to specify a loop back to the beginning of the sequence using the `TABLE,LOOP` command which will repeat the table with no delay. As the last table entry cannot be a loop, a dummy instruction needs to be appended, as in the example below. This table will terminate after 4096 executions; an alternative is to specify to repeat until an external trigger that is never provided, which will loop forever.

```
# ... Instructions that define the CH1 table ...
# add a LOOP from the most recent instruction (−1) back to the start (1)
TABLE,LOOP,1,-1,1,4095
# last three instructions cannot be a loop −− add dummy instructions
TABLE,APPEND,1,100MHz,-30dBm,0,1us
TABLE,APPEND,1,100MHz,-30dBm,0,1us
TABLE,APPEND,1,100MHz,-30dBm,0,1us
```

Listing 8.7: Demonstration of restarting CH1 table using a loop.

8.7 Linear ramps

It is often desirable to linearly ramp a parameter without having to specify the individual table entries manually. The `TABLE,RAMP` command provides a convenient way to generate such ramps.

TABLE,RAMP `TABLE,RAMP,ch,param,start,stop,duration,count`
Appends table entries to the channel `ch` that linearly ramp `param` from `start` to `stop` in `count` steps, with each step lasting for the given `duration`. `param` is one of `FREQ`, `AMPL` or `PHAS`. The `start` and `stop` values can be specified with real units as appropriate for `param`. The values of the other parameters are taken from the last entry in the table. In simple table mode, `count` entries are generated, whereas in advanced table mode only 3 entries are required.

For example, the command `TABLE,RAMP,1,FREQ,80,100,100us,2000` in TSB mode will generate 2000 table entries corresponding to a frequency ramp that sweeps the RF frequency from 80 MHz to 100 MHz in a total time of 200 ms (each step being 10 kHz and taking 100 us).

The command can also be used to specify piecewise-linear envelopes using the `AMPL` parameter, such as in Listing 8.8 resulting in the RF output in Figure 8.5.

```
MODE,1,TSB
# clear the table
TABLE,CLEAR,1
# define initial conditions (i.e. freq and phase)
TABLE,APPEND,1,80MHz,-30dBm,0deg,1us
# ramp power from −30dBm to 0dBm in 100us, then down again
TABLE,RAMP,1,POW,-30,0,1us,100
TABLE,RAMP,1,POW,0,-30,1us,100
# should now have 201 entries
TABLE,ENTRIES,1
# arm the table
TABLE,ARM,1
```

Listing 8.8: Example using `TABLE,RAMP` to create an envelope.

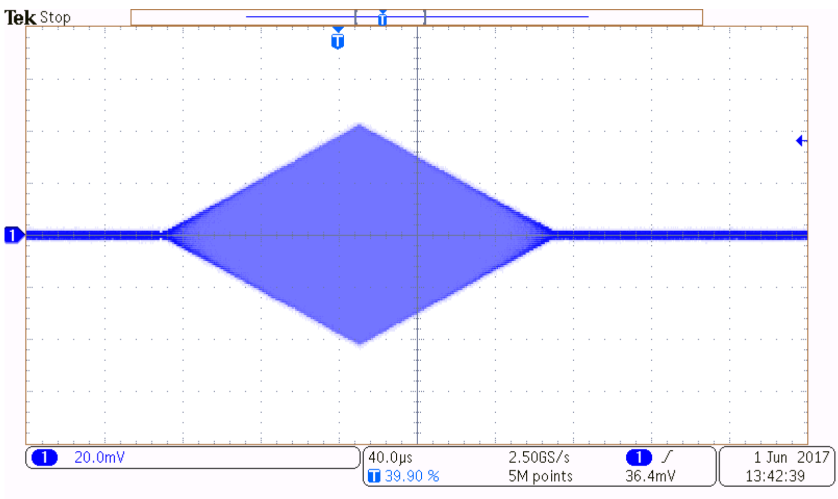


Figure 8.5: Output generated by Listing 8.8 showing linear ramps in amplitude.

The ramp functionality can also be applied to frequency or phase. Listing 8.9 shows how multiple `TABLE,RAMP` commands can be used to execute a sequence of slow frequency ramps that can be watched on a spectrum analyser. A graph of the corresponding frequency of the RF over time is shown in Figure 8.6.

```
MODE,1,TSB
TABLE,CLEAR,1
# set initial conditions
TABLE,APPEND,1,80MHz,0dBm,0,1us
# define first ramp
TABLE,RAMP,1,FREQ,70,80,1m,1000
# append a pause
TABLE,APPEND,1,80,-5dbm,0,1s
# append second and third ramps
TABLE,RAMP,1,FREQ,80,75,5m,200
TABLE,RAMP,1,FREQ,75,85,2m,500
```

Listing 8.9: Example demonstrating use of `TABLE,RAMP` to create multiple frequency ramps.

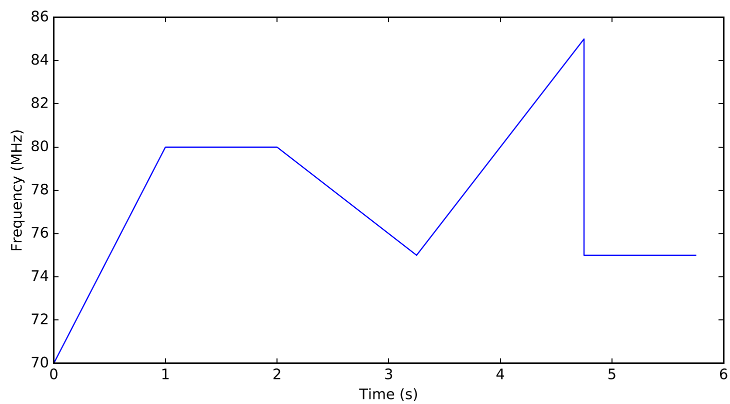


Figure 8.6: Frequency ramps achieved by chaining together `RAMPFREQ` commands in TSB mode.

8.8 Synchronous table execution

When operating both channels in table mode, it is possible to synchronise the two channels. This synchronisation occurs both at FPGA level (so that the table entries are stepped through simultaneously), and at the DDS level (so that the RF generated by the two channels remain in phase).

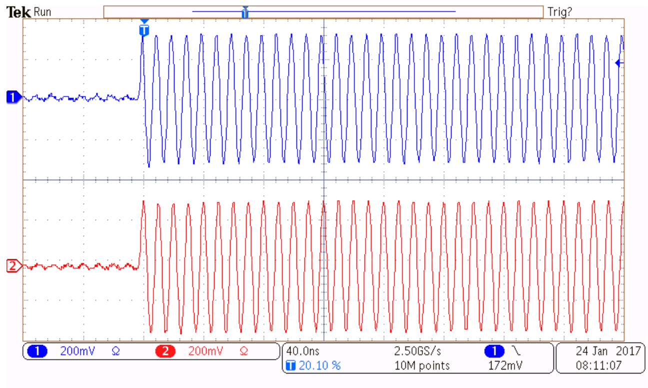


Figure 8.7: Example showing the two RF outputs when executing the same table synchronously on both channels.

This feature is activated by issuing the command `TABLE,SYNC,1`. This configures CH1 as the master and CH2 as the slave, such that the CH2 DDS derives its clock from the CH1 DDS for phase stability. Once enabled, CH2 cannot be started independently of CH1, and CH1 cannot be started unless CH2 has been armed. For this reason, it is recommended to activate the auto-rearm feature on CH2 using the command `TABLE,REARM,2,1`.

The feature must be deactivated using `TABLE,SYNC,0` to run the tables independently, or to take one channel out of table mode. The FPGA is also slightly slower ($\sim 1\,\mu s$) to respond to a start instruction (software or hardware trigger) in synchronous mode, as it must prepare the each channel's DDS for execution.

It is also possible to synchronise the DE15 triggers received by the two tables. The two trigger inputs on the DE15 connector can be hardwired together, but the FPGA can also be instructed to trigger both channels from the same input. This functionality can be controlled by the `TABLE,TRIGSYNC` command.

This does not apply to the high-speed bus, as both tables can be instructed to use the same input as a trigger regardless.

9. Advanced table mode (XRF)

The XRF has access to an *advanced table mode* with increased functionality, flexibility and significantly faster execution than normal (SIF) table mode. Due to the added complexity, it is strongly recommended that users be familiar with simple table mode before reading this chapter.

9.1 Operational principle

Advanced table (TPA) mode performs table execution via the parallel DDS interface. This allows one parameter to be updated at the maximum supported rate, currently 16 ns per instruction. Due to limitations of the AD9910 DDS devices, only one parameter can be modified via PIF, and the other parameters must be updated at the slower rate supported by the serial interface (SIF). Furthermore neither external modulation nor PID can be used in TPA mode.

There are therefore two kinds of instructions in advanced mode: parallel and serial. Parallel instructions allow a single parameter to be updated rapidly, whereas serial instructions are preloaded into the serial interface and then activated in a subsequent command. This load/activate cycle allows changes to be made to the parallel parameter without having to wait for the serial load to complete.

Advanced mode also supports a more extensive selection of loop options, including the ability to increment the parallel parameter for piecewise linear interpolation. This is advantageous for generating smooth envelopes with only a few table instructions.

The FPGA supports I/O at the full update rate, but the rise-time (particularly on the DE15 connector) must be taken into account. The MOGLabs B3120 break-out board is recommended when using the high-speed bus as it has matched track lengths to minimise relative delay between the signals.

It should also be noted that the while the DDS outputs can be synchronised, the rf components following the DDS introduce a small frequency-dependent propagation delay. However, this delay is fixed for a given frequency, and can be calibrated in applications where it is important. Such applications likely need calibration anyway, due to propagation delay introduced by cables and other external components.

9.2 Defining table entries

There are two forms of syntax for defining table entries in advanced mode, corresponding to parallel or serial instructions. Before populating the table, the parallel parameter needs to be set using the `TABLE,XPARAM` command, after which entries can be set for that parameter (syntax 1). A second format is provided for loading serial commands, which is identical to simple table mode (syntax 2).

TABLE,XPARAM `TABLE,XPARAM,ch,param,[fmgain]`
Sets which parameter is to be controlled on the parallel interface. Must be called before the table is populated with entries. The parameter is one of `FREQ`, `PHAS`, `POW` or `AMPL`. The parameters `POW` and `AMPL` are synonyms in this mode. Subsequent PIF table entries can then modify this parameter. When entering `FREQ` mode, the `fmgain` must also be specified (see §9.7).

TABLE,ENTRY(1) `TABLE,ENTRY,ch,num,param,value,duration,flags`
Set the specified table entry to change `param` to the specified `value`, while other parameters remain unchanged. During normal use, `param` should match the parameter set by `TABLE,XPARAM`, but some special instructions are available (§9.8). The `duration` of the entry is rounded to the nearest multiple of 16 ns, and the duration 0x1 can be used to set the minimum possible duration. The same `flags` are supported as in simple table mode (including `TRIG` and `IOxy`), with some extra options as explained in this chapter.

The listing below demonstrates how to set up advanced table mode to control the envelope (amplitude) of the RF signal,

```
# enter fast table mode
MODE,1,TPA
# clear any table entries or settings
TABLE,CLEAR,1
# set amplitude (power) as the parallel parameter
TABLE,XPARAM,1,POW
# define a table
TABLE,APPEND,1,POW,5dbm,16ns
TABLE,APPEND,1,POW,10dbm,50ns
TABLE,APPEND,1,POW,0dbm,16ns
TABLE,APPEND,1,POW,-40dbm,16ns
```

Listing 9.1: Parallel instruction example in advanced table mode.

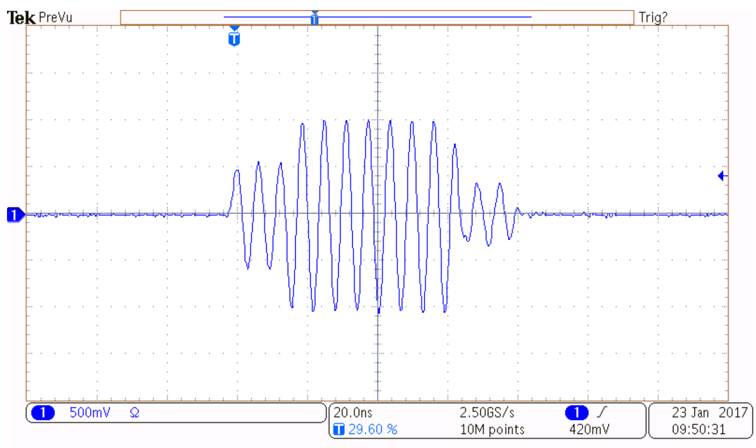


Figure 9.1: Output generated by Listing 9.1.

Parallel-mode instructions also provide functionality for piecewise linear-interpolations, as described in §9.6. This enables the creation of smooth ramps without requiring a large number of instructions. The functions `TABLE,APPEND` and `TABLE,INSERT` can also be used, supporting either syntax.

It is anticipated that many applications will only seek to control a single parameter, in which case only the parallel interface need be used. However, a second command format is provided to simultaneously set all three parameters using the serial interface.

TABLE,ENTRY(2) `TABLE,ENTRY,ch,num,freq,pow,phase,duration,flags`
Defines a serial (SIF) instruction that updates all parameters using the same syntax as simple table mode (§8.2), but the instruction does not take effect immediately. The instructions are queued for load over the serial interface and must be activated by a subsequent table entry using the `UPD` flag. This ensures that the output only changes in accordance with a table instruction.

The time between issuing the serial instruction and the update flag must be at least 960ns otherwise the DDS does not have time to load the instructions. However, any flags specified in the instruction (such as digital output) will occur immediately.

The example below demonstrates how to queue and then activate a serial instruction in advanced table mode.

```
# define a SIF entry using simple mode syntax
TABLE,APPEND,1,100MHz,5dbm,0,1us
# trigger the update to take effect
TABLE,APPEND,1,POW,0dbm,0x1,UPD
```

Listing 9.2: Serial instruction example in advanced table mode.

This may appear complicated, however it makes it possible to execute instructions *during* a serial load. For example, when making a chirped pulse, the amplitude can still be changing on the parallel bus while the next frequency is being loaded on the serial bus.

```
# set up the table
TABLE,CLEAR,1
TABLE,XPARAM,1,POW
# set the initial conditions
TABLE,APPEND,1,120MHz,-5dbm,0,1us
TABLE,APPEND,1,POW,-5dbm,0x1,UPD
# start loading next serial instruction
TABLE,APPEND,1,40MHz,0dbm,0,0x1
```

```
# some other instructions while the SIF loads
TABLE,APPEND,1,POW,-5dbm,320ns
TABLE,APPEND,1,POW,-10dbm,320ns
TABLE,APPEND,1,POW,-5dbm,320ns
# trigger the serial instruction
TABLE,APPEND,1,POW,5dbm,200ns,UPD
# another parallel instruction at new frequency
TABLE,APPEND,1,POW,-5dbm,100ns
# final instruction to power down
TABLE,APPEND,1,POW,0x0,0x1
```

Listing 9.3: Demonstration of parallel instructions during a SIF load

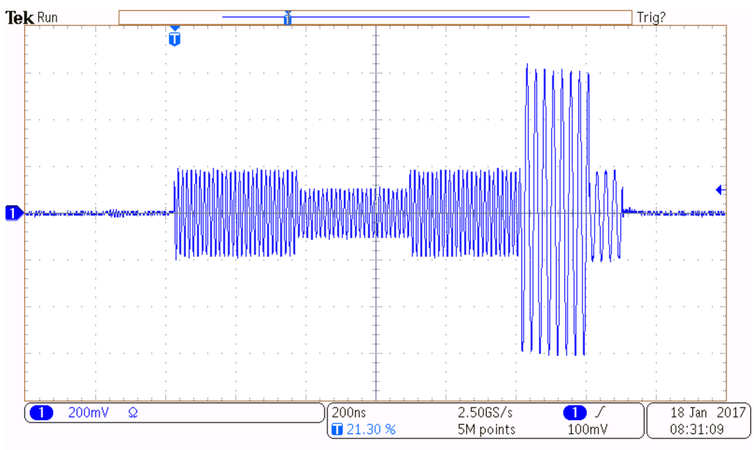


Figure 9.2: Output generated by Listing 9.3

Note: The value corresponding to the parallel parameter is ignored when loading a serial update and must be set separately. For example, in Listing 9.2, the output power will be 0 dBm (not 5 dBm).

9.3 Initial and final states

As in simple table mode, when the TABLE,ARM command is used to ready the table for execution, the output is enabled and amplifiers switched on (if present). The state of the RF after arming is therefore the same as the last table instruction that was run.

In simple table mode, the first instruction specified all three parameters, so the state at each step is well-defined and repeatable. However, in parallel mode only one parameter is specified and the other two may be undefined. It is strongly recommended that the start and end of every table reset all parameters to known values, as demonstrated in the listing below.

```
TABLE,CLEAR,1
TABLE,XPARAM,1,POW
# create table entry specifying a known start condition
TABLE,APPEND,1,80MHz,5dbm,45deg,976ns,OFF
# activate the update
TABLE,APPEND,1,POW,5dbm,16ns,UPD
# -- 1us has elapsed up to here --
#
# ... other table mode entries ...
#
# add a final entry to reset rf state
TABLE,APPEND,80MHz,5dbm,45deg,1us
# activate the update, and turn off the rf
TABLE,APPEND,1,POW,-30dbm,UPD,OFF
```

Listing 9.4: Setting initial and final states in parallel mode

The RF state at the last instruction is held after the table has finished executing, which may involve the RF being on or off depending on the desired application. If it is desired for the RF to be switched off, then the amplitude should be set to zero on the final instruction instead of using the OFF flag, as the RF output will be switched on immediately following the next table rearm.

It is also possible to manually use the commands FREQ, POW and PHASE in advanced table mode to set the initial conditions *when the table is not running*. Once the table has finished executing, the last instruction will remain unless subsequently overwritten by one of these commands. It is therefore strongly recommended to specify the initial and final states.

9.4 Counters

XRF devices are capable of controlling the digital input counters in advanced table mode, and using the counters as a loop condition. This assists in experiment automation, whereby the execution can be paused until a critical count is received. For example, the experiment can be paused until sufficient pulses are recorded from an avalanche photodetector, indicating the experiment is ready to proceed.

Counters can be controlled through the following advanced table mode flags, appended to any table entry. The pin must be enabled as a counter (§7.7) before arming the table, and each command takes effect at the *start* of each table instruction.

CxS[TART] Reset counter `x` to zero and then begin counting

CxP[AUSE] Disable counter `x` and stop accumulating counts

CxR[ESUME] Re-enable counter `x` and resume accumulating counts

CA[LL] Stop and reset all counters

The parameter `x` in each of these flags is the counter pin, such as `A3` for pin 3 of bank A, or `D` for OFFn on the DE15 connector.

Unlike basic mode, where the counter begins accumulating counts as soon as it is enabled, the counter in table mode only accumulates counts after a `CxSTART` or `CxRESUME` flag. This allows precise control of the counting interval.

The syntax for looping until a threshold counter value is reached is

```
LOOP,ch,source,dest,COUNT,IOx,N
```

where `x` is the counter pin and `N` is the count threshold, which is an integer in the range [1,65535].

The example below shows how to configure a counter, control it with table flags, and use it as a loop condition.

```
# configure the counter
EXTIO,MODE,1,HSB,READ # set bank A into read mode
EXTIO,COUNTER,1,HS1,FALLING # set pin A1 to count falling edges
# create the table
MODE,1,TPA
TABLE,CLEAR,1
TABLE,XPARAM,1,POW
TABLE,APPEND,1,POW,-5dBm,100ns,CA1S # start the counter
TABLE,APPEND,1,POW,0dBm,16ns # accumulate counts
TABLE,LOOP,1,-1,0,COUNT,IOA1,1000 # loop until 1000 counts
TABLE,APPEND,1,POW,-30dBm,16ns,CA1P # pause the counter
# run the table
TABLE,START,1
# wait for completion then read back the count
SLEEP,10
TABLE,STATUS,1
EXTIO,COUNTER,1,HS1,READ
```

Listing 9.5: Demonstration of configuring HSA1 as a counter and controlling it in advanced table mode.

9.5 Loops and triggers

Loops and triggers can be specified in advanced table mode using the `TABLE,LOOP` command similarly to simple table mode (§8.4). However, some different restrictions apply to loops in advanced table mode:

- Nested loops are not supported
- Neither the first nor last table instruction can be a loop
- The `TABLE,LOOP` command cannot be applied to a table entry that uses the EXTRAPOLATE feature (§9.6)
- A loop cannot jump by more than 1024 instructions
- The loop condition can specify up to 65535 repeats
- Sequential instructions may contain loops, unlike TSB mode where loops need to be separated by four instructions.

9.6 Linear ramps using extrapolation

One of the powerful features provided in advanced table mode is the ability to specify linear ramps in parallel mode, which reduces the number of instructions necessary to produce smooth piecewise-linear ramps. The FPGA performs the extrapolation and updates the DDS as required. This means that a 1000-point ramp can be implemented as a single table instruction instead of requiring 1000 different individual instructions.

> **Note:** This section describes the low-level implementation of parameter extrapolation. It is strongly recommended to use the high-level helper functionality provided by the `TABLE,RAMP` command, which makes use of the extrapolate feature in TPA mode and provides a simpler user interface.

The extrapolation feature is activated by specifying the `REPn` flag in a table entry, using the syntax shown below.

TABLE,ENTRY[3] `TABLE,ENTRY,ch,num,param,delta,duration,REPn,flags`
Sets the associated table entry to EXTRAPOLATE, adding `delta` to the parameter `param` on each execution. The instruction is repeated *n* times as specified by `REPn`.

The `delta` argument should be specified in hexadecimal when extrapolating power/amplitude, as in this example. Hexadecimal values can be obtained from the output of the `POW` command for each end-point. Subtract the two and divide by the number of steps to obtain the `delta`. ***Do not specify delta in dBm or W.*** Conversely, the `TABLE,RAMP` function does accept values in real-world units.

> **Warning:** Bounds checking is not performed in extrapolation mode; it is up to the user to ensure that parameters do not go out of bounds. In particular, the power `LIMIT` is not obeyed, and amplitude must not go negative. It is ***strongly recommended*** to check the output on an oscilloscope through an RF attenuator before connecting to a device that could be damaged by maximum output power.

Listing 9.6 demonstrates how to linearly ramp the RF power in 100 steps using a single instruction instead of 100 individual instructions by using the `REPn` notation. The result is shown in Figure 9.3. Listing 9.7 demonstrates an equivalent formulation based on the `TABLE,RAMP` command, which provides the convenience of specifying the power in real units (dBm) instead of hexadecimal increments.

```
MODE,1,TPA
TABLE,CLEAR,1
# fast parameter is power
TABLE,XPARAM,1,POW
# set power to OFF
TABLE,APPEND,1,POW,0x0,0x1
# linear ramp up (100 steps)
TABLE,APPEND,1,POW,0x10,0x1,REP100
# linear ramp down (100 steps)
TABLE,APPEND,1,POW,-0x10,0x1,REP100
# reset power to OFF
TABLE,APPEND,1,POW,0x0,0x1
# loop the ramp 2 more times
TABLE,LOOP,1,-1,1,2
# loop the ramp 2 more times
TABLE,APPEND,1,POW,0x0,0x1
```

Listing 9.6: Creation of a triangle wave envelope in advanced table mode, using only five table entries.

```
TABLE,CLEAR,1
TABLE,XPARAM,1,POW
# define a ramp from off (0x0) to 0dBm
TABLE,RAMP,1,POW,0x0,0dBm,16ns,100
# append the reverse of the ramp
TABLE,RAMP,1,POW,0dBm,0x0,16ns,100
# loop back to the beginning twice more
TABLE,LOOP,1,-1,1,2
# last entry cannot be a loop
TABLE,APPEND,1,POW,0x0,16ns
```

Listing 9.7: Alternate specification of triangle wave using `TABLE,RAMP`.

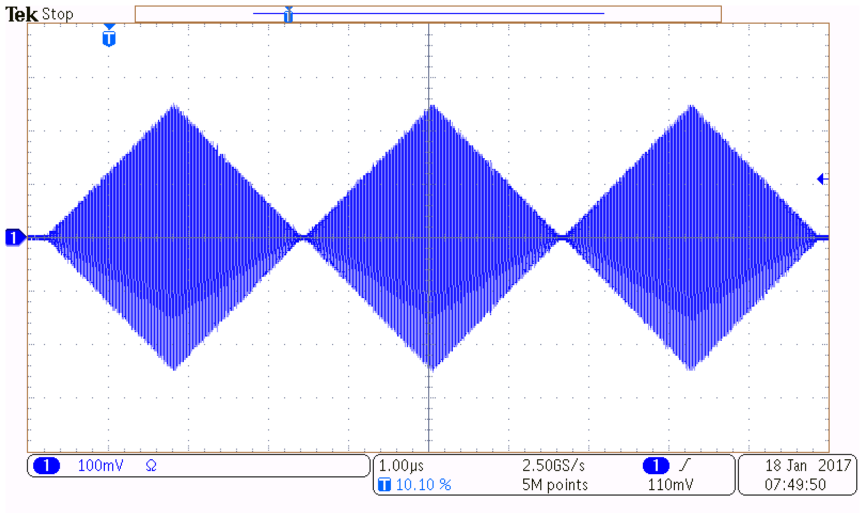


Figure 9.3: Output generated by Listing 9.6.

9.7 Frequency gain

The fast (parallel) interface to the DDS is 16-bit, which is sufficient to fully define the amplitude (14-bits) and phase (16-bits) but not frequency (32-bits). This is an inherent restriction of the DDS, so in order to specify frequency on the parallel interface, a frequency gain is applied by the DDS when receiving values, which amounts to a *bit-shift* of the incoming value (Figure 9.4).

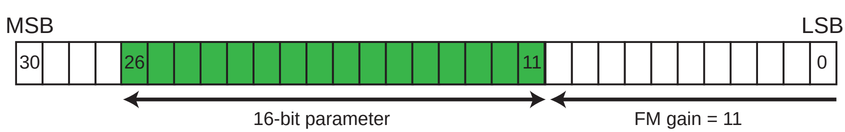


Figure 9.4: Visualisation of how frequency gain allows the 32-bit frequency word to be modified by the 16-bit parallel bus. In this example, the gain is 11 and only the indicated bits can be modified on the parallel bus.

This has the effect of restricting the smallest and largest changes that can be made in parallel frequency mode through the associated frequency discretisation (step size) and value range. If large changes to frequency are required, high gain should be used. If high resolution is required, small gain should be used. The outcome of different gain settings are shown in Table 9.1.

Gain	Step	Max	Gain	Step	Max
0	0.23 Hz	7.54 kHz	8	59.6 Hz	1.95 MHz
1	0.47 Hz	15.3 kHz	9	119 Hz	3.91 MHz
2	0.93 Hz	30.5 kHz	10	238 Hz	7.81 MHz
3	1.86 Hz	61.0 kHz	11	477 Hz	15.6 MHz
4	3.73 Hz	122 kHz	12	953 Hz	31.2 MHz
5	7.45 Hz	244 kHz	13	1.91 kHz	62.5 MHz
6	14.9 Hz	488 kHz	14	3.81 kHz	125 MHz
7	29.8 Hz	977 kHz	15	7.63 kHz	250 MHz

Table 9.1: Effect of frequency gain with default clock configuration.

The gain is specified initially using the `TABLE,XPARAM` command,

`TABLE,XPARAM,ch,FREQ,gain`

where `ch` is the channel and `gain` is the desired gain (0–15).

The range of frequencies that can be achieved in advanced table mode is $f_0 \pm df_{\mathrm{max}}$ where f_0 is the center frequency set with the `FREQ` command, and df_{max} is the value in the above table corresponding to the gain. To assist with understanding these ranges, the `TABLE,XPARAM,ch,FREQ` command will output information about what combinations are possible for the *current* set of parameters.

Note that the frequency gain can presently only be set before the table is populated, and cannot be changed mid-sequence. It is therefore important to ensure that the desired frequency range fits entirely within the accessible range.

For example, to vary the frequency between 70 MHz and 80 MHz with maximum dynamic range, the frequency should be set to 75 MHz using the FREQ command, and the frequency gain set to 10. However, to vary between 65 MHz and 85 MHz, the gain must be increased to 11.

9.8 Other instruction parameters

The following parameters can also be specified in parallel table entries, for special behaviours, as listed below. Instructions that use the serial interface must be followed with a subsequent instruction containing the UPD flag to activate them.

HOLD Do not change the output (also known as a "nop" or "no operation"). Intended to perform I/O operations or trigger a serial update (using the UPD flag) without changing the RF.

REGx Write a 32-bit value directly to register *x* of the DDS using the serial interface. Provided for advanced functionality in consultation with the AD9910 datasheet. Requires subsequent entry with UPD parameter to take effect.

9.9 Additional examples

9.9.1 Gaussian envelope

The following example demonstrates creation of a short (300 ns) Gaussian pulse by specifying the output power at the maximum update rate (instruction duration $0x1 = 16$ ns). The resulting output waveform is shown in Figure 9.5.

```
# set up table mode
MODE,1,TPA
TABLE,CLEAR,1
TABLE,XPARAM,1,POW
# define points along Gaussian
TABLE,APPEND,1,POW,-24.59dBm,0x1
TABLE,APPEND,1,POW,-22.08dBm,0x1
TABLE,APPEND,1,POW,-18.88dBm,0x1
TABLE,APPEND,1,POW,-14.99dBm,0x1
TABLE,APPEND,1,POW,-10.53dBm,0x1
TABLE,APPEND,1,POW,-5.74dBm,0x1
TABLE,APPEND,1,POW,-0.95dBm,0x1
TABLE,APPEND,1,POW, 3.41dBm,0x1
TABLE,APPEND,1,POW, 6.92dBm,0x1
TABLE,APPEND,1,POW, 9.21dBm,0x1
TABLE,APPEND,1,POW,10.00dBm,0x1
TABLE,APPEND,1,POW, 9.21dBm,0x1
TABLE,APPEND,1,POW, 6.92dBm,0x1
TABLE,APPEND,1,POW, 3.41dBm,0x1
TABLE,APPEND,1,POW,-0.95dBm,0x1
TABLE,APPEND,1,POW,-5.74dBm,0x1
TABLE,APPEND,1,POW,-10.53dBm,0x1
TABLE,APPEND,1,POW,-14.99dBm,0x1
TABLE,APPEND,1,POW,-18.88dBm,0x1
TABLE,APPEND,1,POW,-22.08dBm,0x1
TABLE,APPEND,1,POW,-24.59dBm,0x1
TABLE,ARM,1
```

Listing 9.8: Rapid Gaussian pulse (340ns) in fast table mode

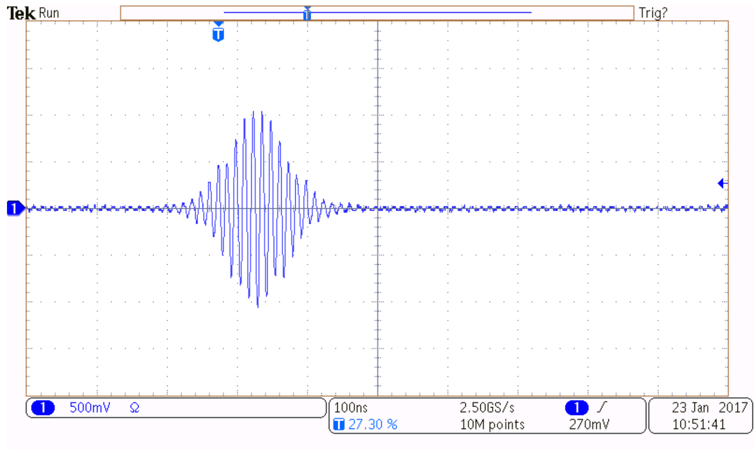


Figure 9.5: Example of a short Gaussian pulse.

9.9.2 *Back-to-back pulses with different frequency*

This example demonstrates how to generate two back-to-back 1µs pulses with different frequencies by loading the second frequency during the first pulse. The EXTRAPOLATE feature is used to smooth the envelope, and the LOOP feature is used to generate the second pulse using the instructions for the first. The amplitude steps are specified in hexadecimal so that the EXTRAPOLATE feature can be used.

```
MODE,1,TPA
TABLE,CLEAR,1
# serial load and trigger first frequency
TABLE,APPEND,1,40MHz,0dbm,0deg,1us
TABLE,APPEND,1,HOLD,0x1,UPD
# begin serial load of second frequency
TABLE,APPEND,1,120MHz,0dbm,0deg,0x1
# define smooth pulse envelope using EXTRAPOLATE
TABLE,APPEND,1,POW,0x001f,0x1,REP3
TABLE,APPEND,1,POW,0x006a,0x1,REP3
TABLE,APPEND,1,POW,0x00d1,0x1,REP3
TABLE,APPEND,1,POW,0x0153,0x1,REP3
TABLE,APPEND,1,POW,0x01db,0x1,REP3
TABLE,APPEND,1,POW,0x0244,0x1,REP3
```

```
TABLE,APPEND,1,POW,0x0264,0x1,REP3
TABLE,APPEND,1,POW,0x0222,0x1,REP3
TABLE,APPEND,1,POW,0x017b,0x1,REP3
TABLE,APPEND,1,POW,0x0088,0x1,REP3
TABLE,APPEND,1,POW,-0x088,0x1,REP3
TABLE,APPEND,1,POW,-0x17b,0x1,REP3
TABLE,APPEND,1,POW,-0x222,0x1,REP3
TABLE,APPEND,1,POW,-0x264,0x1,REP3
TABLE,APPEND,1,POW,-0x244,0x1,REP3
TABLE,APPEND,1,POW,-0x1db,0x1,REP3
TABLE,APPEND,1,POW,-0x153,0x1,REP3
TABLE,APPEND,1,POW,-0x0d1,0x1,REP3
TABLE,APPEND,1,POW,-0x06a,0x1,REP3
TABLE,APPEND,1,POW,-0x01f,0x1,REP3
# trigger the frequency change
TABLE,APPEND,1,HOLD,0x1,UPD
# loop back to create second pulse
TABLE,LOOP,1,-1,4,1
TABLE,APPEND,1,POW,0x0,0x1
```

Listing 9.9: Back-to-back shaped pulses with different frequencies

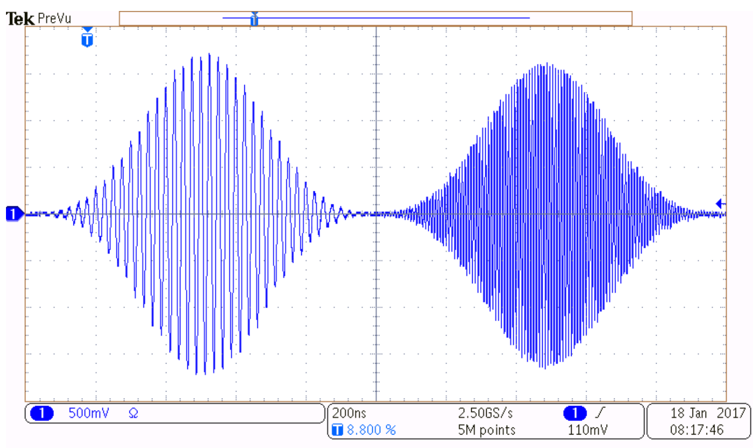


Figure 9.6: Two 1µs Gaussian pulses generated in advanced table mode with amplitude on the parallel bus. A serial instruction changes the frequency from 40 MHz to 120 MHz between the two pulses, allowing control of the frequency and phase of the second pulse.

9.9.3 Parallel frequency mode

The following example demonstrates control of the RF frequency with the parallel interface.

```
# set up table mode
MODE,1,TPA
TABLE,CLEAR,1
# use HSB for triggering
EXTIO,CTRL,1,HSB,AUTO
# change to FREQ mode and set the FM gain
TABLE,XPARAM,1,FREQ,15
# step through a number of frequencies
TABLE,APPEND,1,FREQ,20,0x5,IOA1H
TABLE,APPEND,1,FREQ,40,0x5,IOA1L
TABLE,APPEND,1,FREQ,80,0x5,IOA1H
TABLE,APPEND,1,FREQ,160,0x5,IOA1L
```

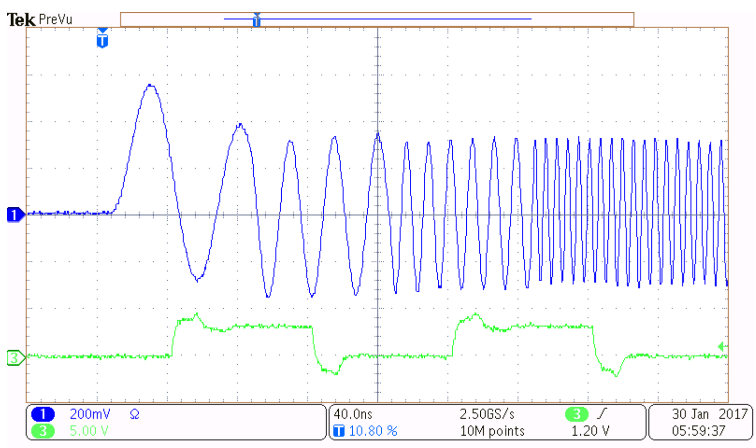


Figure 9.7: Example showing phase continuity with stepwise changes in frequency between 20 MHz and 400 MHz in advanced table mode.

A. Specifications

Parameter	Specification

RF characteristics	
Max output power	+34 dBm (normal operation) +36 dBm (sinc filter disabled)
Amplitude control	14-bit resolution
Frequency	20 to 400 MHz
Frequency control	32-bit resolution; 0.232831 Hz steps
Frequency stability	±1 ppm (0 to 50°C)
Phase	0 to 2π (16-bit resolution)
Phase noise	< -110 dBc @ 1 kHz
Signal to noise	> 100 dBc @ 30 dBm
Intermodulation and spurious	< -55 dBc
Channel crosstalk	< -70 dBc (mean), < -57 dBc (max)
Power on, RF off	< -100 dBm
Extinction ratio	> 110 dB in 10 Hz RBW

Analogue input	
Inputs	2 per channel (4 total)
Function	FM, AM, ϕ or analogue sampling
Sensitivity	±4 V
Bandwidth	10 MHz with 7th order anti-alias
Resolution	12-bit, 65 MHz sampling rate
Analog offset	±3.5 V
VGA gain	-8 to +24 dB

Digital input/output (per channel)	
RF on/off	TTL hardwired, positive logic only
Trigger input	TTL input to continue table execution
Shutter output	TTL output on DE15 connector
High-speed I/O	16 x TTL shared
TTL input high	2.2 V
TTL input low	0.6 V
Absolute max in	7.0 V
Absolute min in	-0.5 V

Table mode	
Min. step size	1 µs (basic table), 16 ns (advanced table)
Max. table length	8191 instructions per channel
FLASH memory	Non-volatile storage of up to 4 tables
Trigger options	Software, or TTL via DE15 connector
Channel sync	Independent, shared trigger, or fully synchronised (software configurable)

Mechanical & power	
Display	320x240 pixel colour LCD with backlight
Fans	4x temperature controlled fans
IEC input	90 to 264 Vac, 47 to 63 Hz
Dimensions	W×H×D = 250 × 79 × 292 mm
Weight	2 kg
Power usage	55 W

B. Firmware upgrades

From time to time, MOGLabs will release updates to the XRF firmware, which enable new functionality or address issues in the version which shipped with your device. This section contains instructions on how to apply firmware updates to your device.

WARNING: Do not attempt to communicate with the XRF while a firmware upgrade is being applied, and do not interrupt an upgrade (or factory reset) in progress.

B.1 Installing a firmware update

The recommended way to install updates is using the `mogrf` application from the *Update firmware* menu item. Running the application will display diagnostic information about your device (Figure B.1).

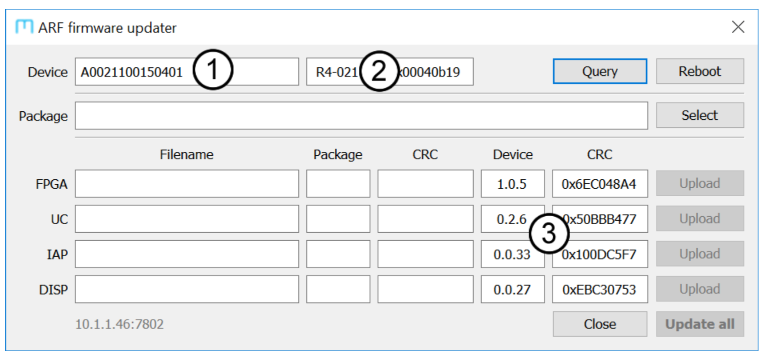


Figure B.1: The firmware update application connected to a unit, showing the serial number (1), model (2) and current firmware versions (3). Ensure that the model numbers are correct before continuing.

> **Note:** Make sure the automatically detected serial number, hardware revision and device model match the device. Uploading incorrect firmware can cause the unit to become non-operational and require return to the manufacturer to be fixed.

Update packages are available from the MOGLabs website and are loaded into the application by pressing the *Select* button. The firmware in the package is compared against the currently running version to determine which upgrades are required (Figure B.2).

Each component of the firmware is compared against the package version and colour coded as follows:

Green: Component matches package version and is up-to-date.

Yellow: Package contains an update which should be applied.

Magenta: Package contains an **older** version than currently installed. Installing this component will **downgrade** the firmware, which may be required if firmware conflict occurs.

Red: Installed version conflicts with package and may be damaged. Installation of package version is strongly recommended.

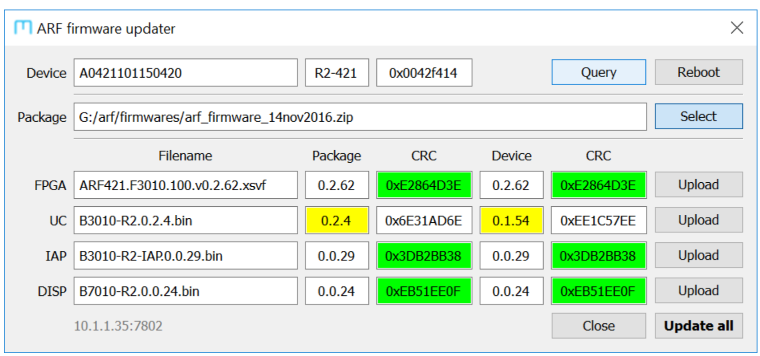


Figure B.2: The firmware update application. The versions running on the device are compared against the selected package, in this instance showing that an update is available for the UC (yellow) and the other components are already up-to-date (green).

Click on *Update all* to install all detected upgrades in sequence, otherwise individual components can be installed by clicking the *Upload* option next to each item.

The upload process proceeds through four stages:

1. The FLASH memory is erased to make room for the new image.
2. The data is uploaded to the device.
3. The device checks the data is received for consistency, to ensure the upload was successful.
4. The device is rebooted to load the new firmware.

The device will reboot after every individual component upgrade, to ensure the upload was successful before moving on to the next component.

Note: In order to upgrade the UC component, DIP3 must be set to ON, otherwise the upload will appear to succeed but the update will not be installed.

B.2 Factory reset

If a firmware upgrade fails and the device subsequently cannot boot, a factory reset (rebooting with DIP4 set to ON) must be applied. The device will then attempt to restore the configuration it was shipped with to restore operation. Once complete, the device will display a message on the front-panel LCD screen requesting that you return DIP4 to OFF and reset the device.

Note that all device settings will be overridden with factory defaults, including network settings, power limits, frequencies, calibrations, and so on. Ensure that relevant values are corrected after the reset.

Once the reset is complete, upgrade can be reattempted to gain access to newer features, or a different firmware can be applied. Please contact MOGLabs if you encounter any difficulties during firmware upgrade.

C. Command language

The protocol for communicating with an XRF is described in chapter 3, and the host software provided to interface with the unit is detailed in chapter 4.

> **Please note:** The command language is being continuously updated across firmware releases to improve functionality and add features. When upgrading firmware, please refer to the most recent version of the manual available at `http://www.moglabs.com`

C.1 Arguments

Most commands require a channel number `ch` (either 1 or 2), and accept a comma-separated list of parameters. Parameters shown in square brackets are optional, and most commands are treated as queries when called without a value.

All commands respond with a string that begins with either OK or ERR to indicate whether it was successful. It is **strongly recommended** that commands be checked for success.

Units can be specified for values associated with frequency, power, phase and time:

Frequency Hz, kHz, MHz (default)

Power dBm (default), dB, mW, W

Phase deg (default), rad

Time ns, us (default), ms, s

Calibrations are used to convert parameters to internal discretised values. Most commands will return a message that includes the *actual* value, which may differ from the *requested* value because of discretisation and/or parameter limits.

If required, values corresponding to internal representation can be specified directly using hexadecimal format with a 0x prefix. This is potentially useful for stepping through the discrete values that the AD9910 is capable of generating.

C.2 General functions

REBOOT, RESET Initiate a soft-reset of the device, reinitialising the microcontroller, FPGA and DDS. Note that all communications links will be immediately closed so there might be no response to this command.

INFO Report information about the unit.

VERSION Report versions of firmware currently running on device. Please include this information in any correspondence with MOGLabs.

TEMP Report measured temperatures and fan speeds.

VMON Report diagnostic monitoring information about power supplies.

C.3 Basic control

MODE `MODE,ch[,type]`

Controls the operational mode of the given channel; `type` is one of NSB, NSA or TSB, corresponding to normal operation, direct DDS control, and table mode respectively (see §1.1). Some options are only available in particular modes, as specified in the commands list.

Note: this command automatically switches off the output of the specified channel.

OFF/ON `OFF,ch[,mode]` `ON,ch[,mode]`

Enable or disable the RF output of the specified channel. The signal and amplifiers can be individually controlled, which allows for more rapid switching response (see §7.6). `mode` is one of:

SIG Turn off/on the RF signal only.

POW Turn off/on the RF high-power amplifier only.
Note that the amplifiers take 2 s to completely power on.

ALL Turn off/on both the RF signal and high-power amplifier (default).

STATUS `STATUS[,ch]`
Reports the current operational status of the specified channel, describing whether the signal and amplifiers are switched on. Returns an error if the device is not operational.

SLEEP `SLEEP,dt`
Pause microcontroller operation for `dt` milliseconds. Intended for use in simple scripts to wait for a short amount of time, e.g. for tables to finish execution.

C.4 Primary RF control

FREQUENCY `FREQ,ch[,value]`
NSB/TPA modes only. Set channel `ch` to specified frequency (between 20 MHz and 400 MHz). The actual frequency will be the closest available given the DDS clock frequency and the 32-bit resolution. The hexadecimal representation of frequency (the *frequency tuning word*) is converted to frequency by multiplying by 0.23283 Hz (that is, $10^9/2^{32}$).

The following examples all set the output to 100 MHz.

```
FREQ,1,100000000.0
FREQ,1,100MHz
FREQ,1,0x1999999A
```

POWER `POW,ch[,value]`
NSB/TPA modes only. Set channel `ch` to specified output power (or amplitude). The output power is computed using a factory calibration, and has 14-bit resolution. Requesting a value higher than the limit set with the `LIMIT` command results in an error.

The following examples will generate approximately 250 mW output power on 421-model devices:

```
POW,1,0x1000
POW,1,250 mW
POW,1,24 dBm
```

AUTOPOW `AUTOPOW,ch[,enable][,power]`

NSB mode only. XRF devices are able to compensate for the frequency dependency of the RF signal path using RF power meters that monitor the actual output power and feed back to the DDS amplitude control. This reduces the error between desired and actual output powers as the frequency is modulated.

`enable` is either 1 or 0 to activate or deactivate the feature respectively, and an optional `power` can be specified. If it is not specified, the current power as set using the `POW` command is used.

Cannot be used in combination with external amplitude modulation.

LIMIT `LIMIT,ch[,value]`

Defines absolute maximum RF power for given channel `ch`, with the same syntax as the `POW` command. This limit is applied within the FPGA to prevent damage to attached components, and cannot be exceeded even by modulation.

If the limit is set below the current power level, the current power is reduced to the limit. The default limit is +30 dBm (about 1 W) for amplified models and +7 dBm (5mW) for unamplified units.

```
LIM,1,33dBm
```

PHASE `PHASE,ch[,phase]`

NSB/TPA modes only. Set channel `ch` to specified phase. The following examples all set the phase to 180 degrees:

```
PHASE,2,0x7fff
PHASE,2,3.14159rad
PHASE,2,180deg
```

ADC `ADC,ch,F`

Returns the approximate measured output (forward) power of the specified channel in dBm, for diagnostic purposes. This measurement is taken after the power amplifiers (if installed) and will be affected by any active modulation settings.

DEBOUNCE `DEBOUNCE,ch[,off/on]`
The TTL digital DE15 control inputs can be debounced to allow operation with noisy signals, such as from a push-button or toggle switch. The `DEBOUNCE` command enables or disables the debounce filter for the selected channel (default: OFF).

SINCFILTER `SINC,ch[,onoff]`
Activate or deactivate the internal sinc filter of the AD9910 for the selected channel. Enabling sinc filter improves the frequency response of the DDS but reduces output power by approximately 3 dB. More details can be found in the AD9910 datasheet regarding the CFR1 register (default: ON).

PHRESET `PHRESET`
Simultaneously activates and then releases the phase accumulator reset of each DDS. This ensures the phase of the two channels is well-defined, even if there is a fixed phase-shift between them.

SYNC `SYNC[,onoff]`
Enable/disable phase synchronisation between the two channels. This ensure the two outputs are phase stable, for applications such as IQ-modulation. Once phase sync is achieved it will be maintained and the command should not be issued repeatedly.

C.5 Modulation settings

MSTAT `MSTAT,chan`
Reports the current modulation status for the specified channel. For example issuing `MSTAT,1` may return the response

PID: OFF, FM: SIF, PM: OFF, AM: PIF

indicating that both AM and FM are enabled, with fast (parallel) modulation on the AM.

MDN `MDN,ch,type[,onoff][,gain]`
Enables (or disables) modulation type of a given type (amplitude, frequency, or phase) for the selected channel. Dual modulation (§5.3) can be enabled by calling `MDN` twice.

type One of FREQ (frequency), AMPL (amplitude) or PHAS (phase).

onoff Can be ON or OFF. If omitted, `MDN` returns whether the specified modulation mode is active.

gain Optionally specify gain for this modulation type; equivalent to subsequently using the `GAIN` command.

GAIN `GAIN,ch,mdntype[,gain]`
Sets the modulation gain for the specified modulation type on the given channel. `gain` is a floating-point number with units (or a hexadecimal two's complement integer) that controls the depth of the modulation (see §5.2).

If `gain` is 0 then the modulation is disabled, and if `gain` is negative then the modulation action is inverted.

If `gain` is not specified, then the currently set gain is returned in both physical units and hexadecimal representation.

FMSPEED `FMSPEED,ch[,speed]`
Controls the bandwidth of frequency/phase modulation for the given channel. `speed` is either `FAST` resulting in 10 MHz bandwidth, or `SLOW` to yield 1 MHz bandwidth (see §5.3). Note that this command has no effect when only one modulation mode is enabled.

CALMOD `CALMOD,ch,type[,value]`
Return or set the zero offset calibration of the modulation input ADCs. The ADCs may return a small nonzero value when zero volts is applied to the input, causing a small unintended shift in the modulation parameter. The zero offset calibration corrects thus by subtracting the calibration from the measured value before performing modulation.

The parameter `type` is either FREQ or AMPL corresponding to the input being adjustment. If `value` is not provided, the current calibration is returned.

`value` is either a numerical value or the keyword AUTO, which will set the calibration to the currently measured value. It is recommended to connect a 50 Ω-terminator to the associated SMA connector when autocalibrating. Automatic calibration **should not** be performed with an unterminated input.

C.6 Digital ramp generator

The AD9910 contains a digital ramp generator (DRG) for linear parameter sweeps. This behaviour is controlled directly in NSA mode using the DDS registers (§C.12) in consultation with the AD9910 datasheet and using the commands below.

RAMPBIT `RAMPBIT,ch[,value]`
Enables or disables ramp functionality. Requires subsequent triggering of the ramp to execute using the `RAMPCTL` function before the ramp executes.

RAMPCTL `RAMPCTL,ch[,value]`
Activates the ramp in the ascending (1) or descending (0) direction. Note that if the ramp was programmed using `RAMPFREQ`, executing the ramp in the opposite direction will cause the output to reset to the start frequency, which is necessary before executing the ramp again. More advanced behaviour is possible using the DDS registers directly.

RAMPTRIG `RAMPTRIG,ch[,value]`
Sets the trigger mode of the ramp; permitted settings are:

EXT, HW The DDS waits for a hardware trigger on the CHx-OFF pin before executing the to ramp.

AUTO, CONT Execute the ramp continuously; once a ramp is completed it will reset and execute again. The ramp begins immediately after the command is issued.

INT, SW Resets the ramp to software control, which can be subsequently software-triggered by toggling `RAMPCTL`.

RAMPFREQ `RAMPFREQ,ch,startfreq,endfreq,timestep,freqstep`
Generates a frequency ramp from `startfreq` to `endfreq` with step duration `timestep` and frequency step size `freqstep`. Unless an external trigger has been set using `RAMPTRIG`, the ramp will begin immediately after programming. The `FREQ` command cannot be used when `RAMPFREQ` has been activated.

Note that issuing a second `RAMPFREQ` command will stop any previously configured ramp and replace it. However, ramps can be joined together in table mode.

Example: `RAMPFREQ,1,400MHz,20MHz,0.1ms,10kHz`

startfreq Start frequency of the ramp.

endfreq End frequency of the ramp.

timestep Duration of each frequency step. The mimimum timestep is 4 nS and maximum is 262 μs.

freqstep Size of each frequency step.

RAMPHOLD `RAMPHOLD,ch[,value]`
Pauses or unpauses a currently executing parameter ramp. Should only be called on a ramp in progress, and not as a software-timed trigger for ramp execution.

C.7 Clock reference

The DDS devices operate from an internal clock (SYSCLK) at frequency $f_{SYSCLK} = 1\,GHz$, which is derived either from the oven-stabilised crystal oscillator at 20 MHz (internal mode) or provided via the SMA input labelled CLK IN (external mode).

Each DDS multiplies this reference clock and stabilises with an internal phase-locked loop (PLL) to generate output frequencies across the 20 – 400 MHz range. This provides flexibility at the expense of a small increase in phase noise. In applications where minimal phase noise is critical, a stable 1 GHz reference should be provided that directly clocks the DDS without the need for frequency multiplication.

CLKSRC `CLKSRC[,source][,ppln]`

Query or set the current clock source. `source` is either `INT` to use the internal 20 MHz oscillator, or `EXT` to use the reference provided to the CLK IN connector on the back-panel, compatible with a number of standard reference frequencies (such as a 10 MHz GPS clock).

When using external reference, the `ppln` clock multiplier value must be provided. This is 1 GHz divided by the external clock frequency, rounded to the nearest integer. Wherever possible, the external clock frequency should be chosen to ensure no remainder to prevent accumulation of timing errors.

If the clock source is 1 GHz, then `ppln` must be set to zero. This disables the PLL and improves the phase-noise of the RF output. Except for this special case, valid ranges for `ppln` are [12,127], which means the reference must be the range 7.87 MHz to 83.3 MHz.

> **Note:** The external reference clock must have power between +3 dBm and +10 dBm. The output of the `CLOCK` command should be checked after using the `CLKSRC` command to ensure that synchronisation was successful. **Never operate the XRF in external clock mode without providing a valid reference clock, as undefined behaviour can result.**

CLOCK `CLOCK,ch`

Measures the current DDS system clock frequency for the specified channel, as measured by the FPGA over a one second window. This should return 1000 MHz, indicating that the system is correctly synchronised to the clock source. If it does not, use `CLKDIAG` to determine whether the PLL has achieved a lock to the reference.

Drift between the internal and external clocks can result in small shifts in this measurement.

CLKDIAG `CLKDIAG`

Reports diagnostic information about the status of the internal clocks. In regular operation, the Reference, System and DDS clocks should always report `OK LOCKED`.

Failure of any of the PLLs to lock can result in undefined behaviour, and is typically a result of the reference clock having incorrect amplitude or excessive phase noise.

C.8 Table mode

Table mode gives access to the powerful sequencing functionality of the XRF/XRF devices (chapter 8). XRF devices also have access to advanced table mode (chapter 9), which is controlled using the same `TABLE` commands. See section 8.2 for details on the table entry structure.

ARM `TABLE,ARM,ch`

Loads the table into the FPGA for execution, typically taking $\sim 100\,\mu s$. The table then begins execution upon receiving a software trigger (`TABLE,START`) or hardware trigger on the DE15 connector (§7.1).

Note: The table length *must* be defined before issuing `ARM` or `START`; failure to do so may result in undefined behaviour. The convenience functions (e.g. `TABLE,APPEND`) automatically update the table length.

START `TABLE,START,ch`
Provides a software trigger to initiate table execution. Calls `TABLE,ARM` if the table is not already ready for execution, which can cause a short delay before output appears.

Early 421-series models may encounter an error that the amplifiers are disabled when using this command. The recommended solution is to manually power-on the amplifiers earlier using either `ON,1,POW` or `TABLE,ARM` before the `TABLE,START` command.

STOP `TABLE,STOP,ch`
Terminates an executing table at the end of the current step. Note that the RF output will remain *on*, holding the instruction being executed when the command was received, until the table is rearmed or the output disabled with the `OFF` command.

REARM `TABLE,REARM,ch[,on/off]`
Enables/disables the automatic re-arming (loading) of the table upon completion such that it can be started again without using the `ARM` command. Table will then begin executing upon a software or hardware trigger.

RESTART `TABLE,RESTART,ch[,on/off]`
Enables/disables an automatic software-controlled restart of the table upon completion. Automatically enables `REARM`. Table will then begin executing upon a software or hardware trigger.

STATUS `TABLE,STATUS,ch`
Reports the current execution status of the table.

ENTRIES `TABLE,ENTRIES,ch[,num]`
Defines the last table entry number for the given channel. Failing to correctly set the number of entries can result in undefined behaviour.

LENGTH A synonym for `TABLE,ENTRIES`

ENTRY `TABLE,ENTRY,ch,num[,freq,ampl,phase,duration][,flags]`
Sets (or returns) the currently loaded table entry `num` of channel `ch`.

Entry numbers start at 1 and tables can contain up to 8191 entries. The structure of this command is detailed in §8.2.

Example: `TABLE,ENTRY,2,1,800MHz,0x1500,0x0000,10us`

HEXENTRY `TABLE,HEXENTRY,ch,num`
Queries the specified table entry, returning the internal hexadecimal representation of the associated frequency, amplitude and phase.

APPEND `TABLE,APPEND,ch,freq,ampl,phase,duration[,flags]`
Inserts the specified entry at the end of the table and increments the `TABLE,ENTRIES` counter.

INSERT `TABLE,INSERT,ch,num,freq,ampl,phase,duration[,flags]`
Insert the table entry at the specified index, shifts all subsequent entries down, and increments the `TABLE,ENTRIES` counter.

DELETE `TABLE,DELETE,ch,num`
Deletes the specified table entry, shifting all subsequent entries up and decrements the `TABLE,ENTRIES` counter.

RAMP `TABLE,RAMP,ch,param,start,stop,duration,count`
Creates a linear ramp in `param`, which is one of `FREQ`, `AMPL` or `PHAS`, from `start` to `stop` in `count` steps, each lasting `duration`. In simple table mode this generates `count` table entries, whereas in advanced table mode it generates up to three.

SAVE `TABLE,SAVE,ch,slot`
Save the current table in a FLASH memory slot, where `slot` is a number from 1 to 4. Uploading tables in binary format is also possible using the `mogrf` host software.

LOAD `TABLE,LOAD,ch,slot`
Loads the table from the specified slot in FLASH memory to the designated channel.

COPY `TABLE,COPY,src,dest`
Copies the table data from the `src` channel to the `dest` channel.

SYNC `TABLE,SYNC[,onoff]`
Enable or disable table synchronisation mode (see §8.8), which sets CH1 as a master table and CH2 as a slave table. Both tables then execute simultaneously when CH1 is started.

TRIGSYNC `TABLE,TRIGSYNC[,onoff]`
Enable or disable DE15 trigger synchronisation between the two channels, which causes CH2 to take its DE15 trigger from CH1. This option is a more precise equivalent of wiring the two inputs together.

NORESET `TABLE,NORESET,ch[,onoff]`
Disables table-mode phase reset for the specified channel. Normally the DDS phase accumulator is reset when the table is started, ensuring that the output waveform has the same starting phase for every execution. However, this requires switching off for a few microseconds while the DDS is reset, which is undesirable for some applications. Using `NORESET` will ensure the output stays on.

XPARAM `TABLE,XPARAM[,mode][,gain]`
TPA mode only (XRF). Set the parameter to be modified on the parallel bus in advanced table mode. The parameter `mode` is one of `FREQ`, `PHASE`, `POWER` or `AMPL`. If `mode` is `FREQ` then an optional FM-gain can be specified (§9.7), otherwise defaults to 15.

DUMP `TABLE,DUMP,ch`
Transmits the raw table data for the specified channel in binary form. The command should not be used from an interactive terminal. The first 4 bytes of the response are payload size, followed by that many bytes of data. The expected payload is $16 \times (N + 1)$ bytes, where N is the number of table entries.

Note that raw table data may be incompatible between different firmware versions.

UPLOAD `TABLE,UPLOAD,ch,nbytes`
Upload binary table data for channel `ch`. The data should first be downloaded with `TABLE,DUMP`. `nbytes` is the byte length of the table.

C.9 PID feedback

The XRF has an integrated PID-controller that can feed back to the amplitude, frequency or phase of the RF output in response to an error signal (see chapter 6).

ENABLE `PID,ENABLE,ch,mode`
Enables the PID controller for the given channel in the specified `mode`, which is one of FREQ, AMPL or PHAS.

DISABLE `PID,DISABLE,ch`
Disables the PID controller for the given channel.

STATUS `PID,STATUS,ch`
Reports the active status of the PID controller and whether saturation occurred. The saturation flag is reset by this query.

VALUE `PID,VALUE,ch`
Returns the output of the PID servo controller as a floating point value in the range [-1,1]. Intended for sanity-checking the servo performance and diagnosing saturated controller issues.

RATE `PID,RATE,ch[,div]`
Sets the rate at which the analogue control signal is digitised and processed. The actual sample rate is 62.5 MHz/2^{div}, where `div` is an integer in [0,7].

OFFSET `PID,OFFSET,ch[,val]`
Applies an analog DC-offset to the input signal to permit locking to a non-zero set-point voltage. If specified, the voltage `val` must be within ±3.5 V.

VGA `PID,VGA,ch[,val]`
Controls the variable gain amplifier in the analog signal processing path, to improve the signal-to-noise ratio of the digitised error signal. The gain value is between -8 dB and +24 dB, and the VGA is disabled by setting it to 0 dB.

AVERAGE `PID,AVERAGE,ch[,on/off]`

Enables/disables averaging of the input control signal. The digitisers operate at much higher rates than the PID sampling rate. With averaging on, the extra measurements contribute to an average signal which is reset at the sampling rate. If averaging is off, the additional measurements are discarded.

DCBLOCK `PID,DCBLOCK,ch[,on/off]`

Enables/disables *DC block* in the digital signal processing chain. The *DC block* is equivalent to a series capacitor in the input control signal prior to the PID controller.

INVERT `PID,INVERT,ch[,on/off]`

Inverts the controller action for the given channel. Typically used to correct for whether the signal is taken from the diffracted or undiffracted beam.

GAIN `PID,GAIN,ch,name[,value]`

Sets the PID gain constant for `name`, which is one of `P` (proportional), `I` (integral) or `D` (derivative), as described in §6.2.

`value` is a floating point number in the range [0,1]. The derivative gain can be negative, which inverts the sense of the derivative action with respect to the other components, and may be beneficial in some applications that require phase lead or lag.

C.10 External IO functions

Controls the behaviour of the digital IO lines on the DE15 and 30-pin high-speed connectors as described in chapter 7.

Command structure: `EXTIO,fn,ch,pin,[parameters]`

ENABLE Enables the main or default function of the pin.
Example: `EXTIO,ENABLE,1,OFF`

The `pin` parameter can be one of

HSBANK Enables/disables the high-speed bank.

OFF Enables/disables the CHx-OFF input for fast external switching.

DISABLE Disables the main function of the specified pin, using the same syntax as the `EXTIO,ENABLE` command.

RESET Resets the pin to the default start-up state.
Example: `EXTIO,RESET,1,HSBANK`

HSBANK Returns control of all high-speed pins in specified port to the microcontroller, with outputs disabled (tri-stated).

OFF Disables fast external switching of RF.

HSn Sets the specified high-speed pin into manual mode and outputs digital LOW (does not tri-state the output).

MODE Sets the mode for the specified pin and enables the pin if necessary.
Syntax: `EXTIO,MODE,1,pin[,mode]`

If `pin` is OFF, then `mode` must be one of

LATCH The OFF pin is enabled and once tripped, the output stays off until reset (*interlock* mode).

TOGGLE The OFF pin directly specifies the state of the RF switch, with logic HIGH = RF off, LOW = RF on. If the pin is physically disconnected, the switch will be turned off and no output will be observed.

If `pin` is HSB, then `mode` must be one of

READ Configure the 8 pins of the bank for input.

WRITE Configure the 8 pins of the bank for output.

This command does not apply to any other pins.

CONTROL Sets or returns the current control mechanism for the specified pin.
Syntax: `EXTIO,CONTROL,ch,pin,mode`

The parameter `mode` is one of:

AUTO Set the pin to automatic (FPGA) control, allowing it to be controlled by `TABLE` mode commands.

MAN Set the pin to manual (microcontroller) control, allowing the `EXTIO,WRITE` and `EXTIO,READ` commands to be used.

If `pin` is HSBANK, the command sets the mode of all 8 pins in the bank. In this case, `mode` is either AUTO or MAN, or an unsigned 8-bit integer where each bit corresponds to setting the associated pin into automatic (1) or manual (0) mode.

CTRL Synonym for `CONTROL`

WRITE Sets the control mechanism to MANUAL for the specified pins and outputs the supplied `data`.
Structure: `EXTIO,WRITE,ch,pin[,data]`
Example: `EXTIO,WRITE,1,HS1,1`

READ Returns the current output value of the specified pin(s). If the pin is in automatic mode, the value is undefined.
Example: `EXTIO,READ,1,OFF` to read the state of the OFF pin of the DE15 connector, or `EXTIO,READ,1,HSBANK` to read the entire high-speed bank.

COUNTER Control the counter associated with the specified pin. Syntax and functionality of the command is described in §7.7.

C.11 Configuration settings

SET, GET Set and report EEPROM configuration values. Each `set` command described below has a corresponding `get` command to report the relevant parameter.

ipaddr `SET,ipaddr,"xxx.xxx.xxx.xxx"`
Set IP address based on decimal dotted-quad string (for example "10.1.1.180"). Note that the double-quotes are part of the syntax and must be included to delimit the IP address string.

ipmask `SET,ipmask,"xxx.xxx.xxx.xxx"`
Set IP mask based on dotted-quad string (for example "255.255.255.0").

ipgw `SET,ipgw,"xxx.xxx.xxx.xxx"`
Set IP gateway based on dotted-quad string (for example "10.1.1.1").

ipport `SET,ipport,port`
Set the TCPIP port number for device communication.

dhcp `SET,dhcp,onoff`
Enable or disable DHCP.

userid `SET,userid,"username"`
Deprecated. Set the username used for legacy web-based firmware upload.

password `SET,pass,"password"`
Deprecated. Defines the password used for legacy firmware upload.

C.12 Direct DDS control

For direct control the the DDS chips, NSA mode is provided. This allows direct access to the DDS registers, but only a limited subset of commands are unavailable.

> **WARNING:** Using these commands bypasses all safeguards and can result in undefined behaviour or damage to connected devices. Their use is intended for **advanced users only**, in close consultation with the AD9910 datasheet.

PSELECT `PSELECT,ch[,num]`
Trigger the DDS to activate the specified single-tone profile (STP), which is immediately output. Once activated, the profile can be configured using the `PROFILE` command. `num` is an integer in [0,7].

PROFILE `PROFILE,ch,frequency,amplitude,phase`
Define the frequency, amplitude and phase for the currently activated profile, as set by the `PSELECT` command.

Example: `PROF,1,80000000.0,0x2000,0x0000`

DDS `DDS,ch,ddsreg[,data]`
Direct access to the internal registers of the DDS chips. If `data` is not supplied, a read operation is assumed and `data` is returned. Single tone profiles (STP0 to STP7) can only be read if active; each can be made active with the `PSELECT` command.

Available registers (`ddsreg`) are:

CFR1 Control function register 1 (32-bit)

CFR2 Control function register 2 (32-bit)

CFR3 Control function register 3 (32-bit)

AUDC Auxiliary DAC control register (32-bit)

RATE I/O Update rate register (32-bit)

FREQ Frequency tuning word (32-bit)

PHAS Phase offset word (16-bit)

AMPL Amplitude scale factor (32-bit)

SYNC Multichip sync (32-bit)

RLIM Digital ramp limit (64-bit)

RSTP Digital ramp step size (64-bit)

RRTE Digital ramp rate (32-bit)

STP[0-7] Single tone profile 0-7 (64-bit)

RAMW RAM word register (32-bit)

DINIT `DINIT,ch`
Executes the startup initialisation script for the given channel. The command allows a re-initialisation of the DDS control registers to their boot-time defaults, which is useful in NSA mode if the device is put into an undefined state.

OSK `OSK,ch,[value]`
NSA mode only. Sets the DDS OSK (output shift keying) pin to the given value. See the AD9910 datasheet for more information.

D. Code examples

The following simple examples demonstrate how to communicate with the XRF over ethernet in several languages, using the bindings provided by MOGLabs. Further examples are available from the MOGLabs website.

D.1 python

Communication is handled by a “device” class, which provides convenience functions for sending commands and queries.

```
#---------------------------------------------
# XRF python example, (c) MOGLabs 2016
#---------------------------------------------
from mogdevice import MOGDevice

# connect to the device
dev = MOGDevice('10.1.1.23')
# print some information
print('Device info:', dev.ask('info'))

# example command: set frequency
dev.cmd('FREQ,1,100MHz')
# example query: check frequency
print('CH1 Freq:', dev.ask('FREQ,1'))
# some queries can return dictionaries
print('Temperatures:', dev.ask_dict('TEMP'))
# other queries respond with binary data
tbl = dev.ask_bin('TABLE,DUMP,1')
print('Binary table:', len(tbl))
# close the connection
dev.close()
```

The next example shows how to construct a Gaussian pulse using `numpy` and the `MOGDevice` class, showing how easy it is to generate arbitrary waveforms. The resulting waveform is shown in Figure D.1.

```
#-----------------------------------------
# XRF Gaussian pulse example, (c) MOGLabs 2016
#-----------------------------------------
from mogdevice import MOGDevice
import numpy as np
# connect to the device
dev = MOGDevice('10.1.1.45')
print('Device␣info:', dev.ask('info'))

# construct the pulse
N = 200
X = np.linspace(-2,2,N)
Y = 30*(np.exp(-X**2)-1) # -30 to 0dBm

dev.cmd('MODE,1,TSB') # set CH1 into table mode
dev.cmd('TABLE,ENTRIES,1,0') # clear existing table
for y in Y: # upload the entries
    dev.cmd('TABLE,APPEND,1,100,%.2f,0,5'%y)
print(dev.cmd('TABLE,ARM,1')) # ready for execution
```

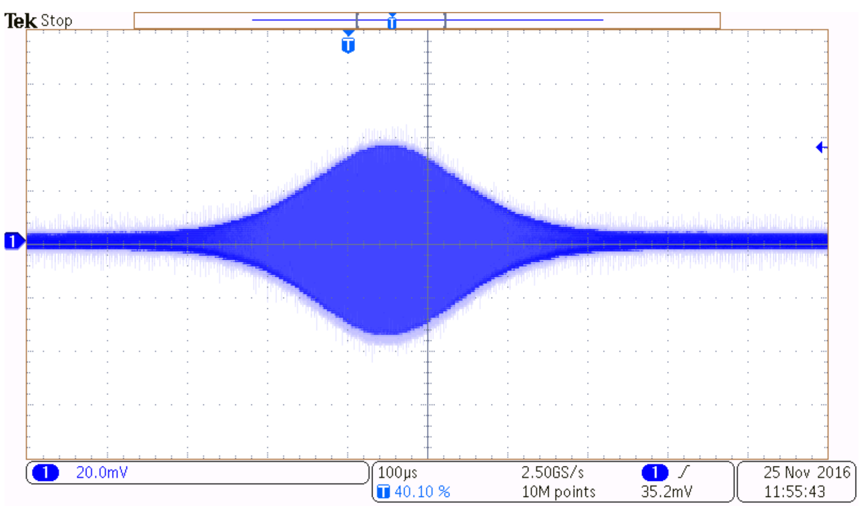


Figure D.1: Pulse with a Gaussian envelope, created in table mode using the example python code.

D.2 matlab

Similar to the python bindings, a class is provided to make controlling the XRF easy using matlab. The listing below demonstrates how to create a simple table that produces a pulse with a Gaussian envelope.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% XRF MATLAB example, (c) MOGLabs 2017
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% create a device instance
dev = mogdevice();
% example: connecting by ethernet
dev.connect('10.1.1.31');
% print some information about the device
disp(dev.ask('INFO'));

% create a gaussian envelope
N = 500;
pulse = exp(-(8*((0:N)/N - 0.5)).^2);
plot(pulse)
% convert mW to dBm
pulse = 10*log10(pulse);

% upload gaussian pulse in simple table mode
dev.cmd('MODE,1,TSB');
dev.cmd('TABLE,CLEAR,1');
disp('Uploading table...')
for i=1:length(pulse)
    % we can use printf notation when sending commands
    dev.cmd('TABLE,APPEND,1,100MHz,%f dBm,0,1us',pulse(i));
end
disp('Done')
dev.cmd('TABLE,ARM,1')
dev.cmd('TABLE,START,1')

% close the connection
delete(dev);
```

D.3 `LabVIEW`

The `LabVIEW` drivers provided make use of `NI-VISA` to provide a unified interface over both ethernet and USB. They perform automatic error checking, and are compatible with `LabVIEW`-2009 and later editions.

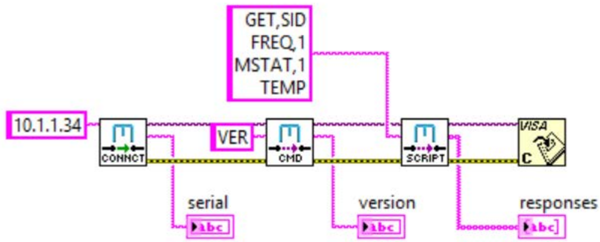


Figure D.2: Example `LabVIEW` program that connects to an XRF unit, and performs a number of queries

When using these drivers, it is strongly recommended that the automatic session close option be enabled, to prevent communications problems if the VI aborts or is interrupted. The option is found in Tools→Options→Environment→Automatically close VISA sessions.

E. Troubleshooting

The following is general troubleshooting advice for unexpected behaviour. Please contact MOGLabs (support@moglabs.com) for further assistance

The command `ADC,ch,F` provides a useful diagnostic for checking the output power from the device without needing to connect test equipment. The power out of the specified channel is measured using a directional coupler after the switch and amplifiers. The true output power should be within a few dB of the reading.

E.1 Computer interface

Advice for resolving common issues is contained in the *Drivers and Connection Guide* available from the *Support* section of our website at `www.moglabs.com`. The guide provides instructions for connecting the XRF to a computer by Ethernet or USB.

E.2 Unexpectedly high output power

This may occur when AM is enabled but the AM input port is left disconnected, which causes the amplitude to rise by the AM gain. Ensure that modulation is disabled when the input port is disconnected. Note that the output power will be less than the limit power (see `LIM` command) even when modulation is enabled, but may be circumvented in advanced modes (NSA or TPA).

E.3 Incorrect output frequency

This typically only arises when FM is enabled or the reference clock provided to CLK IN connector is not supported. Note that a disconnected modulation input is not the same as providing a 0 V input signal.

Confirm that FM is disabled and check that CLKSRC is set to INTERNAL. If synchronisation to an external clock is desired, confirm that the clock satisfies the requirements in §C.7.

E.4 Fan stall error

The XRF contains several fans to manage the thermal load of the high-power RF amplifiers. The fan speeds are temperature-controlled, and they will only turn on if required. Failure of the fans will disable the RF outputs to prevent thermal damage to the unit, and an error message will be displayed. If that occurs, check for obstruction of the fans and for anything that could restrict airflow.

E.5 Unsupported load error

The XRF is designed to drive a 50 Ω load, but is capable of driving mismatched loads without damage. However, driving a (near) short-circuit will draw significantly more power and reduce the lifetime of the amplifiers, so the RF output will be shutdown in this situation. Check the resistance of the load and confirm it is not damaged or accidentally shorted.

Note that when the sinc filter is disabled (see §C.4) this error may occur when driving high power at low frequency (< 25 MHz).

E.6 No RF output power

There are a number of interlock and safety features that can cause the output to be disabled. These scenarios should result in error messages being produced, either on the display or as a response to the command instructing the output to be activated.

The front-panel LEDs indicate whether the device expects that output should be generated. If the LEDs do not light up as expected, there is likely an error condition. Check for errors by connecting with the `mogrf` software and turning on the output through the application.

If there is no error, verify that the CHx-OFF functionality is disabled. In particular, the LATCH feature may cause the output to be turned off immediately after enabling it if the TTL input is left floating.

If the LEDs are lit but no output is observed, verify that AM and CHx-OFF are disabled, as these can cause the output power to be zero despite the apparent requested power being nonzero.

In the case where modulation is disabled, or the output power is ~ 30 dB lower than expected, verify that the power amplifiers are enabled by checking the supply current with the VMON command or opening the Diagnostics window in `mogrf`. Contact MOGLabs for further assistance

E.7 CHx-OFF has no effect

The CHx-OFF functionality for generating pulses is disabled by default (§7.4) and must be enabled through the interactive menu system, the host software, or via a relevant EXTIO command.

E.8 FPGA PROGRAM or VERSION error

This error can occur if a firmware update is aborted, or a factory reset is performed. Please complete a firmware upgrade to the latest version to resolve the issue.

F. Low-frequency output

The XRF provides RF generation across the range 20–400 MHz, but some applications may require frequencies outside this range. It is possible (but not recommended) to access the full range of the integrated DDS by *unlocking* the frequency limits.

The behaviour of the integrated RF power amplifiers is not specified outside the intended operating frequency range, and the **device performance specifications no longer apply**. In particular, the power calibration, maximum output power, distortion, current draw, and monitor outputs will be adversely affected.

WARNING: MOGLabs is not responsible for device behaviour when operated outside the specifications. Calibrate performance on a 50 Ω load before use.

The output power of the device drops rapidly below 20 MHz, and device calibration should not be relied upon. Typical variation in output power is shown in Figure F.1. The amplifier current draw can also increase, which can result in a "load error" shutdown, implying the output power must be reduced at that frequency.

F.1 Unlocking low-frequency output

Due to the variation in behaviour, operation outside the specified frequency range must be explicitly unlocked using the `UNLOCKFREQ` command on a **per-channel** basis. The command can be issued using the *Device Commander* as part of the `mogrf` application.

The `UNLOCKFREQ,ch[,mode]` command can query or set the operational mode of the specified channel `ch`, where `mode` is one of `NORMAL`, `UNLOCKED` or `LF-TAP`, as shown in the example below.

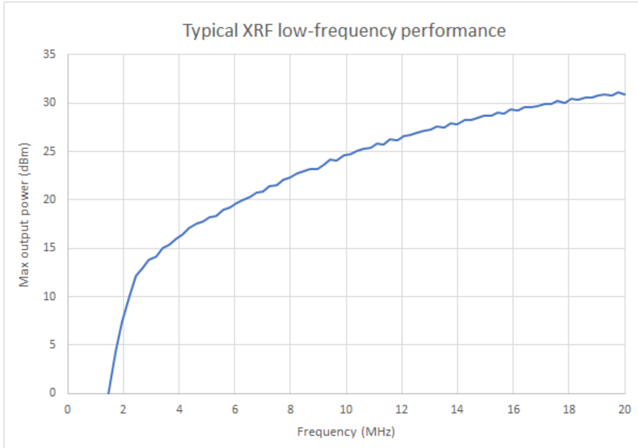


Figure F.1: Example frequency sweep showing variation in output power below 20 MHz at +33 dBm requested power.

```
UNLOCKFREQ,1,UNLOCKED
UNLOCKFREQ,2,NORMAL
```

Listing F.1: Demonstration of using the UNLOCKFREQ command to set CH1 to unlocked mode and CH2 to normal mode.

F.2 Low-frequency tap

An alternative option is to bypass the power amplifiers and tap the RF output directly out of the DDS. This avoids the behaviours related to the amplifier performance at the expense of much lower output power (typically up to -5 dBm) and no output monitoring.

The RF tap is provided on an SMA connector internal to the unit (Figure F.2), which can provide low-frequency generation down to near DC. In this mode the associated external rear-panel SMA outputs cannot be used, and the user must attach cables as appropriate to route the RF from each of the tap outputs to external connections outside the unit.

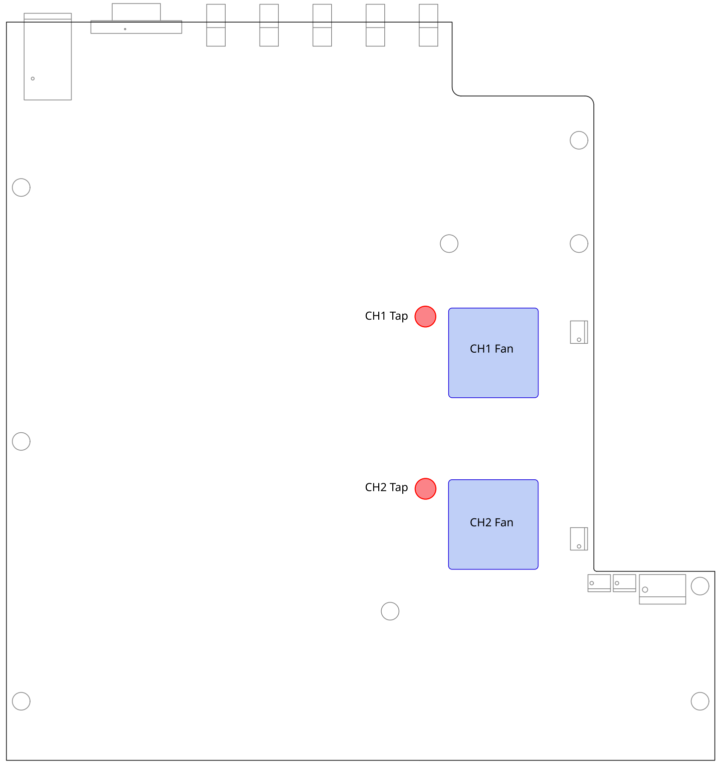


Figure F.2: Location of the low-frequency tap SMA connectors on the mainboard PCB. The outputs must be activated using the `UNLOCKFREQ` command to be used.

The tap is activated using the command `UNLOCKFREQ,ch,LF-TAP` where `ch` is the channel of interest (1 or 2). **The unit must be power-cycled after changing the tap mode** to ensure consistent operation.

When operating in this mode the power calibration is only approximate, so care should be taken when setting the output power and it should be verified before attaching an external amplifier. Indicative performance of the output is shown in Figure F.3, but this should be individually calibrated at the frequencies and powers of interest.

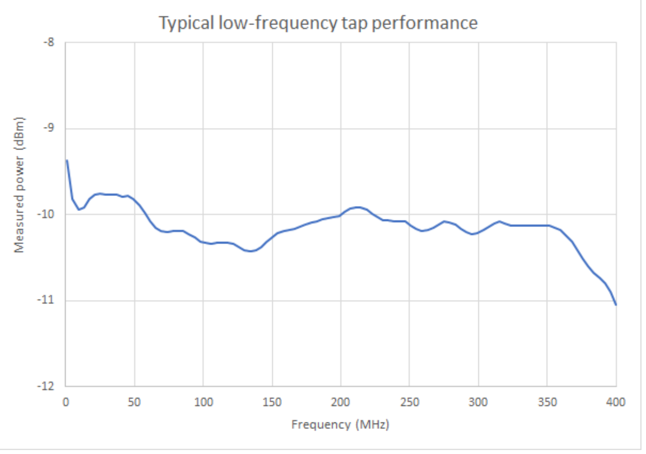


Figure F.3: Example frequency sweep showing output power of low-frequency output at −10 dBm requested power.

MOG Laboratories Pty Ltd
49 University St, Carlton VIC 3053, Australia
Tel: +61 3 9939 0677 info@moglabs.com