EV12AQ600-ADX-EVM
EV12AQ605-ADX-EVM

User Guide

Author(s): Teledyne e2v
Document ID: 18-2105
Classification: Public
Revision: F
Print date: 2019-05-28

Contents

Document History

Revision	Date	Section	Description	Author
F		1.1, 4, 4.6	Added abbreviation, fixed spelling, fixed error in Table 6.	TSPD
E		5.2	Updated description of factory calibration.	TSPD
D		5.1	Updated screenshots.	TSPD
C	2019-04-29	All	Updated with EV12AQ605-ADX-EVM.	TSPD
B	2018-12-11	5.2	Added documentation of calibration.	TSPD
		5.1	Added documentation F11 button.	TSPD
		2.1	Changed max frequency to 6400 MHz.	TSPD
A	2018-04-26	-	First release	TSPD

1 Introduction

This document presents the user guide of the EV12AQ600-ADX-EVM and EV12AQ605-ADX-EVM — a data acquisition platform built around *EV12AQ600* / *EV12AQ605*, analog-to-digital converters from Teledyne e2v and the *ADX* interleaving correction IP from Teledyne SP Devices.

1.1 Definitions and Abbreviations

Table 1 lists the definitions and abbreviations used in this document and provides an explanation for each entry.

Table 1: Definitions and abbreviations used in this document.

Term	Explanation
ADC	Analog-to-digital converter
AFE	Analog front-end
API	Application programming interface
EVM	Evaluation module
FPGA	Field-programmable gate array
GSPS	10^9 samples per second.
IP	Intellectual property
LSB	Least significant bit
MSB	Most significant bit
PCB	Printed circuit board
PLL	Phase-locked loop
SDK	Software development kit
SFDR	Spurious-free dynamic range
SMA	sub-miniature version A
SNDR	Signal-to-noise and distortion ratio
SNR	Signal-to-noise ratio
TILD	Total interleaving distortion
THD	Total harmonic distortion

2 Features

The EV12AQ600-ADX-EVM and EV12AQ605-ADX-EVM features the EV12AQ600 / EV12AQ605, 12-bit quad channel ADCs with an embedded cross-point switch. Each channel may run at 1.6 GSPS and the

interleaving configurations further increase the guaranteed conversion rate to 3.2 GSPS in two-channel mode and 6.4 GSPS in one-channel mode.
The ADC also introduces the *ESIstream* serial link protocol. The receiver IP residing in the FPGA is written in Verilog and may be used as a reference design.
In one- and two-channel mode, the EVM also features the ADX interleaving correction IP which compensates for static and frequency-dependent mismatch between the time-interleaved cores.

2.1 Specification

Table 2 lists the specification of the EV12AQ600-ADX-EVM & EV12AQ605-ADX-EVM.

Table 2: Specification of the EV12AQ600-ADX-EVM & EV12AQ605-ADX-EVM.

Term	Min	Typ	Max	Unit
Input range (analog inputs)		2.25		Vpp
Clock frequency	$\approx$ 4000*		6400	MHz

* See Section 7

3 Getting Started

This section provides information on how to get up and running with the EVM. First, install the SDK as described in Section 3.1. Second, connect the peripheral signals to the EVM; the connections are specified in Section 3.1.1. Finally, start the device and collect some data; the collection procedure is described in Section 3.2.

3.1 SDK Installation

The Software Development Kit (SDK) contains the ADQAPI, drivers, examples and documentation required to successfully interface with the EVM. On Microsoft Windows, the SDK is installed by running

```
ADQ-setup_rXXXXX.exe
```

where `XXXXX` is the version number. After the installation the example code is located in

```
<Path to installation directory>/C_examples/
```

and the documentation in

```
<Path to installation directory>/Documentation/
```

3.1.1 Peripheral Connections

The EVM has seven edge-mounted SMA connectors and six upwards-facing SMA connectors. Each connector is marked with a label printed on the PCB. Table 3 lists the peripheral connections, identified by their label, and provides a short description of their respective function. Please refer to Section 4 for more detailed information about a particular feature.

Table 3: Peripheral connections on the EVM.

Connector	Description
EXTCLK	External clock reference input
CH A	Analog input channel A
CH B	Analog input channel B
CH C	Analog input channel C
CH D	Analog input channel D
TRIG	External trigger input
CLK	External clock input
SYNCO P/N	SYNC output (differential)
CLKOUT P/N	ADC clock output (differential)
EXTSYNC P/N	External SYNC input (differential)

3.2 Collecting Data

The data acquisition and control is performed through ADCaptureLab, see Section 5.

4 Detailed System Description

This section provides in-depth information about a few selected aspects of the EVM, e.g. how the ADC interleaving mode is configured for the different channel combinations, clock configuration and trigger options.

4.1 Status LEDs

Table 4 lists the status LEDs and their function.

4.2 Interleaving Configuration

The EVM has four main modes of operation: one-channel (4-ways interleaved), two-channel (2-ways interleaved) and four-channel (no interleaving). Switching between the modes requires a soft reset of

Table 4: PCB Status LEDs.

LED	Description
FW OK	FPGA boot sequence completed successfully
FW ERR	FPGA boot sequence failed
FPGA INIT	FPGA programming start
FPGA DONE	FPGA image loaded successfully
1 CH	One-channel mode active
2 CH	Two-channel mode active
4 CH	Four-channel mode active

the EVM, i.e. no power cycling is required but the EVM will have to be reacquired by the host computer. Refer to Section 5 for information on how to perform the switch from ADCaptureLab.
The EVM indicates which mode is active by lighting up the corresponding LED located at the top of the PCB. Additionally, once the ADC cross-point switch is configured during the boot-up process, the corresponding LEDs will light up indicating which analog inputs are active.
The cross-point switch and clock interleaving modes are referred to using the values from the EV12AQ600 product specification [1].

4.2.1 One-channel Mode

In one-channel mode, the cross-point switch is placed in mode '0', meaning *ADC input 0* is fed to all cores. Additionally, the clock interleaving mode is set to '0', meaning all cores are interleaved. A simplified block diagram of the configuration is presented in Fig. 1.

4.2.2 Two-channel Mode

In two-channel mode, the cross-point switch is placed in mode '2', meaning *ADC input 0* is fed to cores A and B and *ADC input 3* is fed to cores C and D. Additionally, the clock interleaving mode is set to '2', meaning that the clocks to cores A and C are in phase and so are B and D. A simplified block diagram of the configuration is presented in Fig. 2.

4.2.3 Four-channel Mode

In four-channel mode, the cross-point switch is placed in mode '4', meaning each ADC input is fed to a single core, starting with *ADC input 0* to core A and so on. Additionally, the clock interleaving mode is set to '3', meaning that the clocks to all cores are in phase with each other. A simplified block diagram of the configuration is presented in Fig. 3.

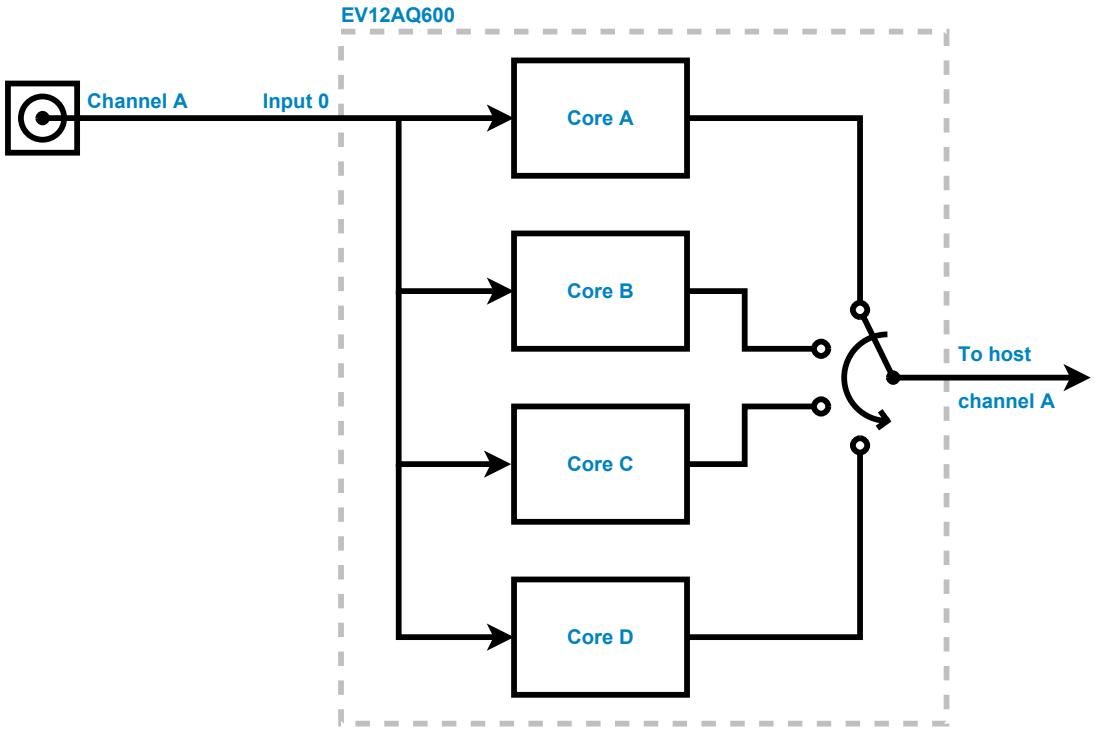

Figure 1: Analog input and cross-point switch configuration in the one-channel mode. The interleaving symbol does not reflect the sample order.

4.3 Clocking

The API [2] function `SetClockSource()` performs all the necessary operations associated with changing the clock source, such as recalibrating the PLL and performing a reset of the ADC followed by a resynchronization of the ESIstream interface.
The EVM starts up using the internal oscillator as a reference to the main PLL. The user may return to this clocking mode by specifying the 'internal clock reference' option in a call to `SetClockSource()`.

4.3.1 External Reference

The **EXTCLK** input may be used to provide a 10 MHz reference to the on-board PLL. The signal is passed through a jitter cleaning PLL before entering the main PLL which feeds the ADC with its core system clock on pins A11 and A12. When the jitter cleaning PLL is locked, the **EXTREF LOCKED** LED will light up. The external reference is activated by specifying the 'external clock reference' option in a call to `SetClockSource()`.

4.3.2 External Clock

The **CLK** input may be used to directly provide the ADC with its core system clock. The single-ended signal passes through a balun, a clock multiplexer and a clock buffer before entering the ADC on pins A11 and A12. The external clock is activated by specifying the 'external clock source' option in a call to

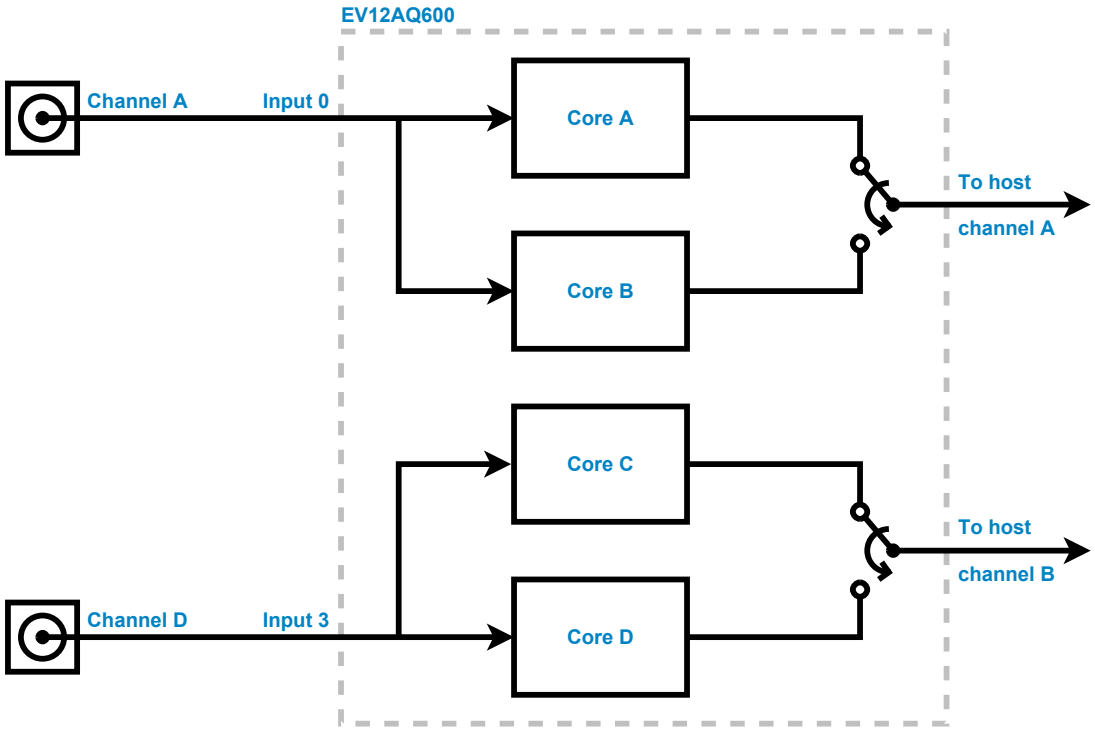

Figure 2: Analog inputs and cross-point switch configuration in the two-channel mode. The interleaving symbol does not reflect the sample order.

`SetClockSource()`.

4.4 Trigger Options

Currently the *software trigger* mode is supported by the EVM. In this mode, the host computer is responsible for issuing the trigger signal via the API function `SWTrig()` or by a corresponding operation in ADCaptureLab.

4.5 Limitations

The Software development kit (SDK) is used across a wide range of Teledyne SP Devices digitizers. For the EV12AQ600-ADX-EVM and EV12AQ605-ADX-EVM, the supported features are listed in Table 5. Even though supported, no example code is provided for the C++ and Python APIs.

4.6 Data Format

The 12-bit ADC codes are converted in FPGA firmware to 16-bit data. The 16-bit data is the MSB aligned 2's complement of the ADC data. The implemented operation is listed in (1).

$$Y = \left(2^4 \cdot X + 2^3\right) \oplus 2^{15} \tag{1}$$

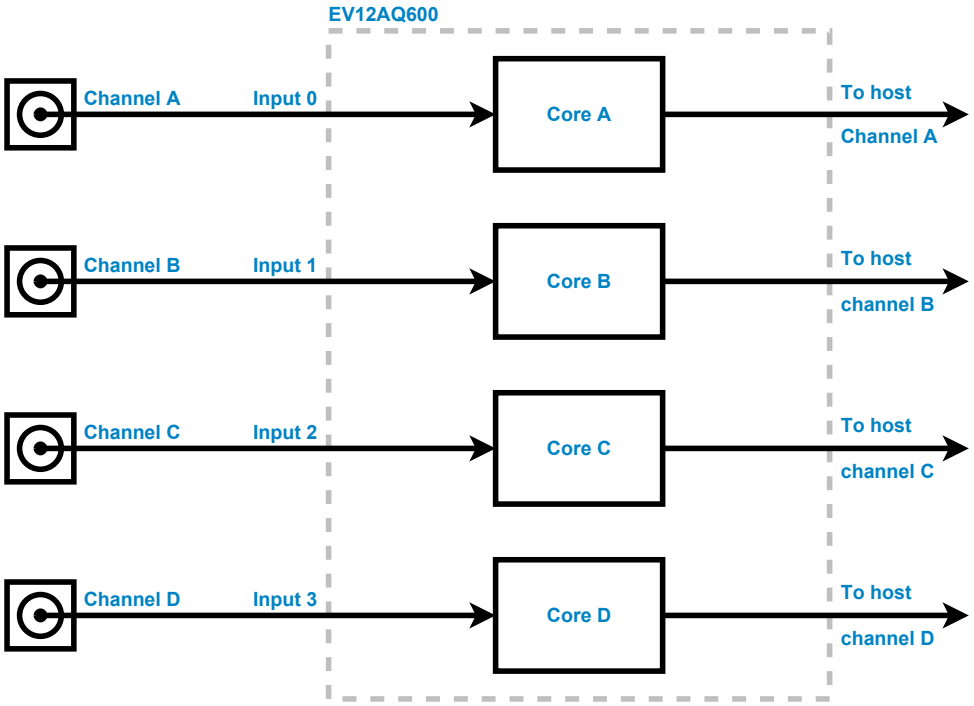

Figure 3: Analog inputs and cross-point switch configuration in the four-channel mode.

Table 5: The supported software features.

Feature	Supported	Unsupported
Microsoft Windows (7 and above)	✔	
ADCaptureLab	✔	
C (API)	✔	
C++ (API)*	✔	
Python (API)*	✔	
Linux		✖
Matlab (API)		✖
LabVIEW		✖

* EVM specific example code is not provided

where Y is the 16-bit 2's complement samples, X is the 12-bit ADC data, and $\oplus$ is the bitwise exclusive or operator. Table 6 list the 12-bit ADC codes and the corresponding 16-bit representation for a few values. In this document, excluding this section, 'ADC codes' refers to the 16-bit 2's complement values.

Important

In this document, 'ADC codes' refers to the 16-bit data samples transferred to the host.

Table 6: ADC digital output coding table.

ADC Binary output (12 bits)	2's complement (16 bits)
111111111111	0111111111111000
111111111110	0111111111101000
100000000000	0000000000001000
011111111111	1111111111111000
000000000001	1000000000011000
000000000000	1000000000001000

4.6.1 Converting from ADC Codes to Volts

The unit of the received data is ADC codes. The input voltage is mapped to the range $[-2^{15-1}, 2^{15-1}-1]$, meaning one LSB is

$$\mathrm{LSB}_{\mathrm{Volt}} = \frac{V_{\mathrm{FS}}}{2^{16}}\ \mathrm{V}. \tag{2}$$

where V_{FS} is the input range, see Section 2.1. The ADC codes may then be converted to Volt by using (3).

$$X_{\mathrm{mV}} = \frac{V_{\mathrm{FS}}}{2^{16}} \cdot X_{\mathrm{ADC}} \tag{3}$$

5 ADCaptureLab

ADCaptureLab is a graphical application that may be used to control the EVM. The interface consists of two tabs, one for data acquisition and plotting (*Main Tab*), and one for EVM settings (*Settings Tab*). ADCaptureLab will find and setup all available devices by default. The devices will be listed on the *Main Tab*.

Tip

The ADCaptureLab window can be cycled between windowed and full screen mode by pressing the F11 key.

5.1 Main Tab

The main tab is shown in Fig. 4 and used for data acquisition and visualization. Pressing the *Acquire data* button will cause a new batch of data to be triggered, collected and displayed in the time domain and frequency domain plots.

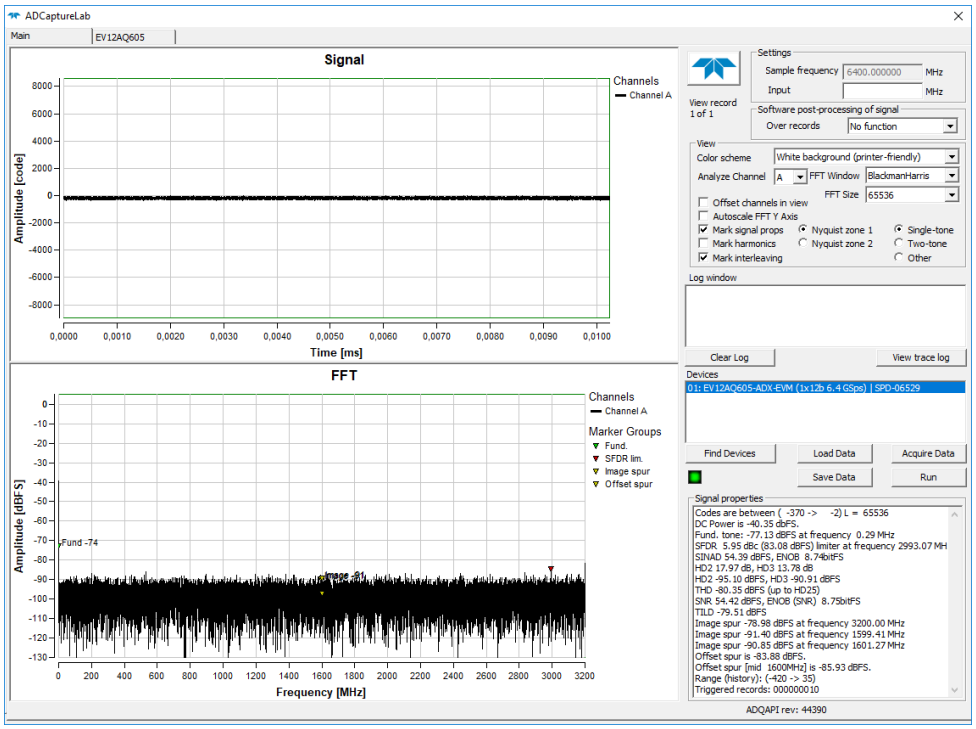

Figure 4: Screen capture from ADCaptureLab main tab.

Performance metrics such as interleaving/distortion spur amplitudes, SNR, SNDR, SFDR and THD are listed in the signal properties window, which can be found on the lower right-hand side. Only a single input channel is analyzed, and the selected channel can be changed in a drop-down menu in the upper right-hand side of the window.

Note

The FPGA data path converts the 12-bit ADC data to MSB-adjusted 16-bit data. The plots displayed in ADCaptureLab will therefore have full-scale ranges of -32768 to 32768 ADC codes (see Section 4.6)

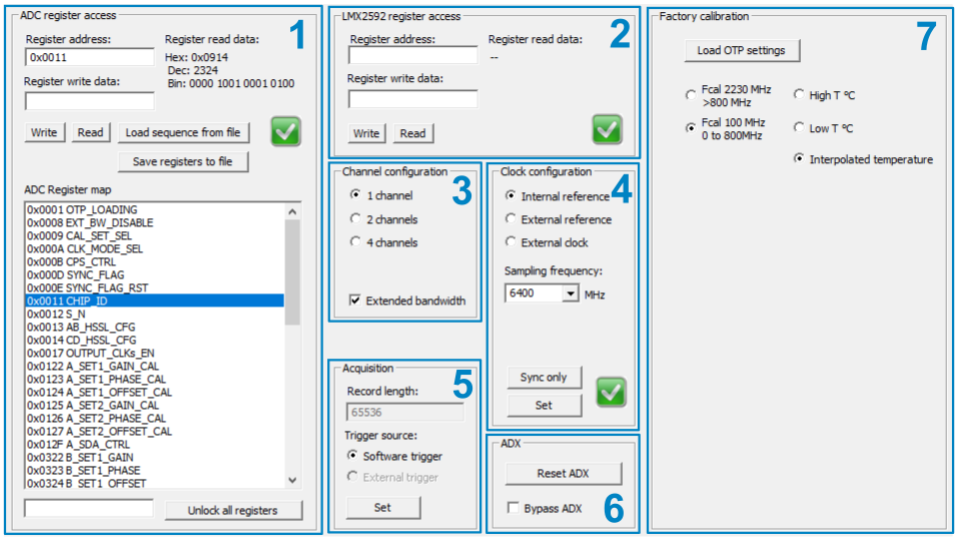

Figure 5: Screen capture from ADCaptureLab EV12AQ600-ADX-EVM settings dialog.

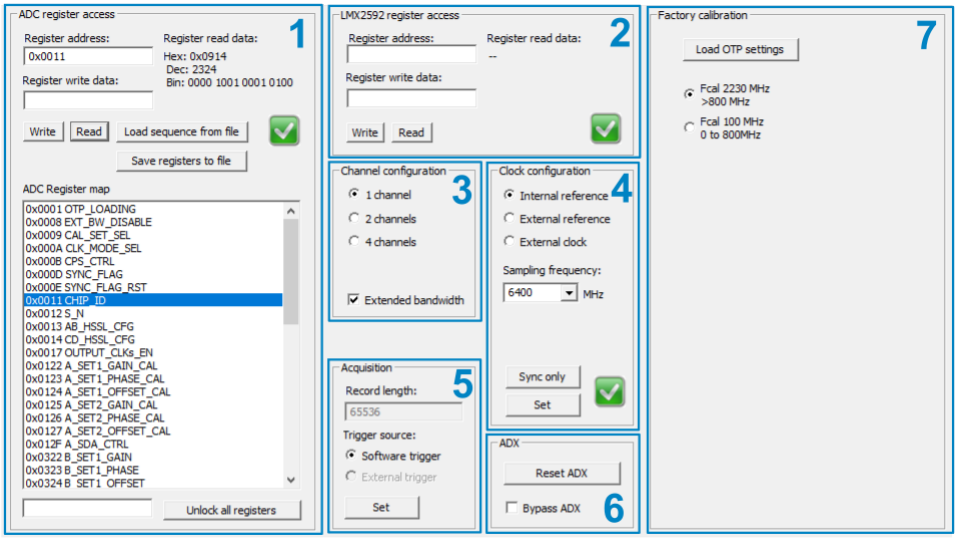

Figure 6: Screen capture from ADCaptureLab EV12AQ605-ADX-EVM settings dialog.

5.2 Settings Tab

The settings tab is shown in Fig. 6. The numbers in the list below refer to the numbers in the figure.

1. **ADC register access**

   The register addresses can either be entered manually, or by clicking on one of the rows in the register map list. A number preceded by 0x is interpreted as a hex value, and otherwise as a decimal value. Values are read from the ADC when the “Read” button is pressed and written to the ADC when the “Write” button is pressed.

   The button “Write sequence from file” loads a file containing register values and writes them to the ADC.

   The button “Save registers to file” reads out all register values from the ADC to a file.

2. **LMX2529 register access**

   The registers are 16 bits wide. Please refer to the LMX2529 datasheet for usage. [3]

3. **Channel configurations**

   Switch between the available channel modes. Switching to a different channel mode also loads a different FPGA image, which takes a few seconds. The *Extended bandwidth* check box enables/disables the extended bandwidth setting of the ADC.

4. **Clock source and sampling frequency selection**

   The user can choose between internal clock generation, external 10 MHz reference, or direct external clocking. The sampling frequency drop down box allows selection of sample rates between 2 and 6 GSPS. No changes take effect until the *Set* button is pressed.

5. **Acquisition settings**

   Currently only a single record length is allowed, and only software trigger is allowed as a trigger source.

6. **ADX control**

   The bypass button allows a complete bypass of the ADX interleaving algorithm, which is useful for comparing the interleaving performance with/without ADX. The reset button clears the current mismatch estimation inside ADX. After a reset, it may take a few seconds before the interleaving spurs are reduced again.

7. Factory calibration

Regarding interleaving tuning, AQ600 device contains 4 factory calibration sets that can be selected as following:

- “Fcal 100 MHz” and “Low T °C”
  for low frequency (Fin <800 MHz) and low temperature (<75 °C)
- “Fcal 2230 MHz” and “Low T °C”
  for high frequency (Fin >800 MHz) and low temperature (<75 °C)
- “Fcal 100 MHz” and “High T °C”
  for low frequency (Fin <800 MHz) and high temperature (>75 °C)
- “Fcal 2230 MHz” and “High T °C”
  for high frequency (Fin >800 MHz) and high temperature (>75 °C)

In case the Eval Kit is operating at a stable temperature, interleaving performances fine tuning is possible using *temperature linear interpolation feature*[1]. Two other configurations can be selected as following:

- “Fcal 100 MHz” and “Interpolated temperature”
  for low frequency (Fin <800 MHz) and stable temperature operating conditions
- “Fcal 2230 MHz” and “Interpolated temperature”
  for high frequency (Fin >800 MHz) and stable temperature operating conditions

The AQ605 device contains 2 factory calibration sets that can be selected as following:

- “Fcal 100 MHz” and “Low T °C”
  for low frequency (Fin <800 MHz)
- “Fcal 2230 MHz” and “Low T °C”
  for high frequency (Fin >800 MHz)

For both AQ600 and AQ605, all those factory calibration values are optimized for IN0 input.

Note

Each time the clock or channel configuration settings are changed, the ADC will be reset. This causes all ADC registers to revert to their default settings and any manual changes to the register values will be lost.

Warning

The LMX2592 PLL is intricately involved in the clocking and initialization of the ADC/FPGA interface and the FPGA data path. Writing to the PLL registers may cause the EVM to stop working, and require a restart. For changing clock frequency and clock source, we recommend using the clock configuration section of the interface instead.

[1] *Temperature linear interpolation feature*: Factory calibrations are done at two specific temperatures. Based on the value of the die junction temperature diode, a linear interpolation of the factory calibration values is done.

5.2.1 Register file format

To preserve a sequence of register writes for reuse, an `.rseq` file can be created. This file can be written to the board using the “Write sequence from file” button.
The `.rseq` file is a simple ASCII text file, where each row can contain either a register write instruction, a pause, or a comment. The following is an example:

```
# Write the value 0x0002 to register 0x0009
W 0x0009 0x0002
# Pause for 10 milliseconds
S 10
# Done
```

6 Troubleshooting

This section aims to provide some guidance when troubleshooting unexpected behavior. It is recommended that the user application is written in a robust manner, able to capture and report error codes from failed ADQAPI function calls. In the event of a function call failure, reading the ADQAPI trace log for additional information is a useful first step. Trace logging must be activated by calling `ADQControlUnit_EnableErrorTrace()` with the `trace_level` argument set to 3.
Running the device from ADCaptureLab creates a log file automatically. The file is placed in the current user’s `AppData` folder, more specifically

```
C:/Users/<current user>/AppData/Local/SP Devices
```

assuming the operating system uses `C:` as the system drive.

7 Known issues

- Clock rates below 4 GHz in the one-channel mode are currently not supported.
- Clock rates below 2 GHz in the two-channel mode are currently not supported.
- Clock rates below 1 GHz in the four-channel mode are currently not supported.

References

[1] Teledyne e2v, *Product specification EV12AQ600 (Rev. A)*, October 2016. Datasheet.

[2] Teledyne Signal Processing Devices Sweden AB, *14-1351 ADQAPI Reference Guide*. Technical Manual.

[3] Texas Instruments, *LMX2592 Wideband Frequency Synthesizer with Integrated VCO (Rev. F)*, November 2017. Datasheet.