UM10408_2

DAC1x08 demonstrator: Demonstration board for DAC1x08D

Rev. 02 — 2 July 2012 **User manual**

Document information

Info	Content
Keywords	JESD204A, PCB2064-3, PCB2064-4.0, Demonstration board, DAC, Labview, DAC1408D, DAC1208D, DAC1008D
Abstract	This document describes the use of DAC1x08D Demonstrator for the JESD204A-compliant digital-to-analog DAC1x08D converters family.

Revision history

Rev	Date	Description
2.0	2012-07-02	Rebranded.
1.5	2010-07-29	This document applies also to DAC1008D and DAC1208D demo board.
1.4	2010-05-05	Discrepancies fix and troubleshooting topic added.
1.3	2010-03-10	Update to V2.2 labview software
1.2	2010-02-01	Update jumper settings
1.1	2010-01-18	Marcom campaign II release. 64K FPGA option added.
1.0	2009-11-25	New demoboard for HVQFN64 package. Reference is PCB2064-3.0
0.3	2009-07-03	Update on clocks and on PCB2064-2.0
0.2	2009-06-11	Update
0.1	2009-01-29	Initial version.

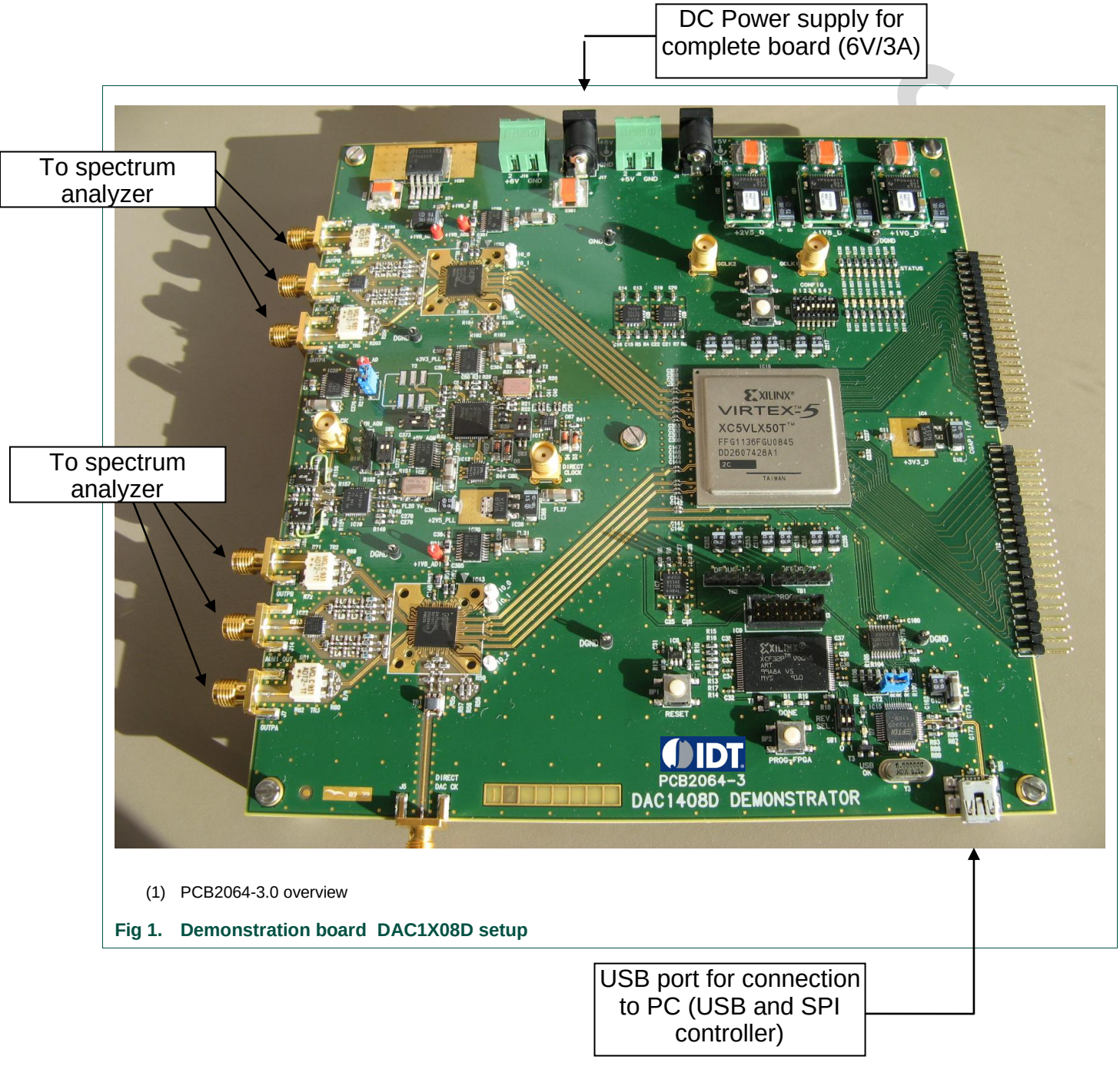
Contact information

For additional information or sales office addresses, please visit: http://www.idt.com.

1. Introduction

1.1 Setup overview

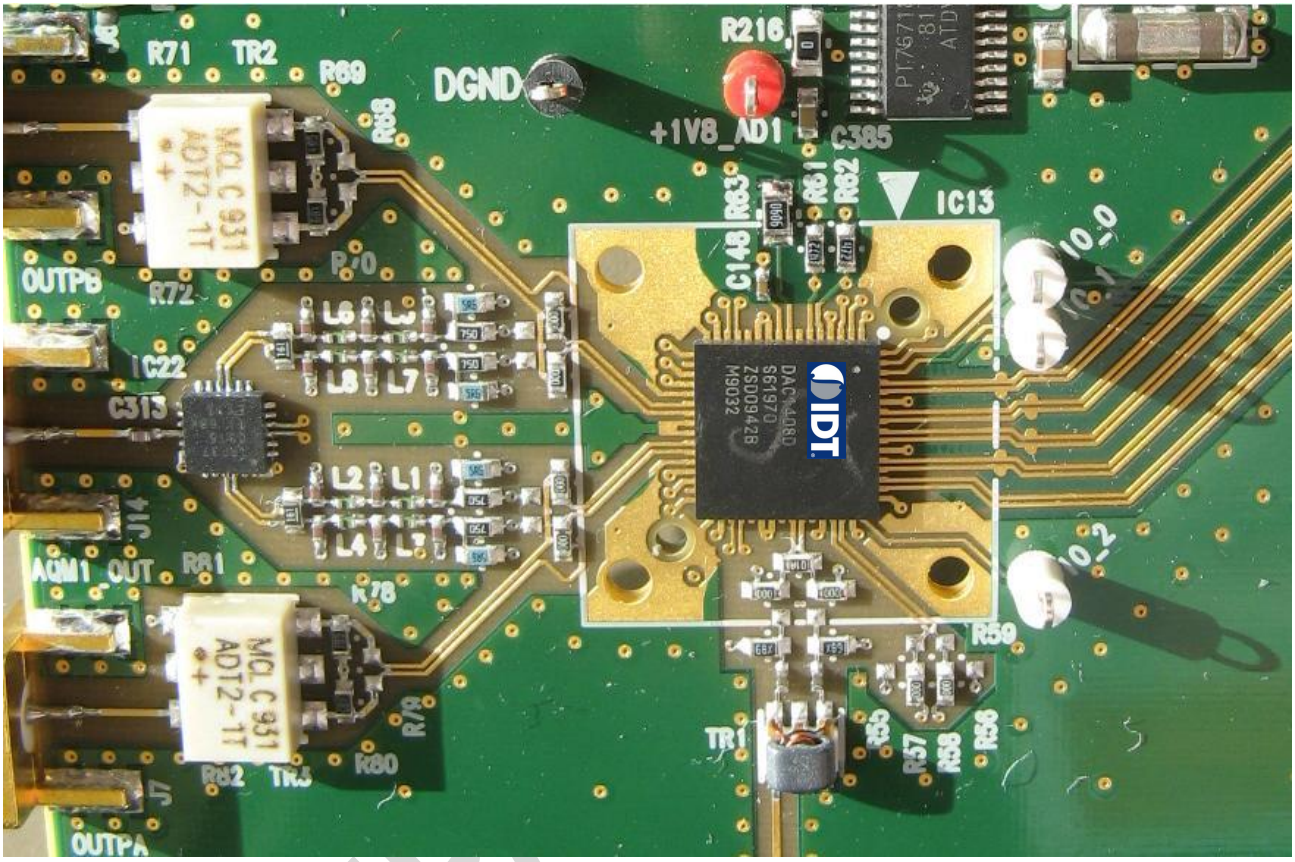
Fig 1 presents the connections to measure DAC1x08D Demonstrator.



(1) PCB2064-3.0 overview

Fig 1. Demonstration board DAC1X08D setup

1.2 Essential Features of the Demonstration Board



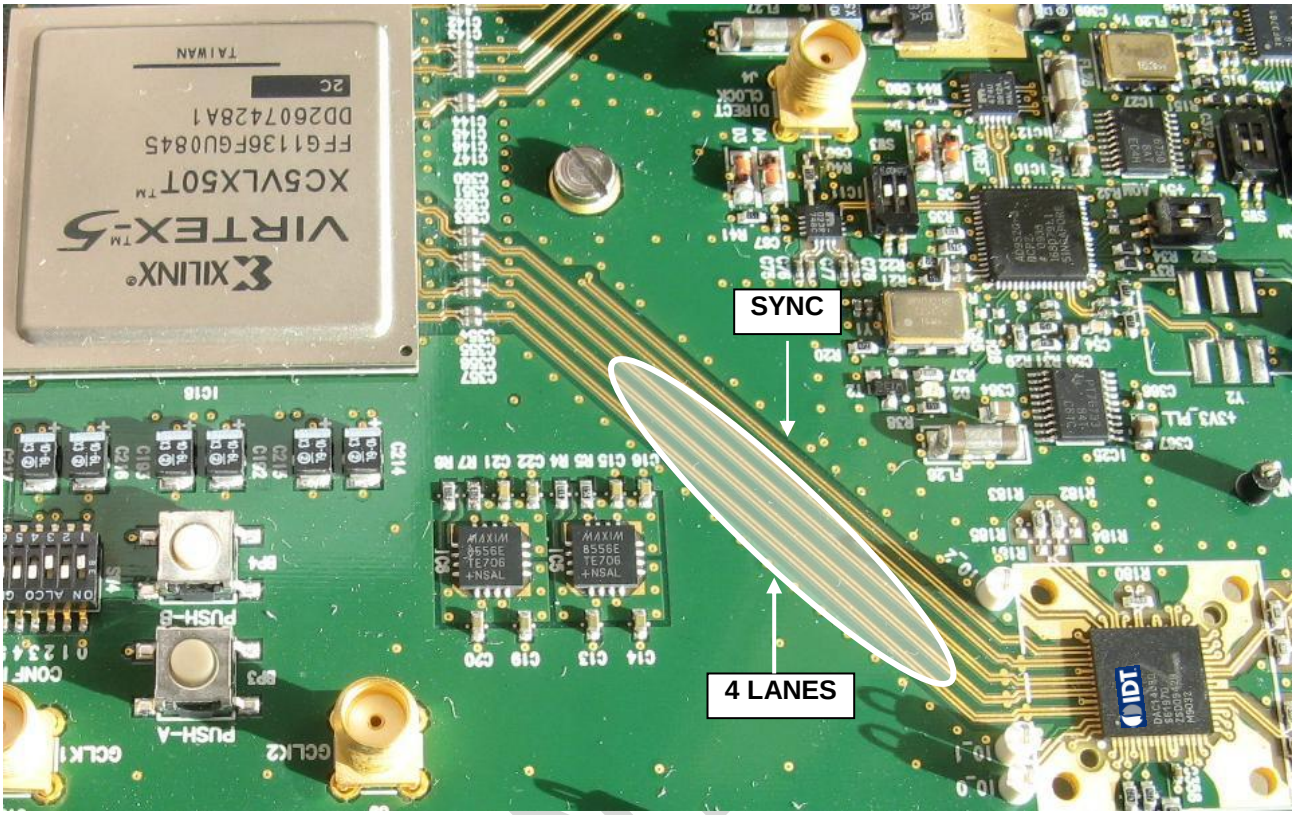
(1) IDT serial DAC device (9x9mm package)

Fig 2. DAC1X08D with 4 lanes in pairs of CML compliant differential

Fig 2 shows the DAC1x08D in its environment. The input is a series CML connection capable of sustaining a throughput rate of 3.125Gsps as specified by the JESD204A standard.

The output is connected to a transformer and then to an SMA output. Alternatively, an analog quadrature modulator can be used by means of de-soldering/soldering 0 ohms resistors.

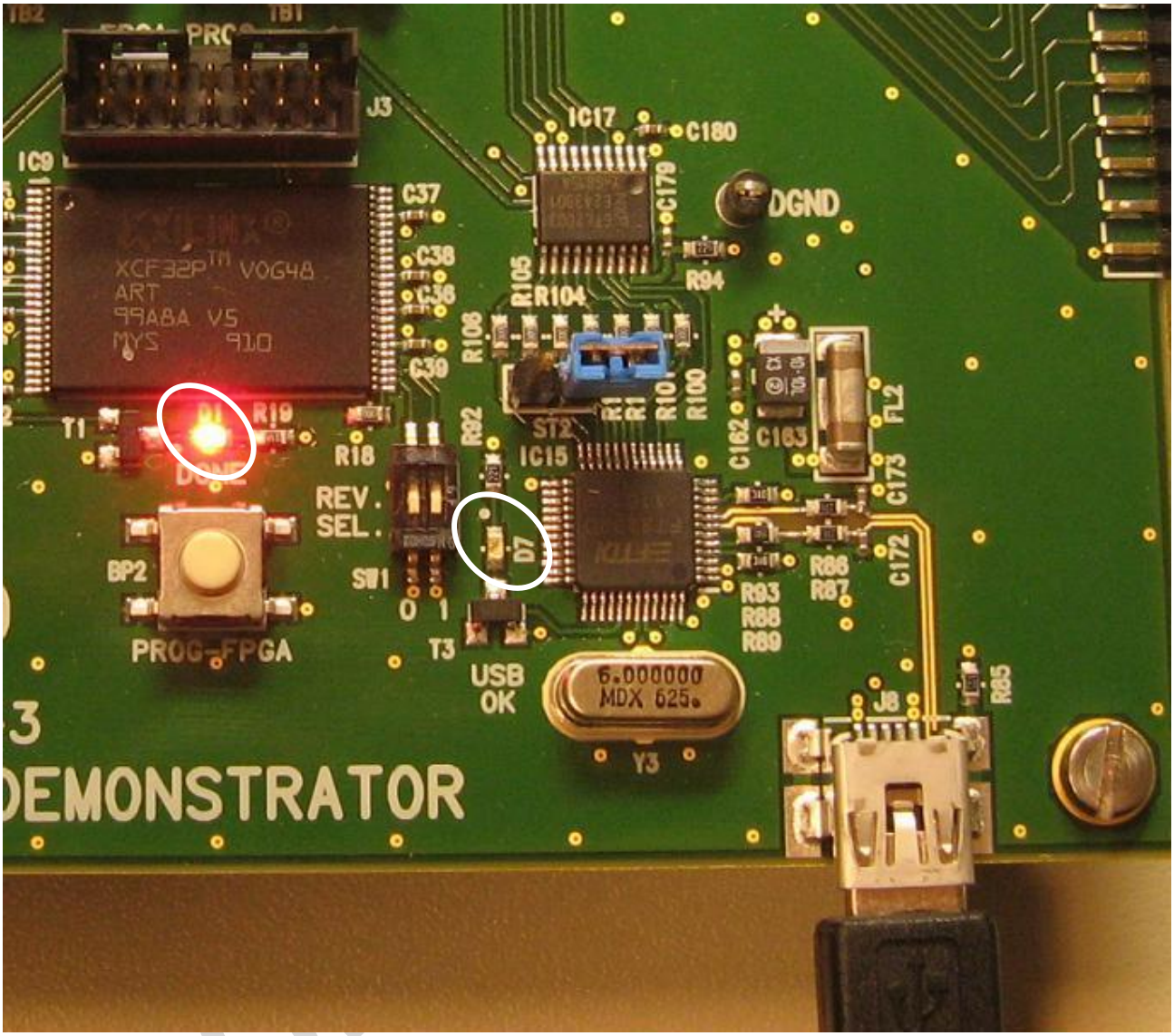
The logic device Field Programmable Gate Array (FPGA) is connected to the DAC1x08D via 4 Lanes with each lane in differential CML referenced to the positive supply. Moreover a synchronization signal, SYNC, is routed in differential also, between the FPGA and the DAC1x08D.



(1) Jesd204A serial interface

Fig 3. FPGA Logic device connected to the DAC1X08D via 4 Lanes and the SYNC

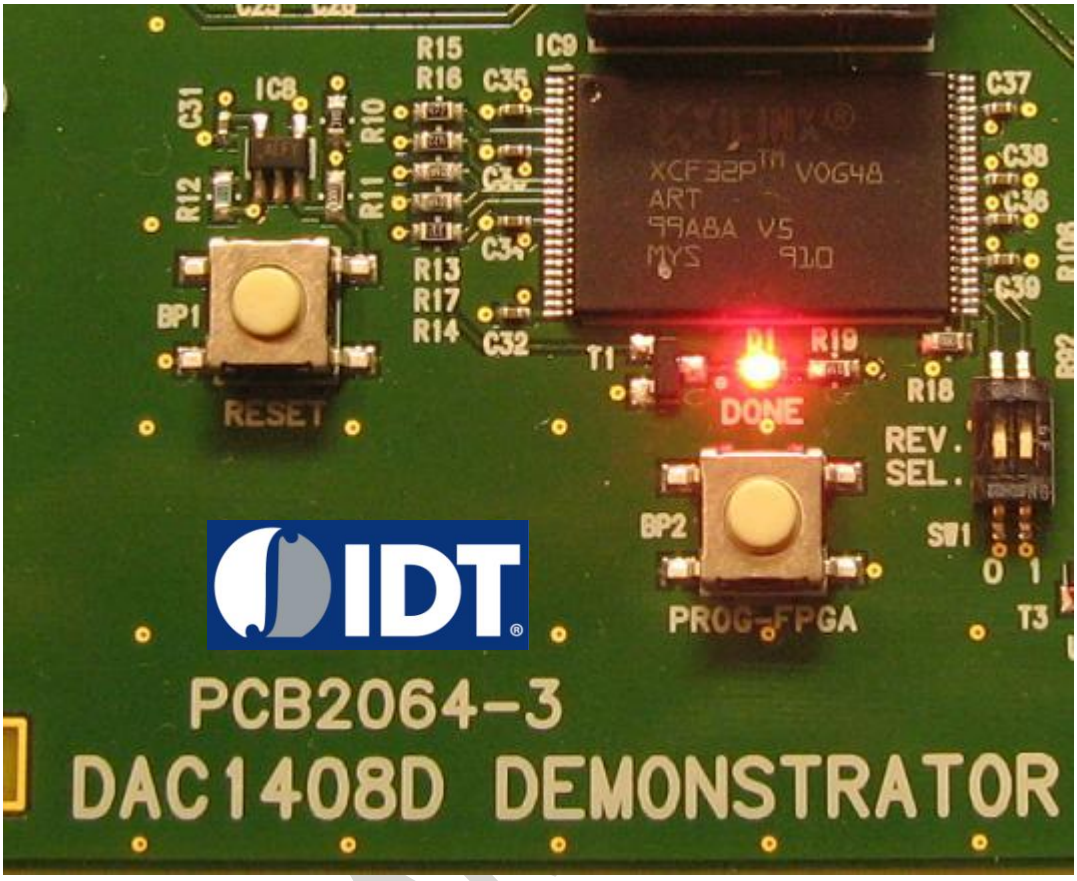
The board contains also a flash memory as shown in Fig. 4, to store the configuration file of the FPGA. This flash memory is loaded automatically into the logic device at start up. After the bit-stream has downloaded into the FPGA, the diode D1 lights up indicating that everything has went well.



(1)

Fig 4. On board memory with LED D1 indicating FPGA up-loaded and running and D7 indicating USB host detected

Furthermore, after connecting the USB port and installing the driver, the LED D7 indicates that the USB host has been detected and is up and running.



(1)

Fig 5. BP1 (Manual Reset) and BP2 (Manual upload of the flash content into the FPGA)

Push Button BP1 is a manual reset of the FPGA and the two DACs. User must press this button each time he starts a new test.

Push Button BP2 is a manual upload of the FPGA contents from the flash memory. This is automatically performed at power up of the board.

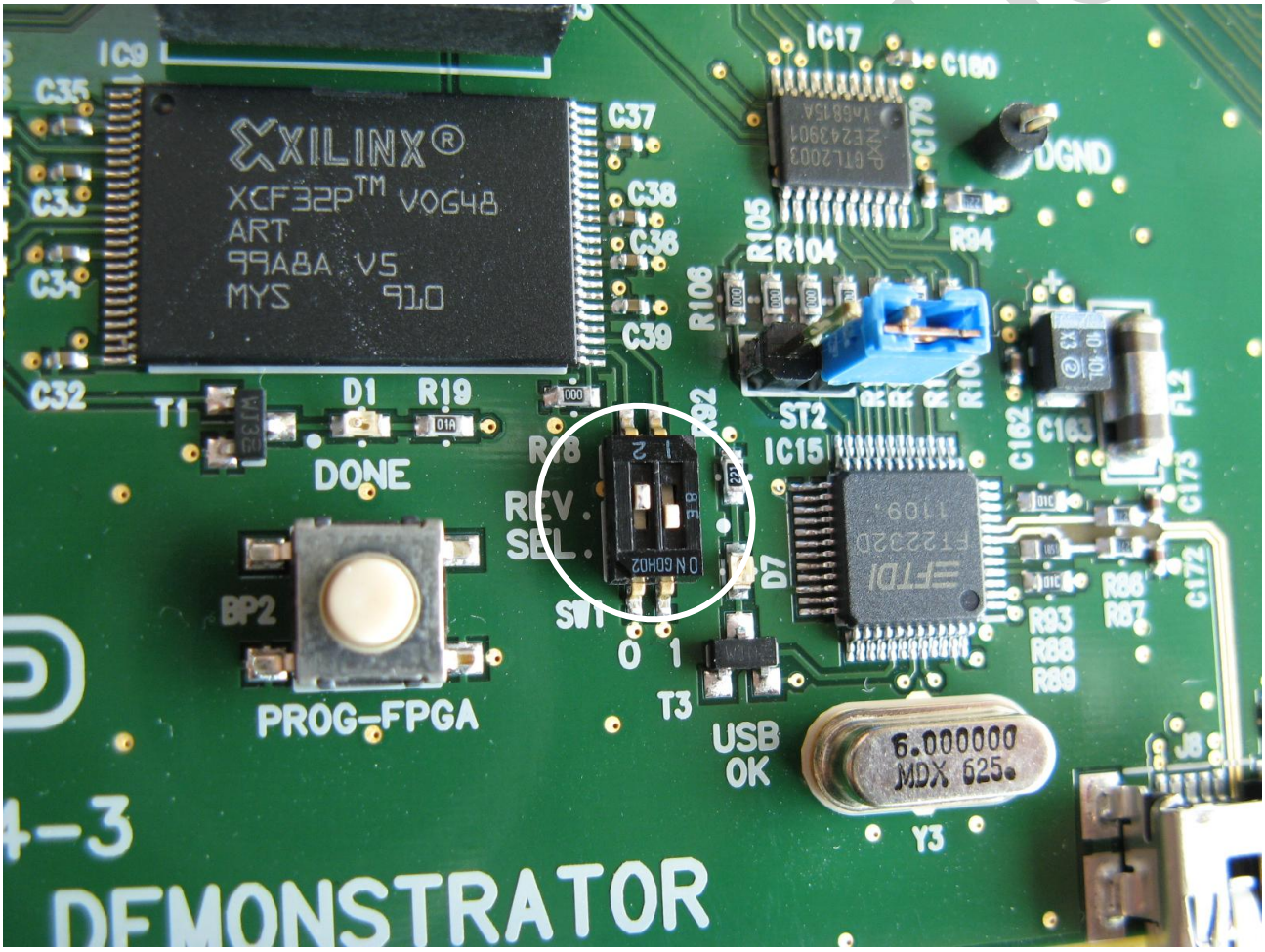
Dip switch SW1 is used to select the code loaded into the FPGA. The flash memory is large enough to hold two codes. Default position is ON-ON.

Table 1. SW1
Table description (optional)

0	1	Max size of the pattern loaded into FPGA memory	Max FPGA operating frequency
ON	ON	8K samples	310Mhz
OFF	ON	30K samples	190Mhz

0	1	Max size of the pattern loaded into FPGA memory	Max FPGA operating frequency
ON	OFF	Do not use this setting	
OFF	OFF	Do not use this setting	

Each time SW1 setting is modified, the FPGA code must be updated. Just press BP2 to trigger the upload process and wait until D1 lit.



(1) Selecting the 30K mode

Fig 6. Configuring the FPGA

Larger memories allow DAC ACPR measurements.

The FPGA operating frequency equals the maximum DAC input data rate.

(1)

Fig 7. D9 – D16 indicates the status of the FPGA

There are two rows of LEDs. Each reflects the status of one of the Jesd204A transmitter: D17~D24 are tied to the upper link and D8~D15 are tied to the lower one.

Table 2. FPGA status LEDs

Table description (optional)

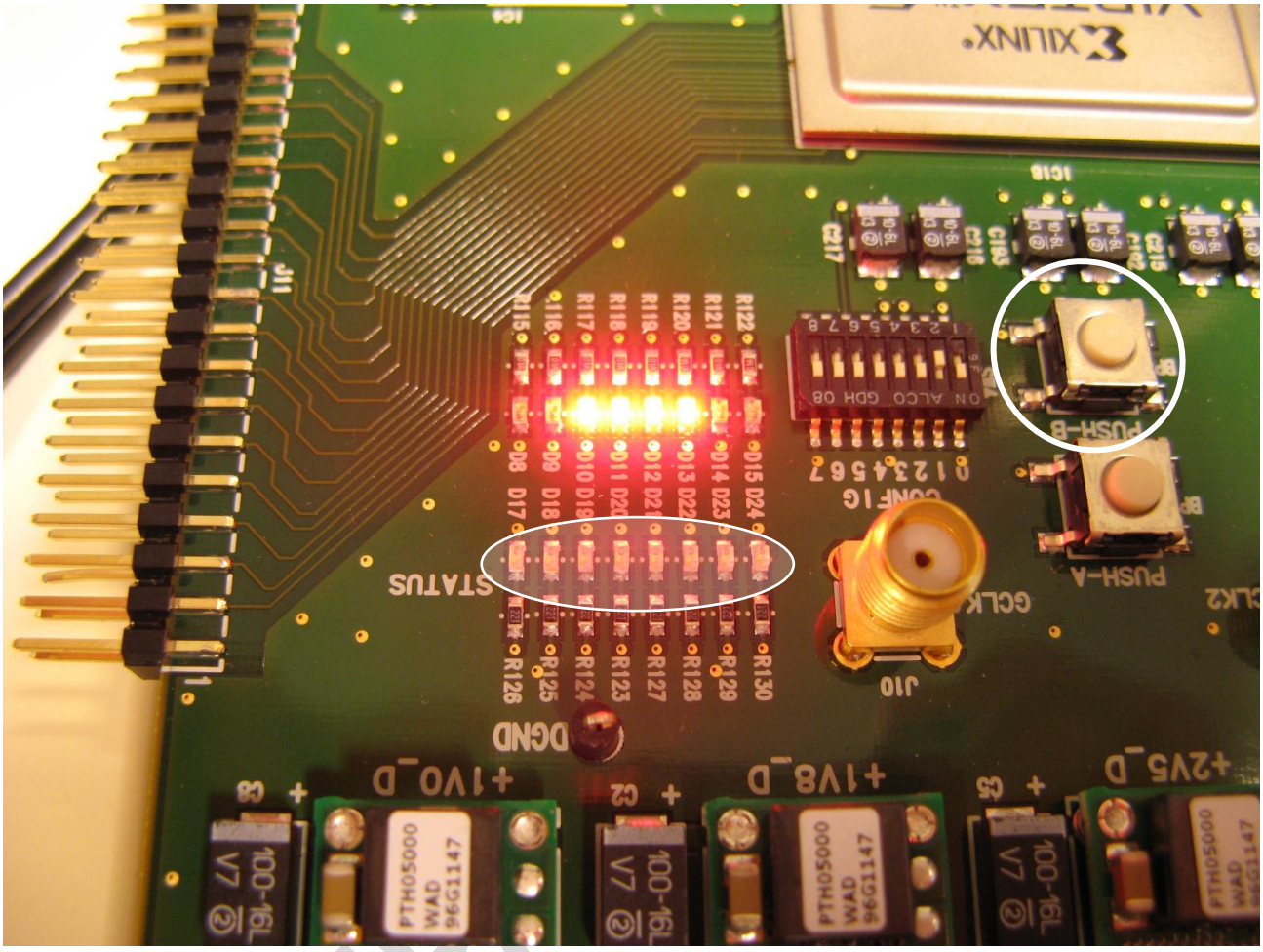
Upper	Lower	Meaning
D17	D8	always off
D18	D9	SYNC_REQUEST

Upper	Lower	Meaning
D19	D10	FPGA GTP0 lock status
D20	D11	FPGA GTP1 lock status
D21	D12	GTP0 reset done
D22	D13	GTP1 reset done
D23	D14	always off
D24	D15	FPGA Reset

At startup time or after pressing reset button, LEDs D9, D10, D11, D13, D18, D20, D21,D22 should lit.

The SYNC_REQUEST signal is a synchronization request signal used at the beginning of the transmission. It is always present between the FPGA and the DAC until the data is transferred from the logic device to the DAC1x08D. It is also used by the receiver to trigger loss of synchronization and requests re-initialization. When the data has been transferred D9 (D18) turns off.

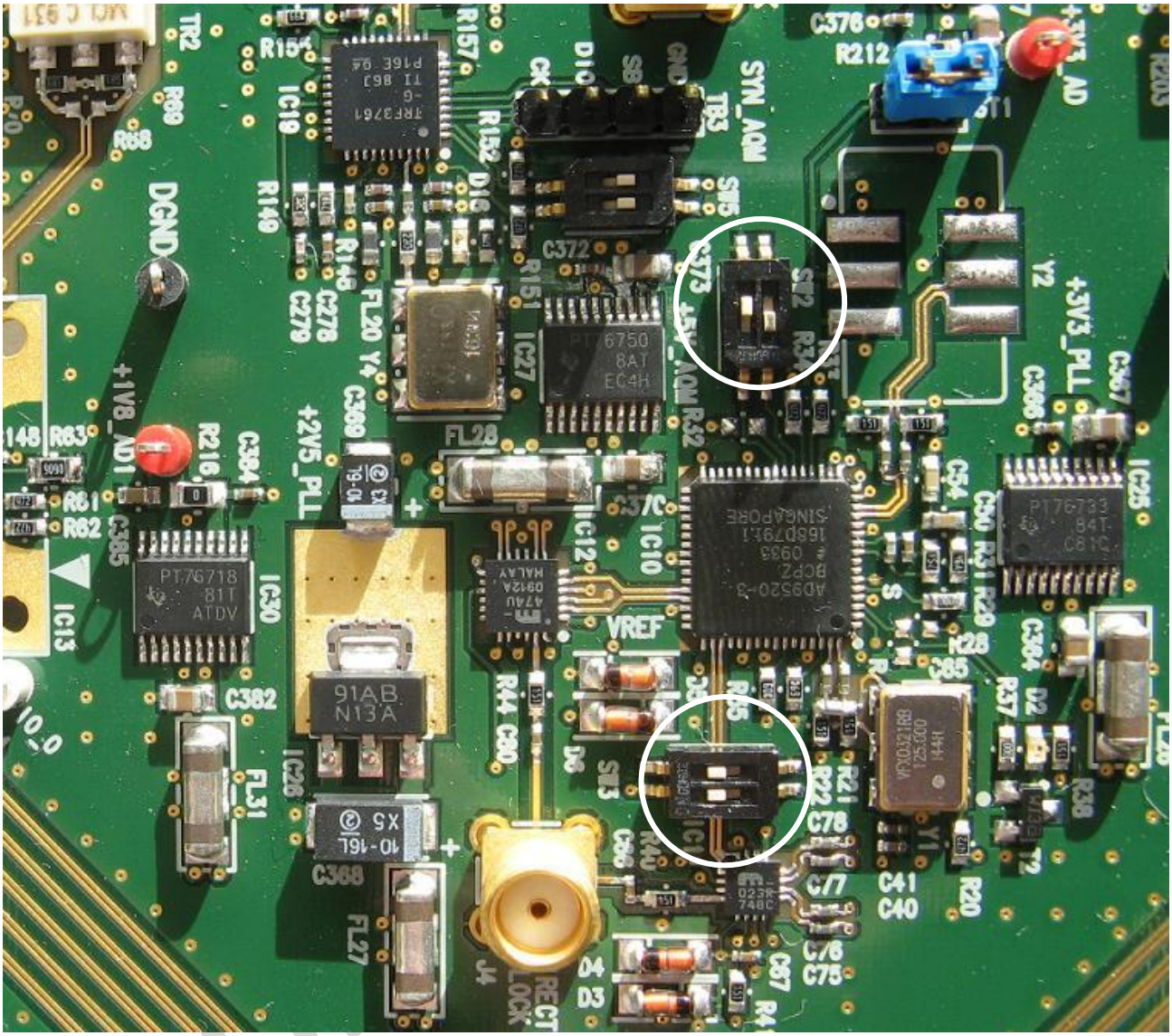
Using both DAC devices (IC13 & IC23) is optional. It is possible to hold IC23 (top DAC1x08) in reset and then use only IC13 (bottom DAC). Pressing push button BP4 disables IC23. To reflect this state, Led D17~D24 are turned off.
Pressing the main reset button (BP1) will re-activate DAC IC23 as part of the FPGA reset process.



(1)

Fig 8. Disabling DAC IC23

The clock signal can be generated on the board as there is a Phase Locked Loop (PLL) available.



(1) Default configuration

Fig 9. On board clock generation

By default, the frame clock needed by the FPGA and the two DACs is provided by IC10. In that mode, device internal VCO is used and locked to a 125Mhz reference oscillator. The actual clock frequency provided to the DAC and FPGA is set via software.

Dip switch SW2 sets IC10 startup behavior.

Table 3. SW2

IC10 startup behavior

Postion	ON	OFF
1	Load default registers settings at startup. Resulting frequency will	No default registers settings at

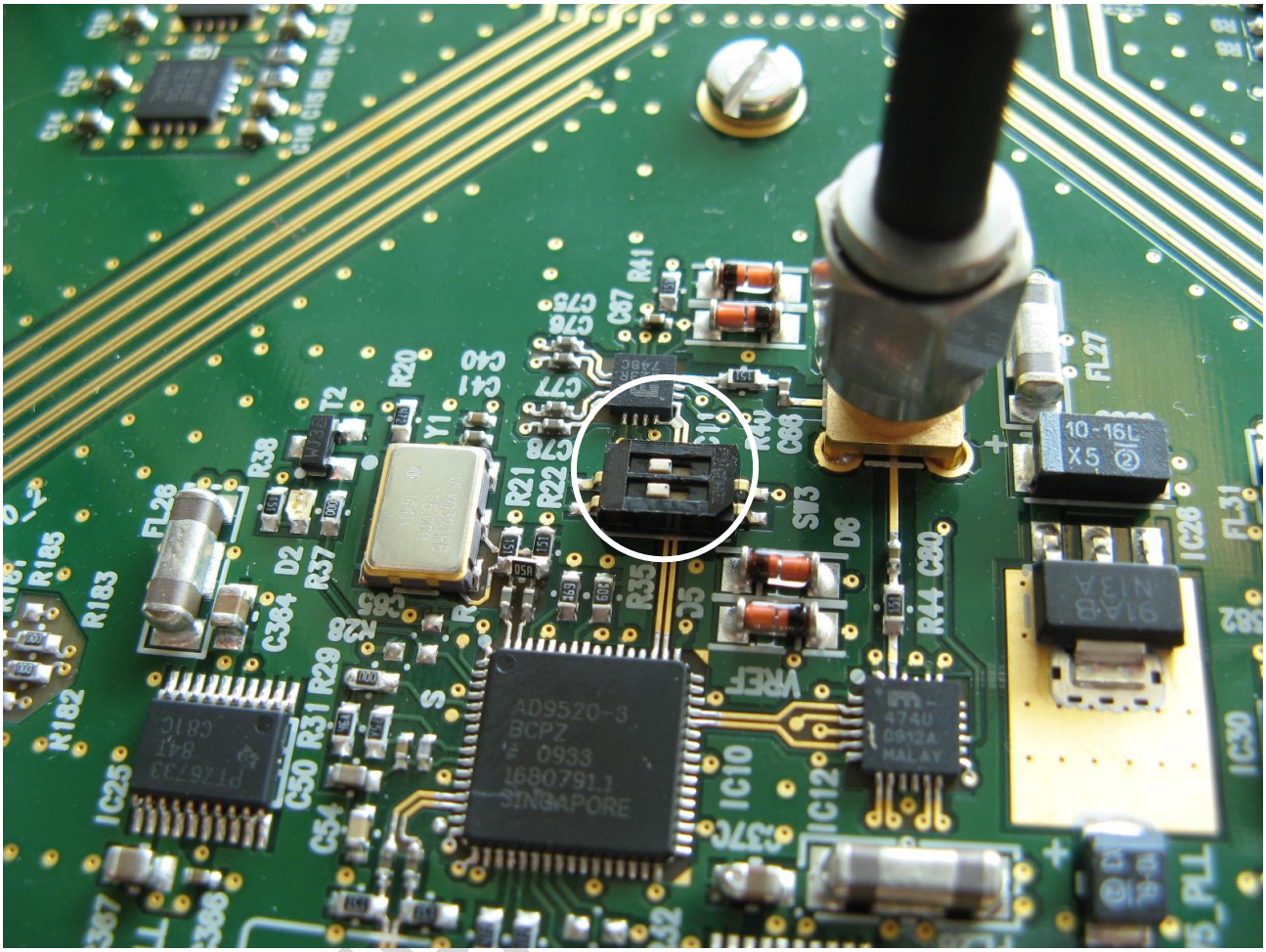
Postion	ON	OFF
	be 312.5Mhz.	startup.
2	Power down device	Device active.

Remark: the clock can also come externally through SMA connector J4. Multiplexers are available so as to route the right clock signal to the devices, the DAC1x08D and the FPGA. The clock source is selected using dipswitch SW3:

Table 4. SW3
Clock source selection

Position	1	2	Action
	ON	ON	On board PLL (IC10)
	OFF	OFF	External clock (J4)
	Other combinations will lead to unexpected behavior		

J4 is 50 ohms terminated. The recommended power is +13dBm.



(1) SW3 configured to select an external clock input

Fig 10. Using an external clock

2. Example

2.1 Setup example

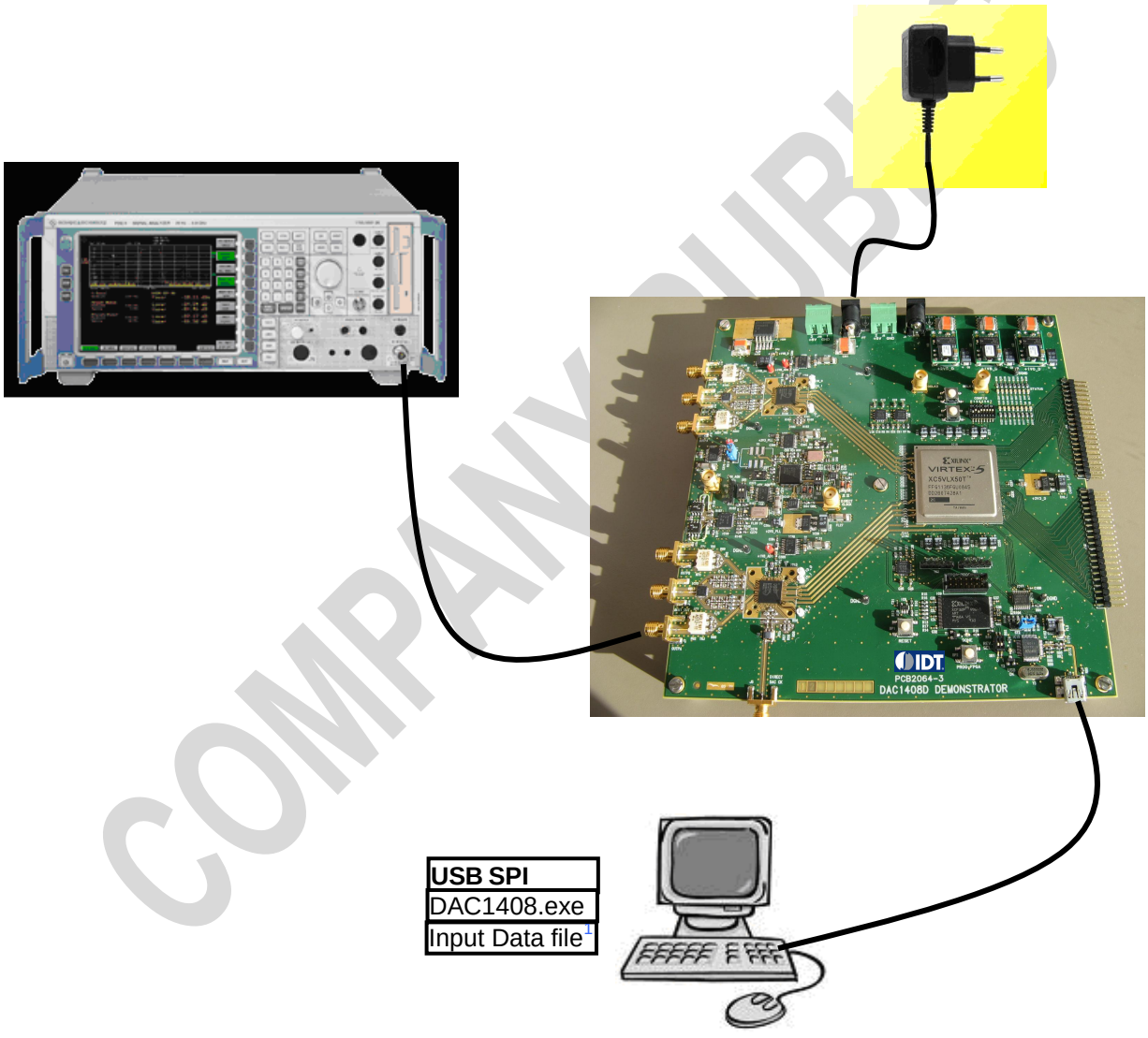


Fig 11. DAC1X08D Demonstrator Hardware setup

3. SPI quick start

3.1 Install

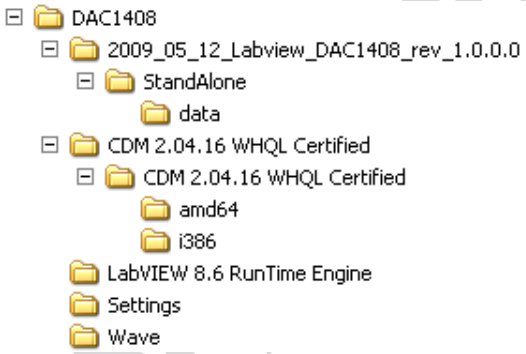
The demonstration board is delivered with the following software:

Labview Runtime: LVRTE86f1std

Labview executable: DAC1408.exe

Appropriate drivers

These are stored as follows:

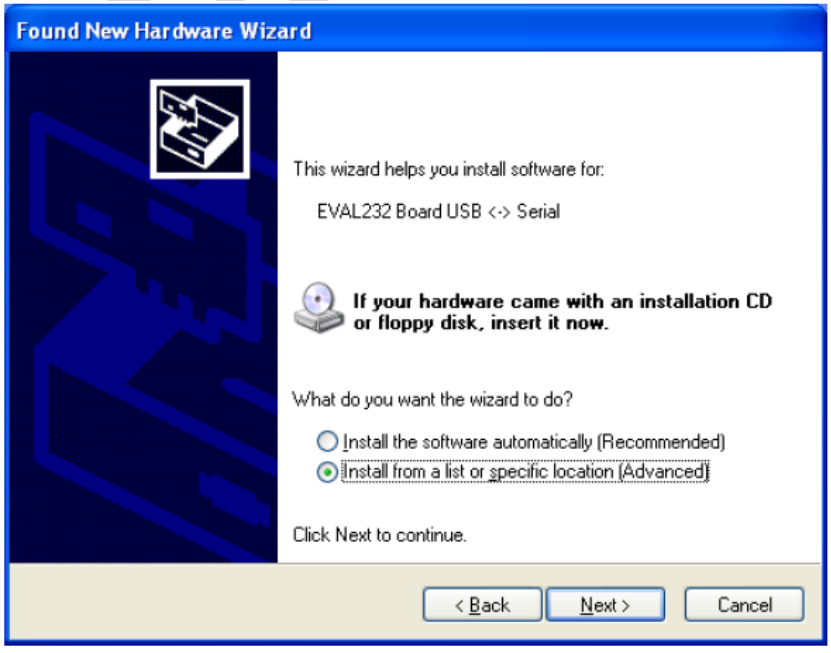


- Step 1

Connect the device to a USB port on your PC. Windows '**Found New Hardware Wizard**' will be launched. Select **'No, not this time**' from the options available and then click '**Next**' to proceed with the installation.

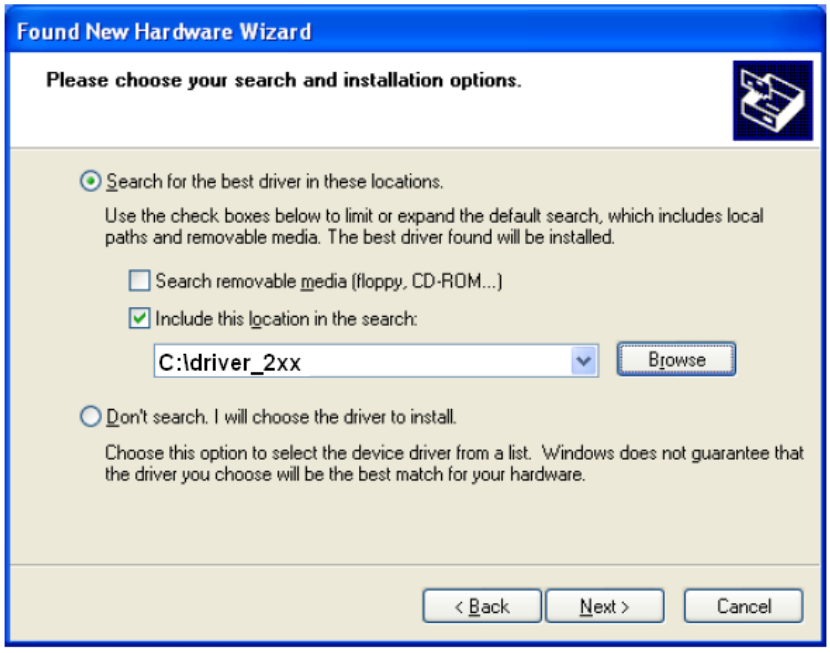
- Step 2

Select '**Install from a list or specific location (Advanced)**' as shown below and then click '**Next**'.



- Step 3

Select '**Search for the best driver in these locations**' and enter the file path of the folder '**DAC1408\CDM 2.04.16 WHQL Certified**' in the combo-box ('C:\driver_2xx' in the example below) or browse to it by clicking the browse button. Once the file path has been entered in the box, click '**next**' to proceed.



- Step 4

Windows should then display a message indicating that the installation was successful. Click 'Finish' to complete the installation for the first port of the device.

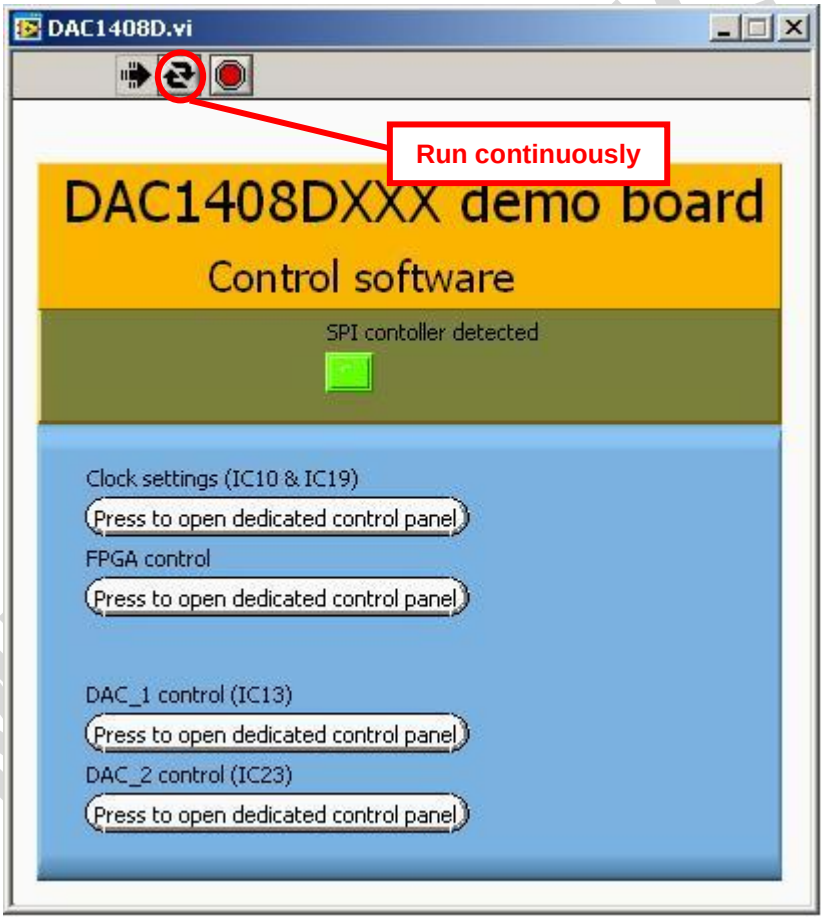
3.2 SPI interface

- Step 1

Install the LabVIEW Run-time Engine, LVRTE86f1std (if not already installed).

- Step 2

Start the LabVIEW application “DAC1408.exe”.



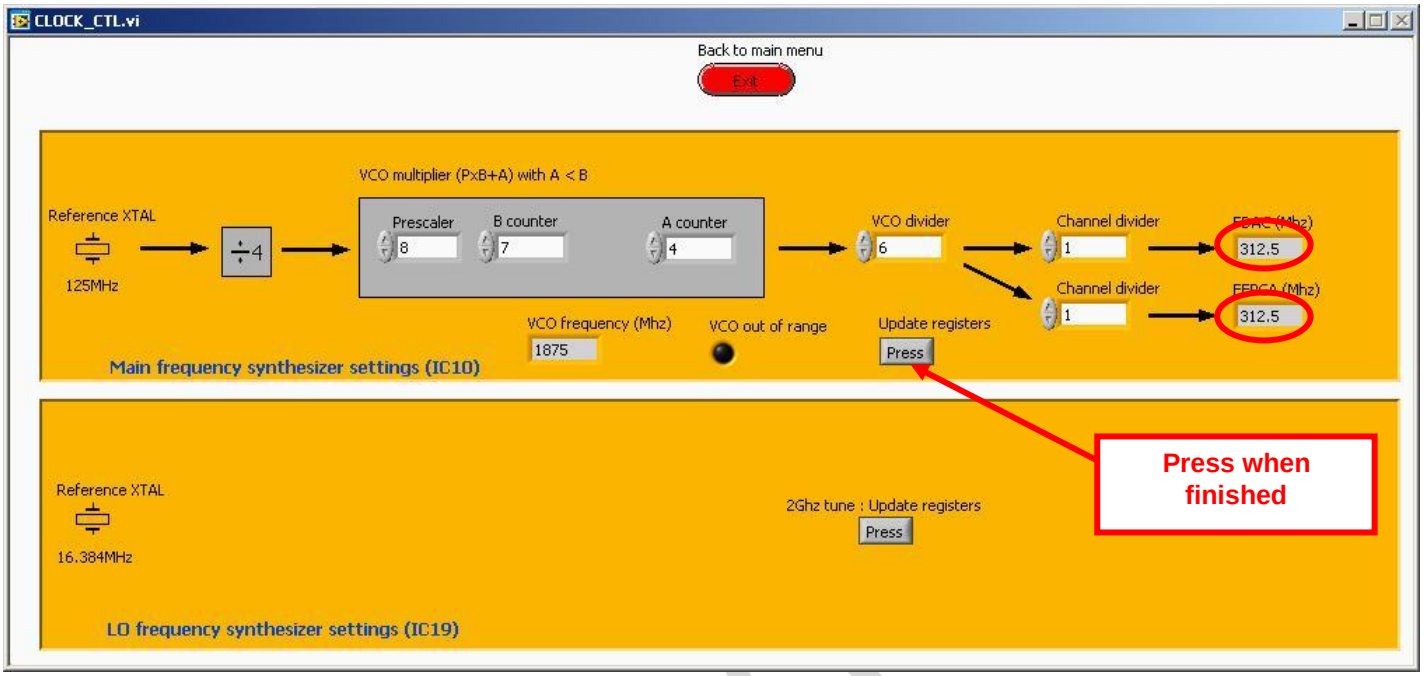
This is the main page of the GUI.

Click on the ‘**run continuously**’ button. The “SPI controller detected” marker should be green to indicate correct communication between the software and the board.

- Step 3

Press the ‘**clock settings**’ button.

At startup, the on-board synthesizer is configured to generate 312.5Mhz for both DACs and FPGA.



FDAC indicator reflects the DAC frequency resulting from the current divider settings. It is not the actual board frequency.

FFPGA indicator reflects the FPGA frequency resulting from the current divider settings. It is not the actual board frequency.

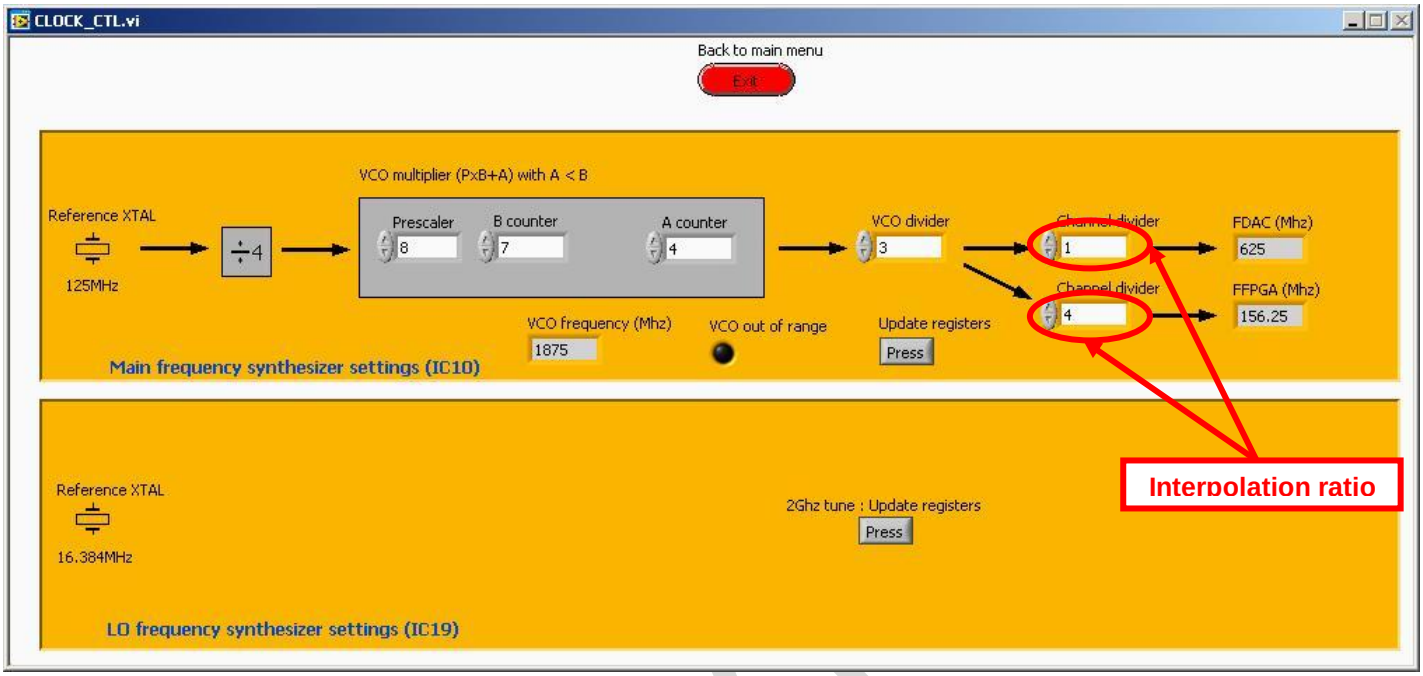
To apply these settings to the board, the user must press the '**Update registers**'.

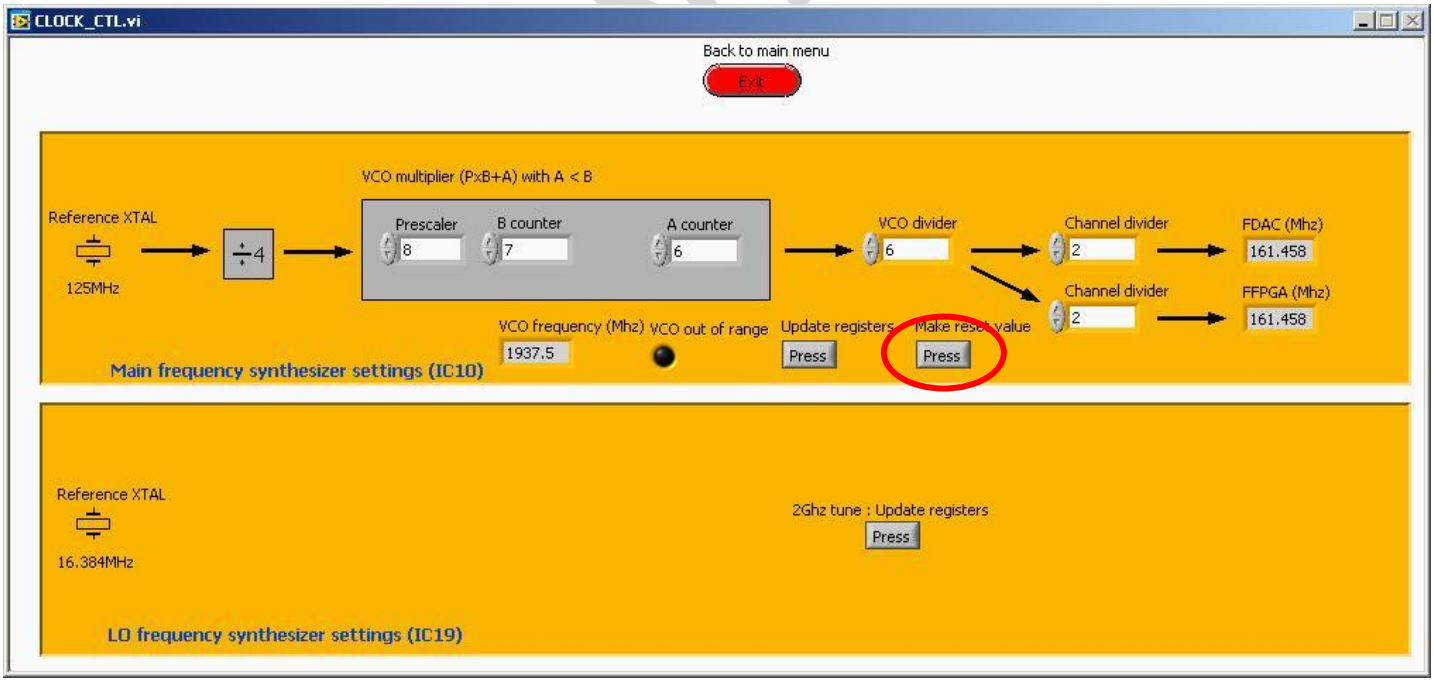
After each registers update, led D2 should lit. This reflects the correct frequency lock of IC10.

When the DACs are set to use their internal pll then FDAC must equal FFPGA.

If the DACs are set in pll by-pass mode then the ratio FDAC over FFPGA must reflect the interpolation ratio. The various synthesizer dividers are here for that purpose.

In the example below, FDAC = 4 x FFPGA. This means that the DAC is in pll bypass mode and that the output sample rate equals four times the input samples rate.



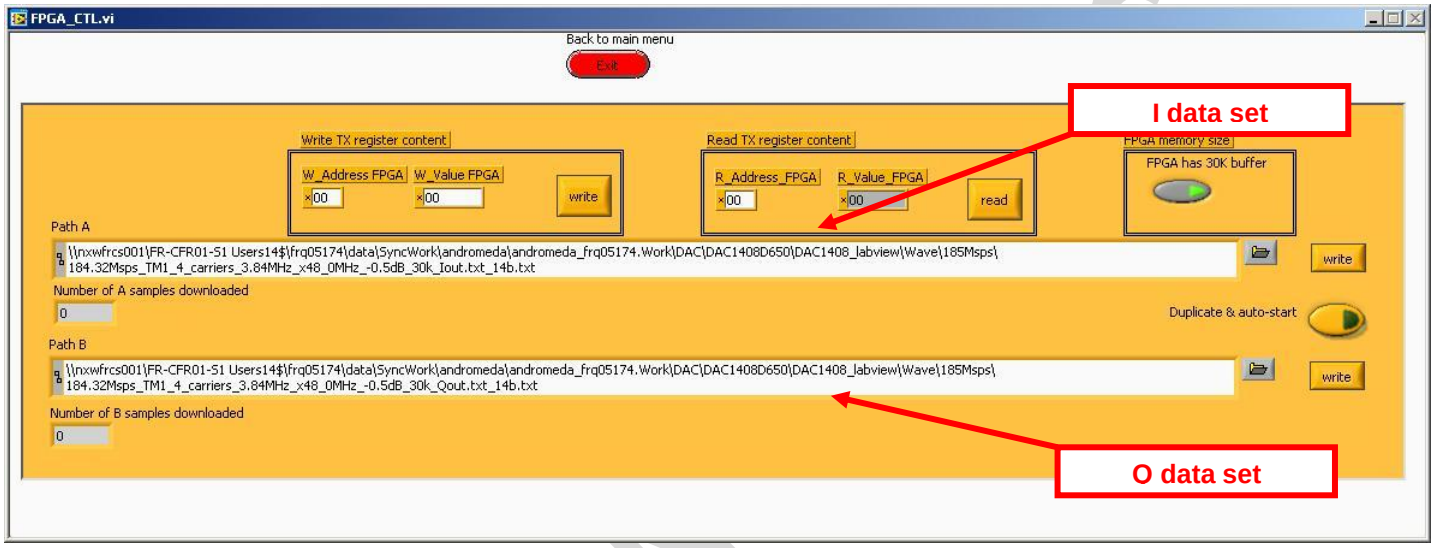


IC10 has an embedded flash memory. During power-on, it fetches the startup value from there. The content of the flash can be over-written with the current settings using the "**Make reset value**" button. At next power-up, these will be the default settings of the frequency synthesizer.

Once finished, close the window using the red button '**Exit**'.

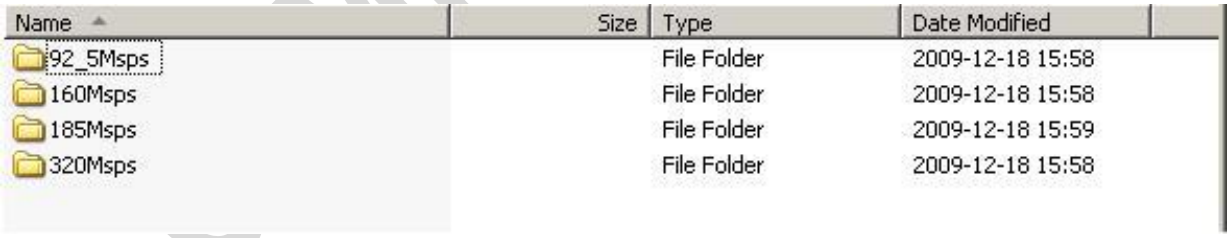
- Step 4

Press the '**FPGA control**' button.



This GUI is used to load the data that will be sent to the DAC by the FPGA.

Load a dataset/pattern provided in the folder *DAC1408\Wave*, in the Path A.



Remark: the format of the patterns provided is a simple text file, with 4 hexadecimal numbers per row and a maximum of 8192 or 30720 rows.

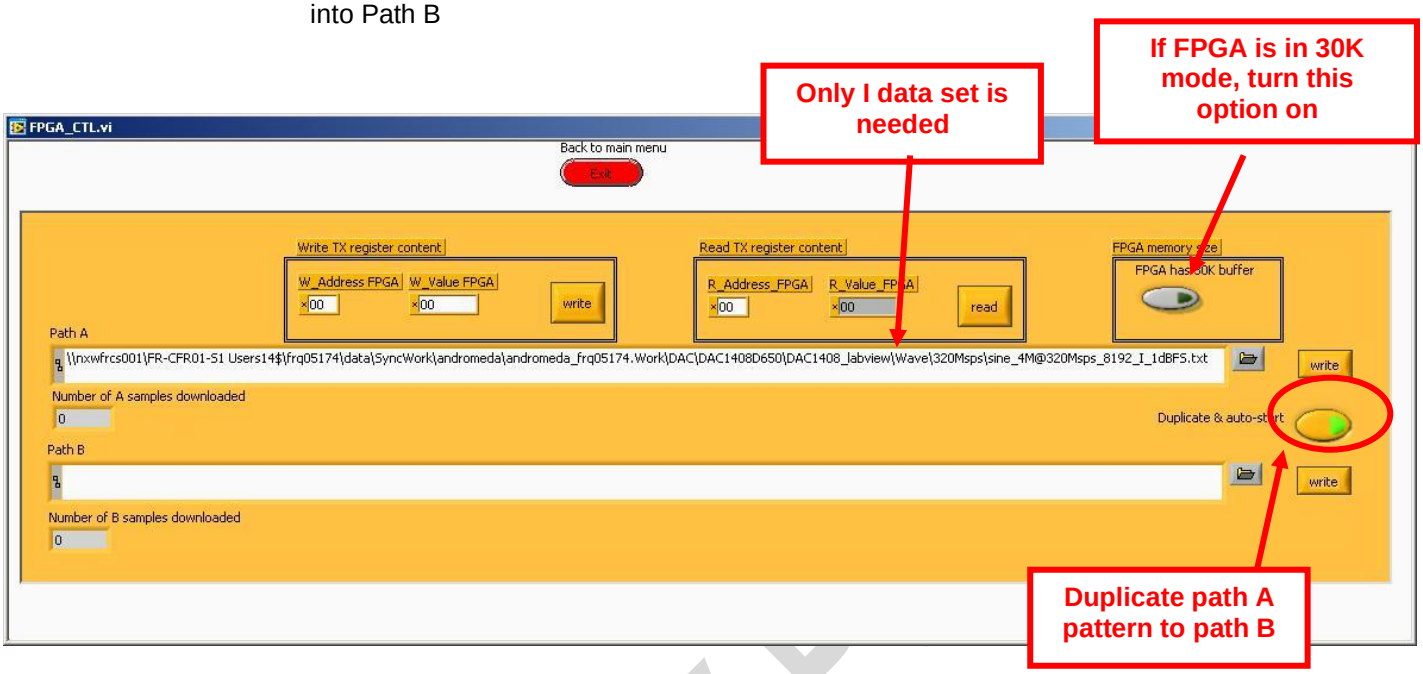
Remark: if the FPGA code selected with switch SW1 is a 30K samples, then 30K button must be on.

Remark: Clicking the "write" button automatically triggers the TX FPGA registers configuration.

Remark: Write order is important. Always write Path A data set before Path B one.

For simple signals (sine wave), one can turn the '**Duplicate & auto-start**' feature on.

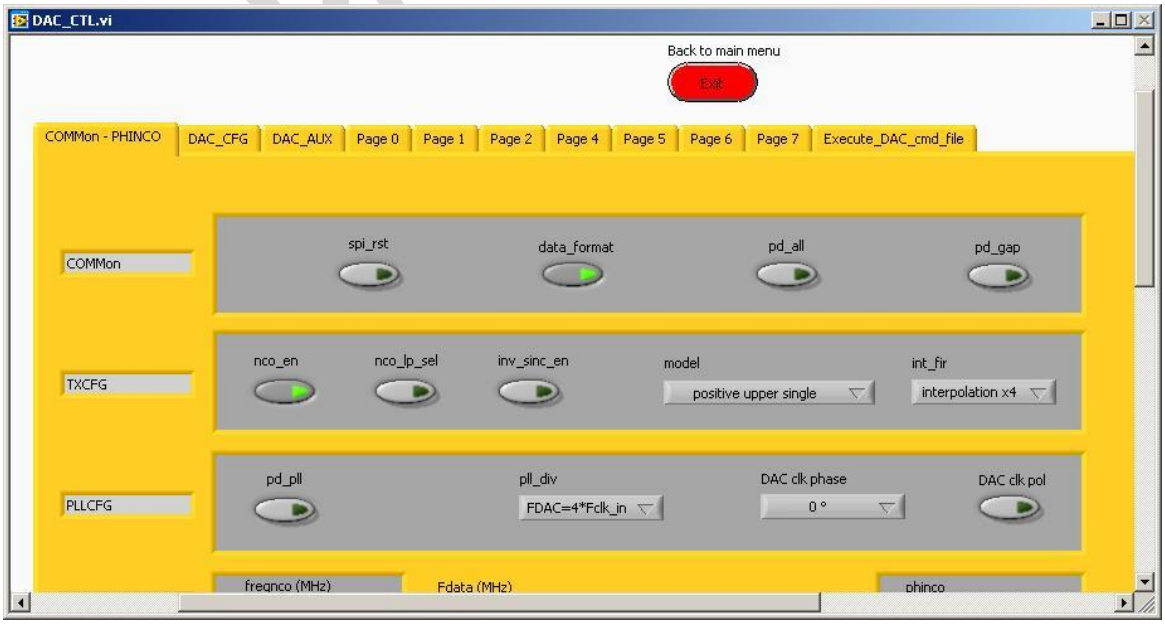
Then there is no need to load Path B, as the content of Path A is automatically copied into Path B



Once finished, close the window using the red button '**Exit**'.

- Step 5

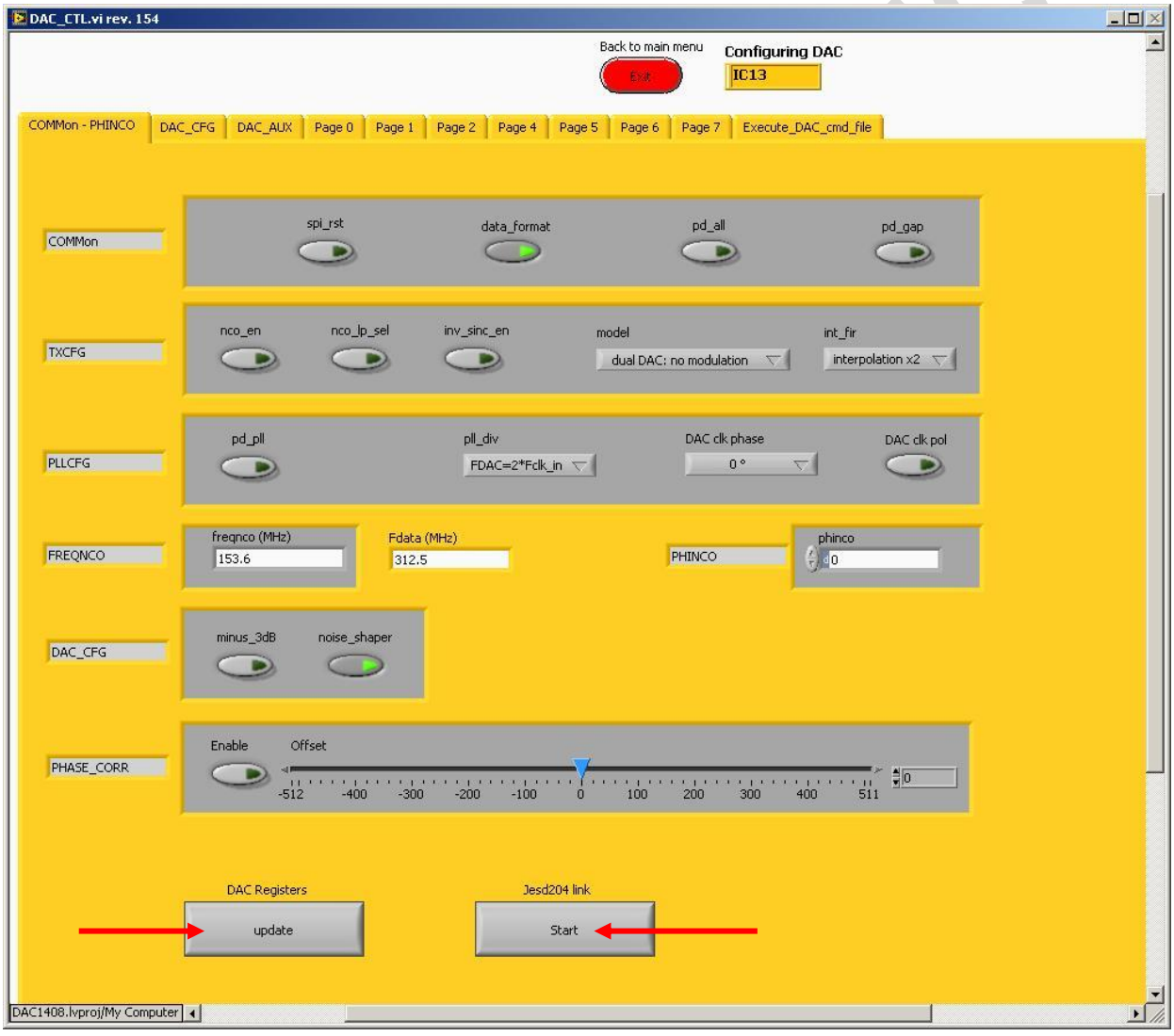
Press '**DAC_1 control**' button.



There are three ways to configure the DAC:

1. Using the first three tabs: “COMMon – PHINCO”, “DAC_CFG” and “DAC_AUX”, the user can configure the device in a graphical way,
2. Using the tabs: “Page0”, “Page1”, “Page2”, “Page3”, “Page4”, “Page5”, “Page6” and « Page7 », the user can access the device registers in detail. This requires an advanced knowledge of the device.
3. Using tab “Execute_DAC_cmd_file”, the user can run presets that configure the DAC in a known behaviour.

The first method is depicted below:

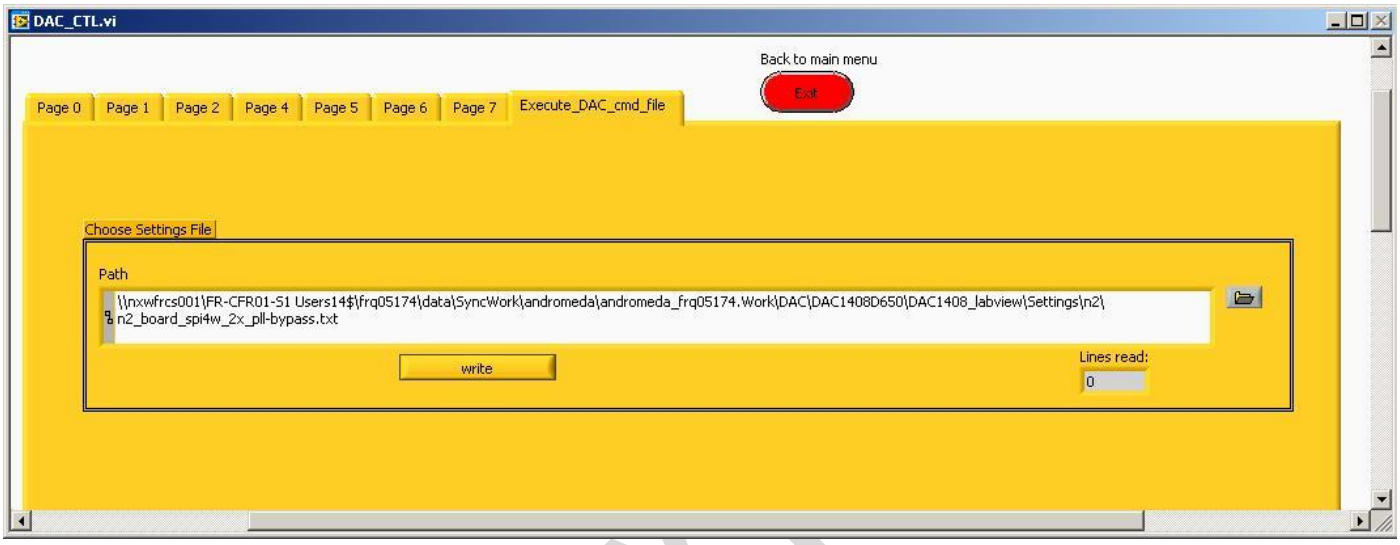


Use the buttons, sliders and numeric fields to set the DAC configuration. Then, once you are finished, press “**DAC Registers-update**” and “**Jesd204 link-update**” buttons. The expected signal should be available.

The “**Jesd204 link-update**” button needs to be pressed only once between two resets. Once the jesd204a link is running, the user can change the DAC settings (e.g. minus_3dB) without restarting the link. He just needs to press “**DAC Registers-update**” to see the changes.

The third method is accessible via the tab “**Execute_DAC_cmd_file**”.

This GUI is used to load script file that contains registers setting for the DAC. These are ASCII files that can be modified at will.



Load a command file provided with the software at the following folder:

DAC1408\Settings

Name	Size	Type	Date Modifie
n2_board_spi4w_2x_pll-bypass.txt	1 KB	Text Document	2009-10-27
n2_board_spi4w_2x_pll-bypass_ssbm.txt	1 KB	Text Document	2009-10-27
n2_board_spi4w_2x_pll-on.txt	1 KB	Text Document	2009-10-27
n2_board_spi4w_2x_pll-on_ssbm.txt	1 KB	Text Document	2009-10-27
n2_board_spi4w_4x_pll-bypass.txt	1 KB	Text Document	2009-11-24
n2_board_spi4w_4x_pll-bypass_ssbm.txt	1 KB	Text Document	2009-10-27
n2_board_spi4w_4x_pll-on.txt	1 KB	Text Document	2009-11-20
n2_board_spi4w_4x_pll-on_ssbm.txt	1 KB	Text Document	2009-10-27
n2_board_spi4w_8x_pll-bypass.txt	1 KB	Text Document	2009-10-26
n2_board_spi4w_8x_pll-on.txt	1 KB	Text Document	2009-11-25

Then press the button ‘**Write**’ to configure the device.

Led D9 should now turn off to indicate that the Jesd204A SYNC request signal is de-asserted. This means that the DAC and the FPGA are now well synchronized.

The naming convention for the scripts files is the following:

- 2x: 2 times interpolation filter turned on,
- 4x: 4 times interpolation filter turned on,

- 8x: 8 times interpolation filter turned on,
- pll-on: DAC's internal pll is used to generate output sample rate,
- pll-bypass: output sample rate has to be provided from the main synthesizer,
- ssbm: DAC's internal NCO plus single side band modulator is turned on,

Once finished, close the window using the red button '**Exit**'.

- Step 6

Press '**DAC_2 control**' button.

Proceed like Step 5 to configure the second DAC labeled IC23.

Signals will only be available at DACs outputs when both devices are configured. This is because the SYNC_REQUEST signal of each DAC is combined inside the TX FPGA, as stated in the jesd204a specification.

Led D18 should now turn off to indicate that the Jesd204A SYNC request signal is de-asserted. This means that the DAC and the FPGA are now well synchronized.

Once finished, close the window using the red button '**Exit**'.

4. Annex 1: default dip-switches and jumpers settings

Table 5. SW1 default
FPGA flash

1	2
ON	ON

Table 6. SW2 default
IC10 registers settings

1	2
ON	OFF

Table 7. SW3 default
Main clock input

1	2
ON	ON

Table 8. SW4 default
FPGA configuration

1	2	3	4	5	6	7	8
OFF	ON	OFF	ON	ON	ON	ON	ON

Table 9. SW5 default
IC19 configuration

1	2
OFF	OFF

Table 10. ST2 default
SPI 3W/4W mode

1
OFF

Table 11. ST1 default

Optional VCXO power supply

1
ON

5. Annex 2: Troubleshooting

5.1 Multiples FT2232 devices connected to the host PC

DAC1x08D demo board features a FT2232D USB to SPI bridge IC.

When multiple boards featuring the same IC are connected to the host PC, the software is not able to differentiate them. Commands shall be sent to the wrong system.

To prevent this, one should make sure that there is only one FT2232D device connected to the host at a time.

5.2 Sanity checks if the system doesn't generate the expected waveform

1. Is the FLASH led (D1) on?
   If not, check SW1 switch setting,
2. Is the USB led (D7) on?
   If not, then unplug and plug the USB cable from the host PC,
3. Is main synthesizer led (D2) on?
   If not, then one needs to retune the pll dividers settings (refer to SPI interface). Make sure that the 'VCO out of range' indicator isn't on.
4. Are the FPGA leds (D9, D10, D11, D12, D13, D18, D19, D20, D21, D22) on?
   If not, press main reset push button (BP1).
   If this doesn't solve the issue, check switches SW3 according to the clock source used.
5. Are the jesd204a links synchronized?
   After configuring the FPGA and both DACs (refer to SPI interface), led D9 and D18 should be off to signify that all links are synchronized.
   If one of them is still on, this is probably due to improper clock configuration.
6. Every led is on or off as expected, but the output spectrum does not look good.
   Check how the pattern memory size has been configured on the board (switch SW1) and in the software (tab 'FPGA control'). Both settings should be aligned, i.e. 8K or 30K.

6. Index

7. Contents