Introduction :

This software is a ready to use software to perform the slow control for the chip Picmic.

This software was written in python by:

- Gilles Claus for the low level functions,
- Hugo Schott for the GUI,
- Matthieu Specht for the high level functions.

This software was tested on Windows XP, 7 and 10.

To be working on Windows XP, please find the procedure in the annexes.

The current version of this software if the 0.3.0.

Prerequisite : Python installation

This software is a python script providing a GUI for the slow control of the Picmic chip. In order to have it running correctly, you need to have python installed on your computer, along with the needed modules.

This is the guarantied working configuration:

- Python version 3.7.6
- Pyqt5 version 5.14.1
- Matplotlib 3.1.3
- Dwf 0.1.0
- Pyfirmata 1.1.0

We are using the package and environment manager miniconda to install all the needed software .

You can find the documentation about miniconda on this website :

https://docs.conda.io/projects/conda/en/latest/index.html

Here is the procedure we used to install. After installing the miniconda package (running the Miniconda3-latest-Windows-x86.exe), start an Anaconda Prompt and create your virtual environment:

conda create --name *ENVNAME* python=3.7.6 matplotlib

Activate your virtual environnment

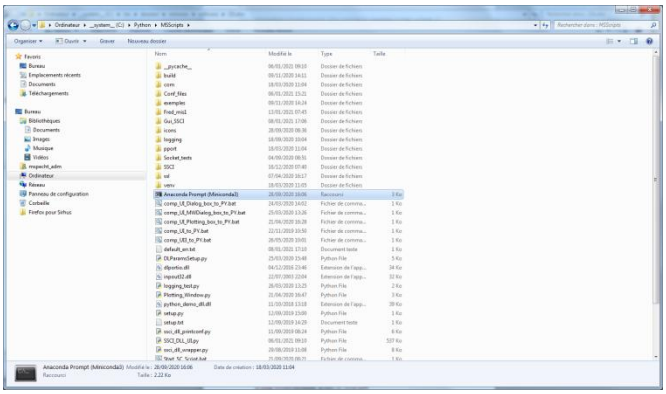
conda activate *ENVNAME*

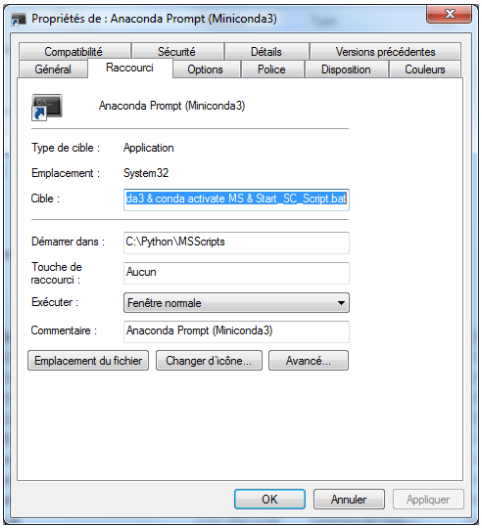
install the necessary modules : PyQt5, dwf and pyfirmata

pip install pyqt5 dwf pyfirmata

Once installed you can modify the properties of the Anaconda prompt to allow it to start directly the software when started,

In the Picmic0_SC directory, change the properties of the Anaconda Prompt shortcut





Add the following to the target field

& conda activate ENVNAME & Start_SC_Script.bat

The Software can now be started once the Arduino drivers have been installed.

Prerequisite : Arduino software installation

In order to have the driver installed for the Arduino board that performs the slow control for picmic, the Arduino software have to be installed on the system. The version used by now is the version 1.8.9.

Just run the arduino-1.8.9-windows.exe to install the Arduino software along with the Arduino boards drivers.

Any information can be found on the Arduino web page :

https://www.arduino.cc/

Starting the software:

This software is basically a python script, you can start with the following command:

python sc_picmic_GUI_xxx.py

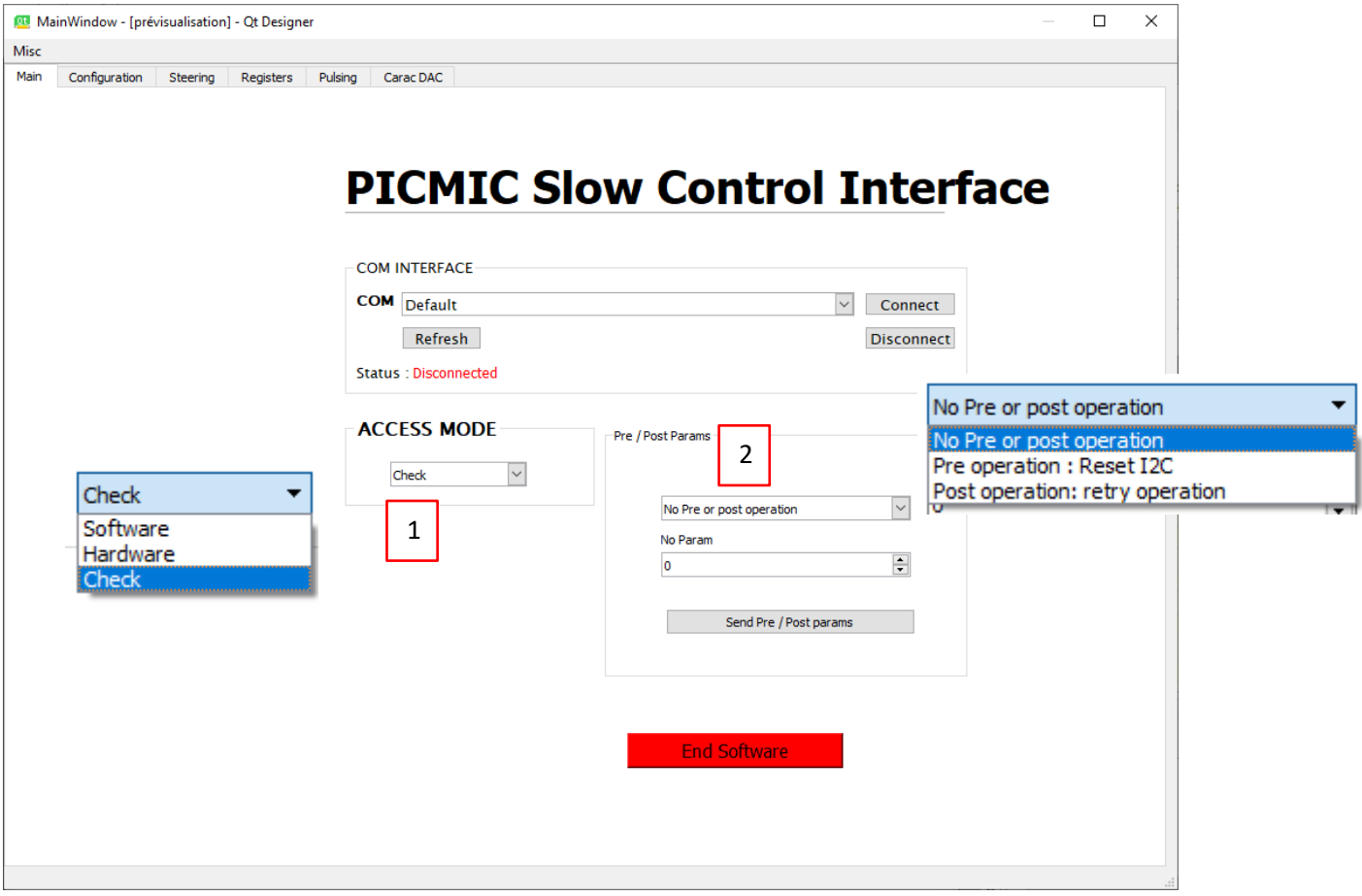
With xxx the current version nb; the version number at the writing of the documentation is the version 041

The software should start as shown below:

You have two selectable options before connecting to the Arduino board :

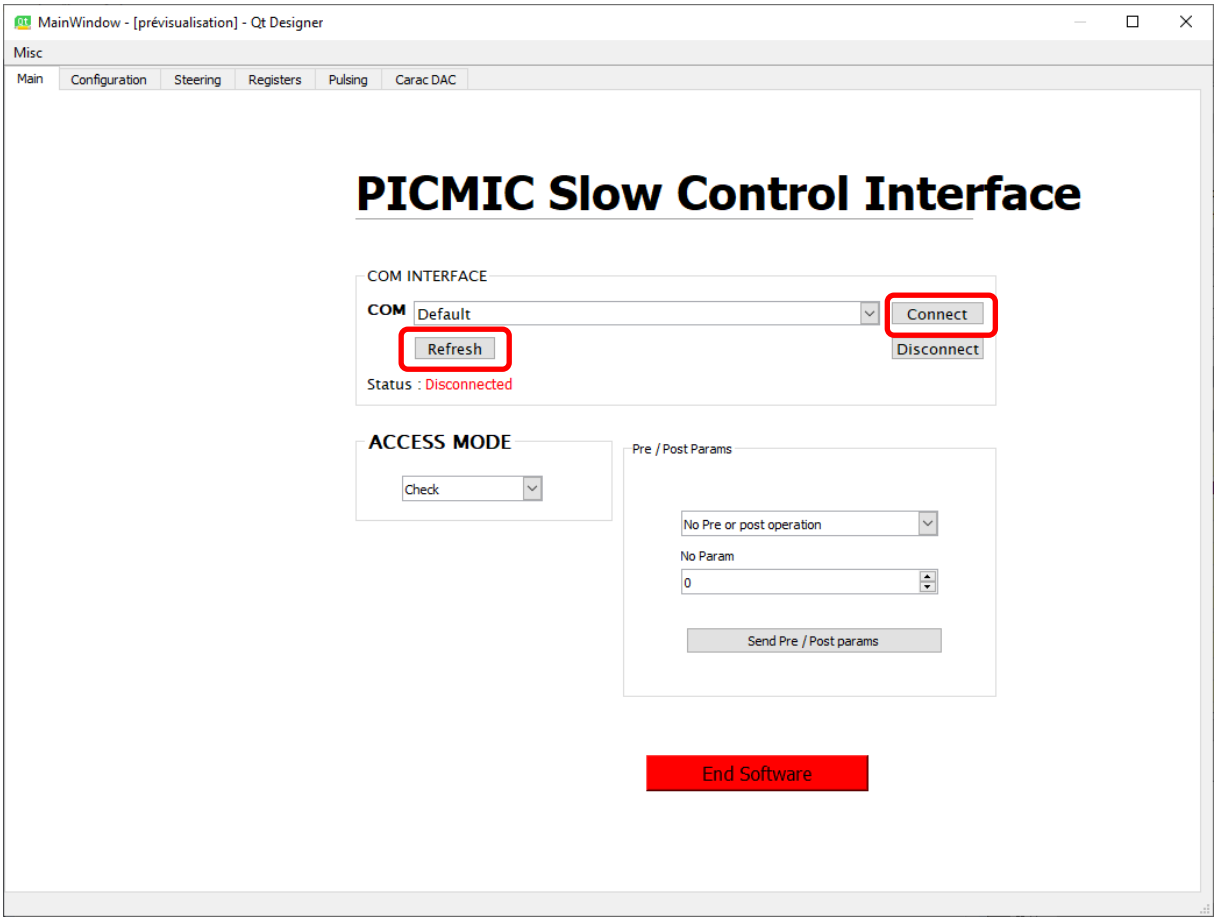
- 1 : Access mode : three options:
  - Software: all I2C communication are made with the memory image of the chip
    - No communication are made with the real chip
  - Hardware: all I2C communications are made with the chip
  - Check: all I2C communications are made with the chip
    - Each byte written is read back and compared

- 2 : Pre/Post operation : three options:
  - No pre or post operation
  - Pre operation: before each I2C transaction a reset_I2C is asserted
    - Param : reset duration in ms
  - Post operation: each I2C transaction is repeated until it is successful
    - Param: maximum number of retries

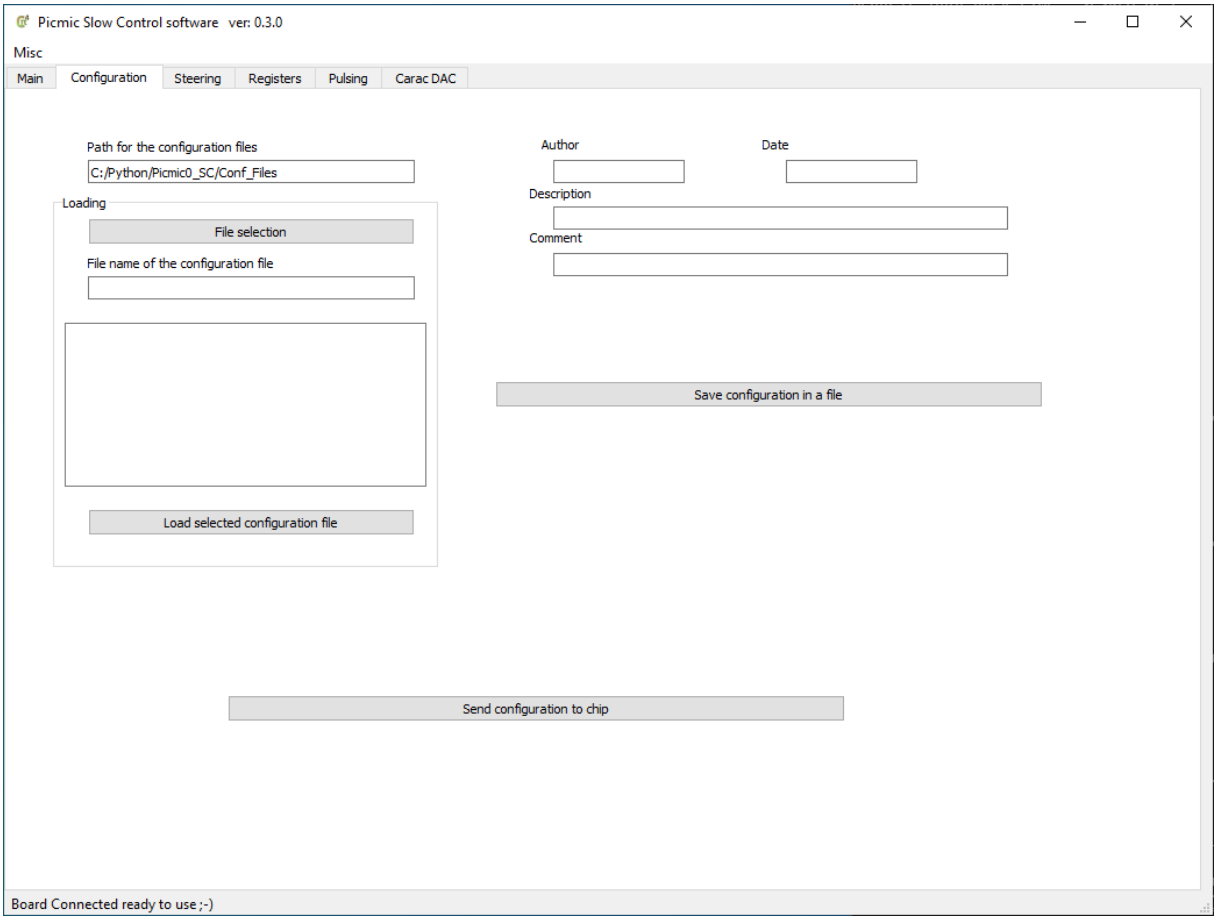


Then you can push the *Refresh* button to check the connected boards, and if your usb cable is correctly conned, the right com port will be selected.

Then you can push the connect button to activate the connection.

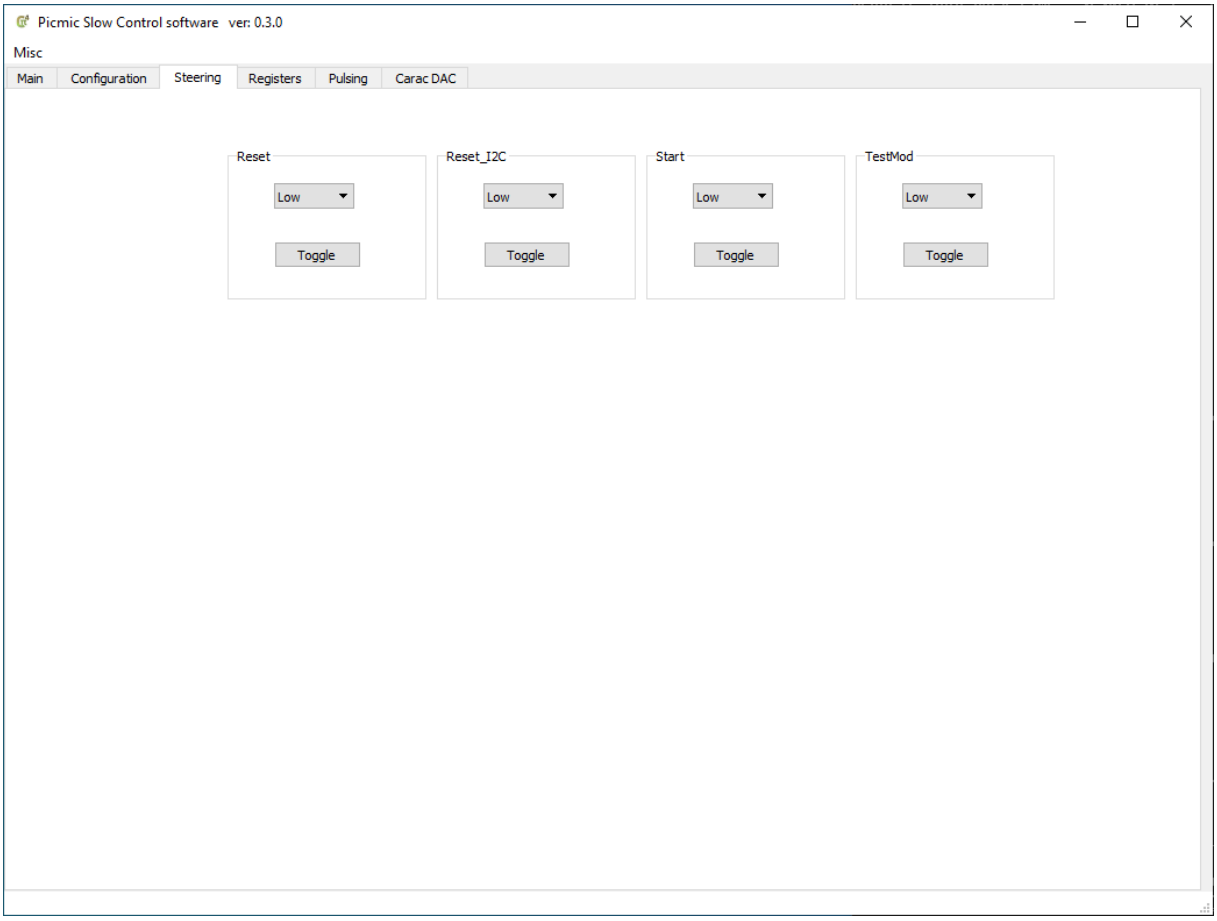


Configuration Tab:



- 1 : File selection: click on this button to select the configuration file you want to load
    - Once selected, you ll have the description of the configuration file selected in the Text edit field
- 2 : Load selected configuration file : this button triggers the loading of the selected configuration file, and the file params will be displayed in the top right fields.
- 3 : Send configuration to chip: this button will send all the registers displayed in the software to the picmic chip
- 4 : Save configuration in a file: the button will save the current configuration to a file.
    - Please do not forget to modify the params fields (top right fields).

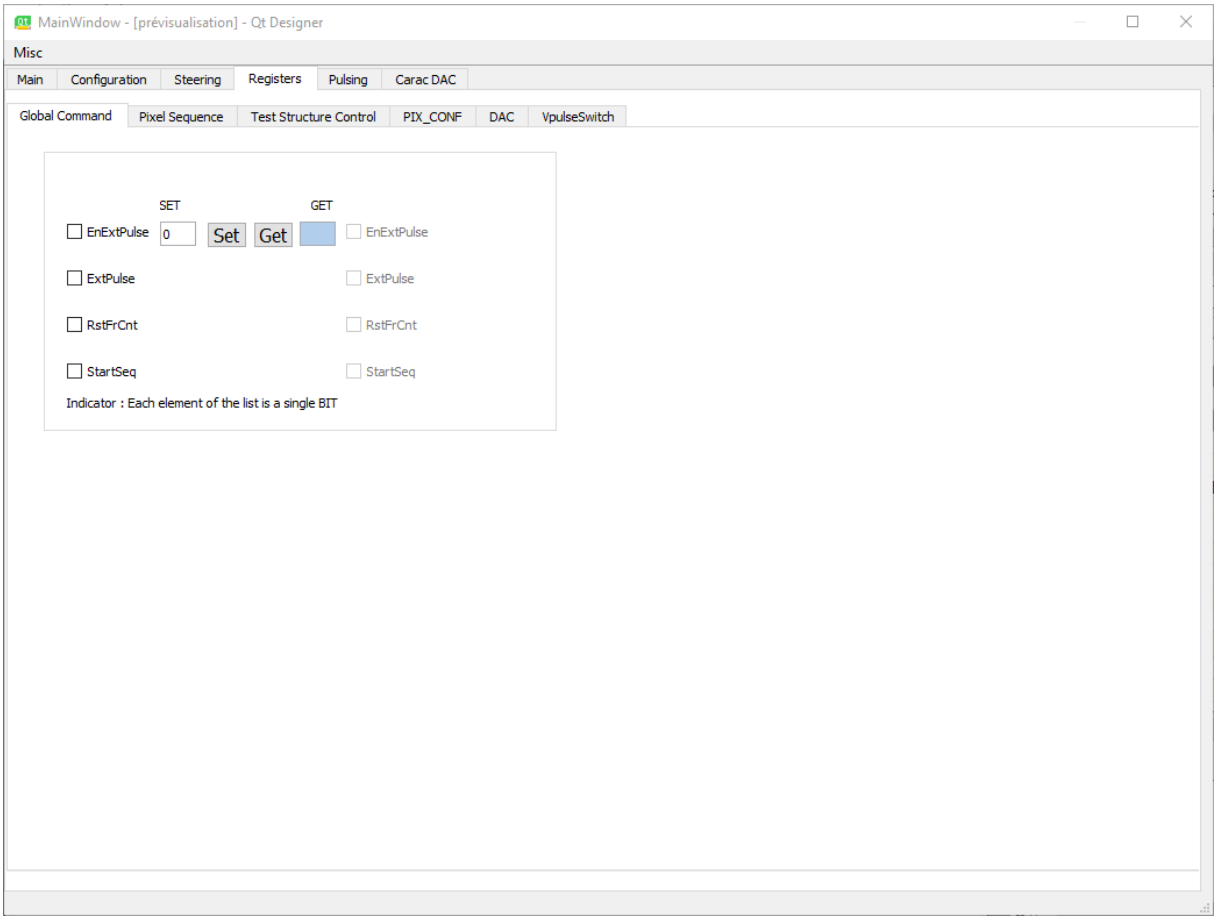
Steering Tab:



In this tab, you can change the levels of the steering signals:

1. Reset : Head Reset for the picmic chip
2. Reset_I2C: Reset for the I2C state machine inside picmic
3. Start : hardware start signal (note, the start can be programmatically set, see the registers tab)
4. Testmod: Test mod signal, not used yet.

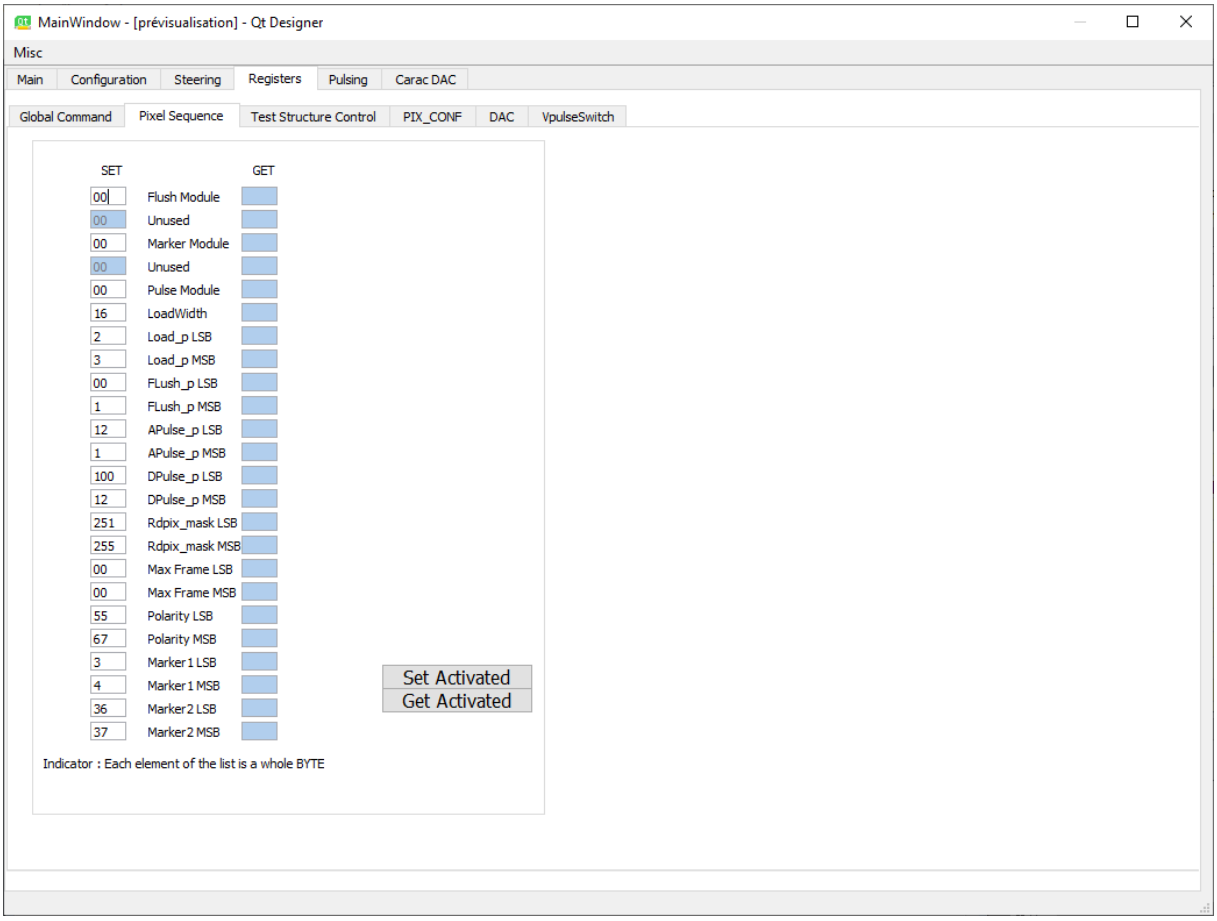
Registers Tab: Global Command



The values can be either set as a byte, or bit by bit.

The set button will send the byte value to picmic, and the get button will read back the value of picmic register and display it.

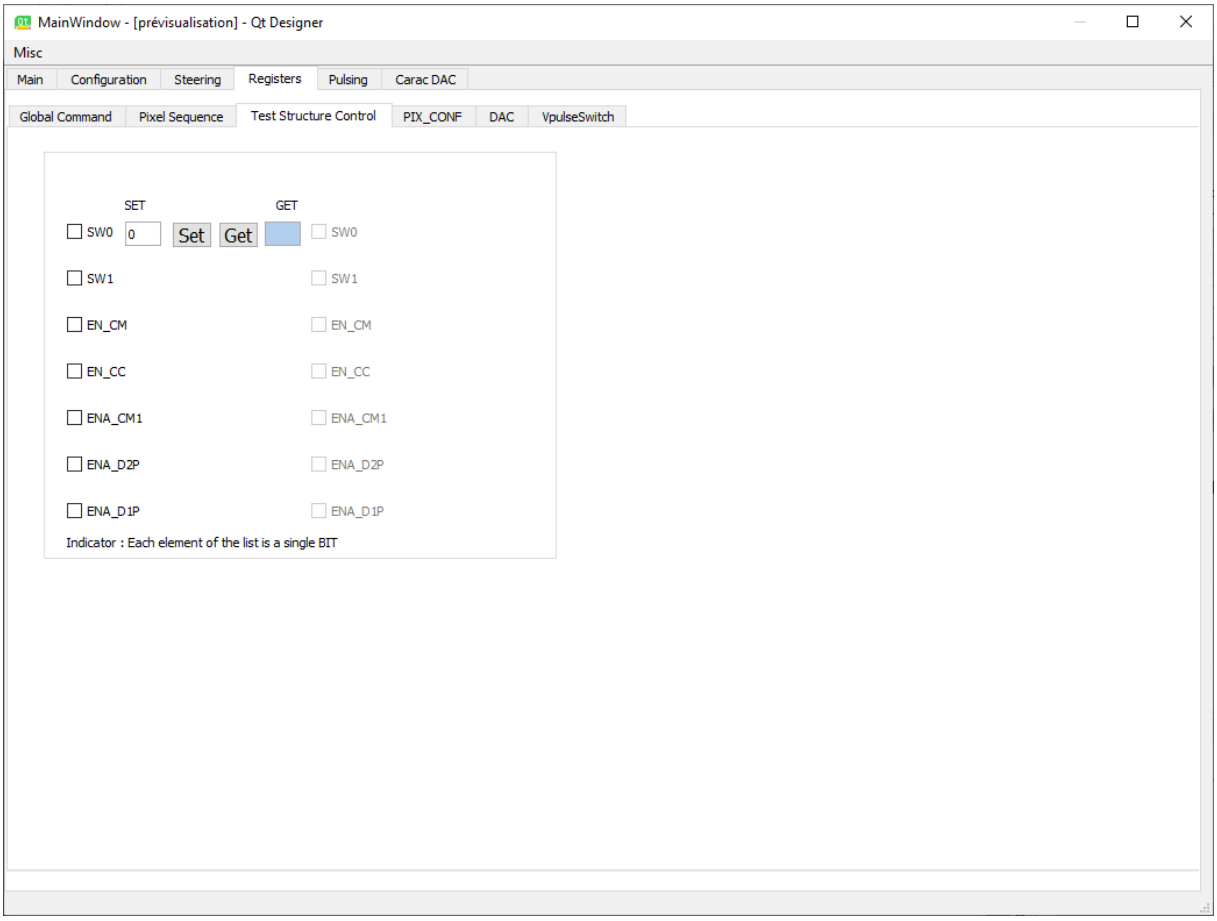
Register Tab:Pixel sequence



The set activated button will send all the registers values to picmic

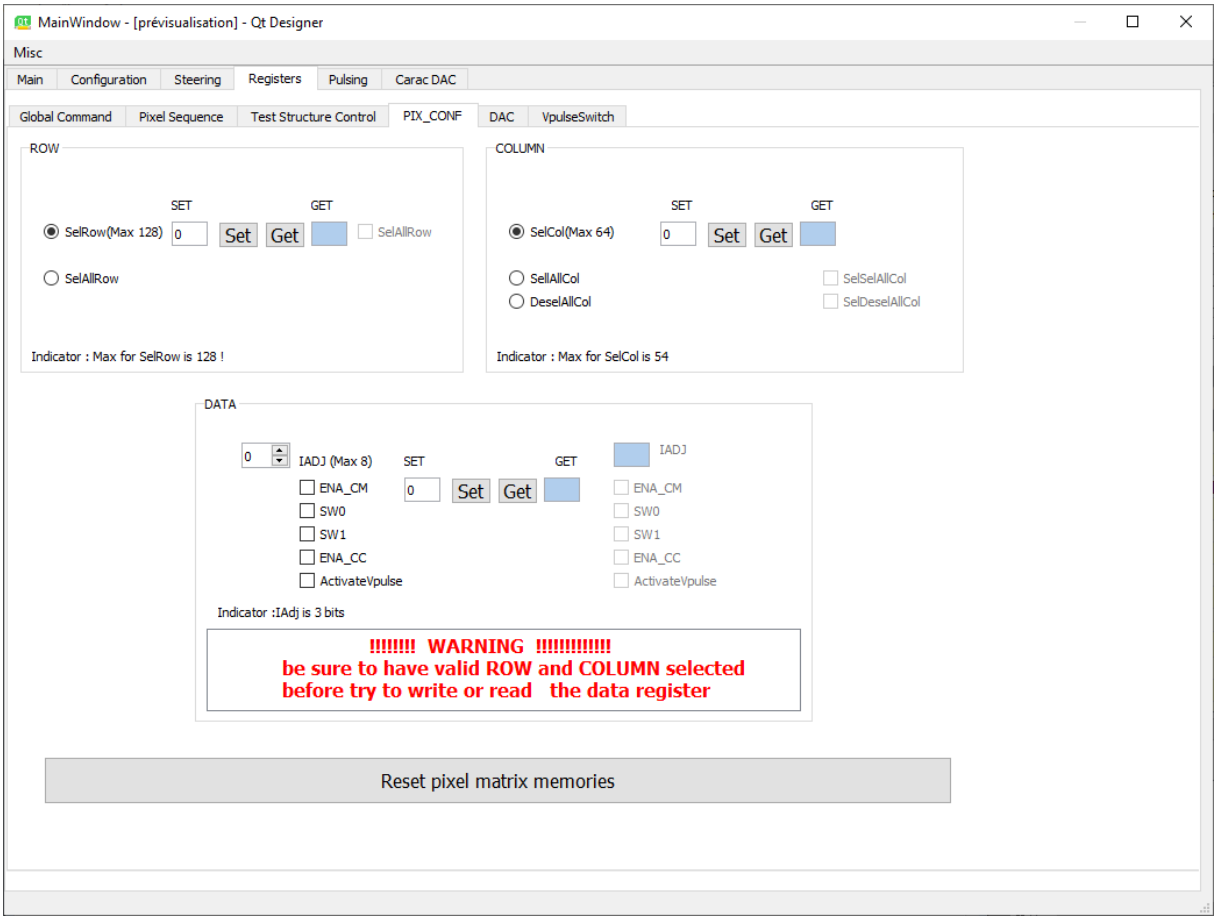
The Get avtivated button will read all the registers values from picmic

Register Tab:Test Structure



The set button will send the byte value to picmic, and the get button will read back the value of picmic register and display it.

Register Tab:Pixel Config

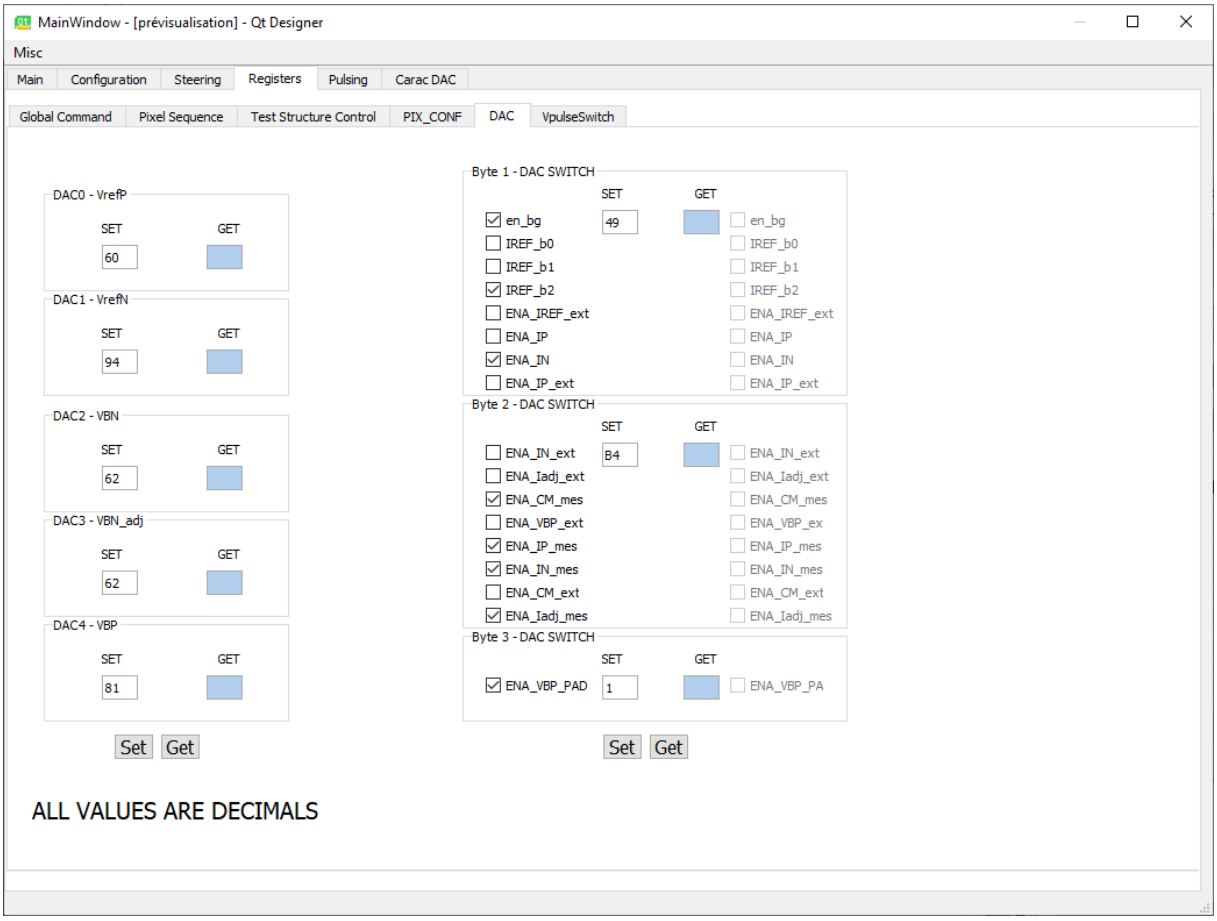


On this tab, you can set the pixel memory registers.

The pixel memories are not reset at startup, so it is advised to set the pixel memory registers to a know value.

When you click on the ***Reset pixel matrix memories*** button, the value of the byte (1) is sent to all the pixel memories.

Registers Tab: DAC



For the dacs (left part of the tab):

The set button (1) will send all the dac values to picmic

The Get button (2) will read all the dac registers from picmic and display them

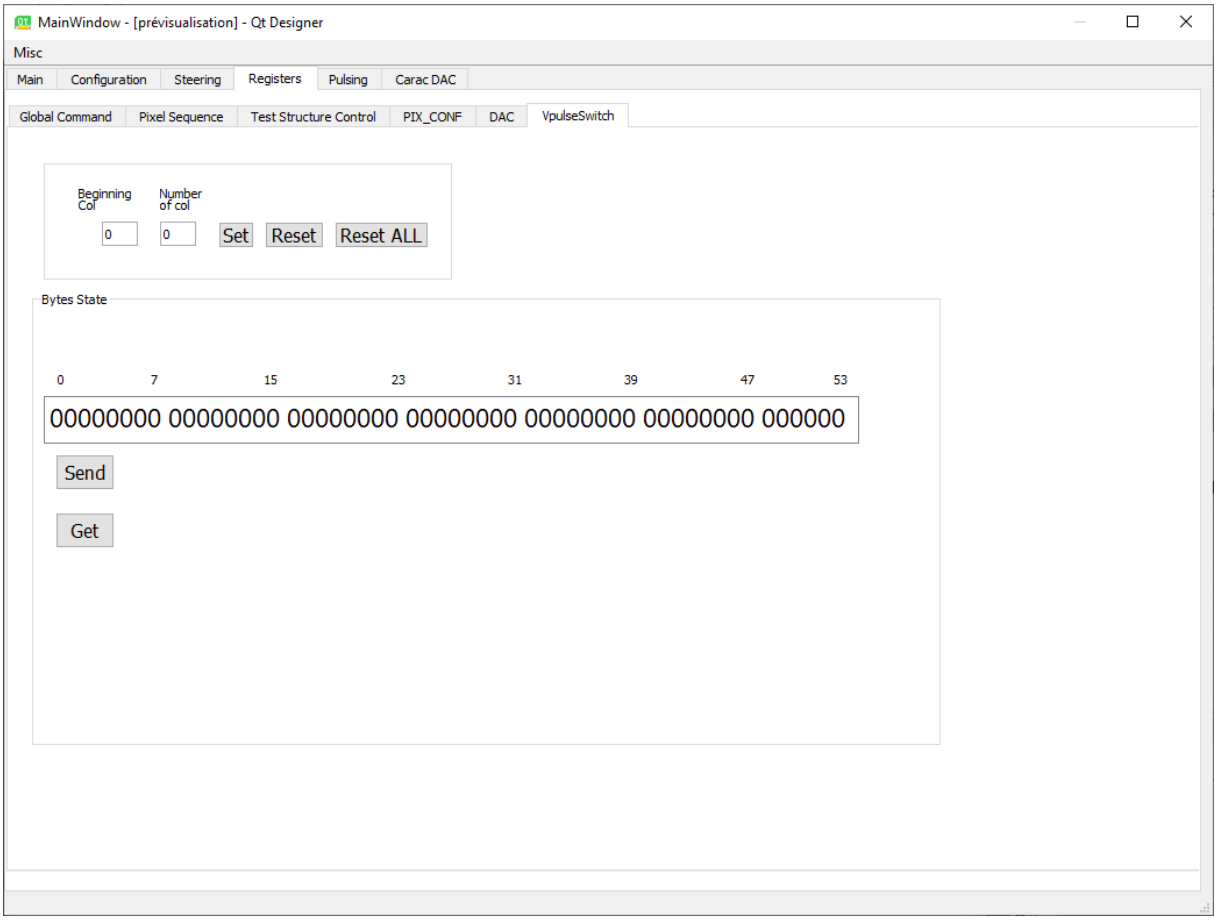
For the Dac Switches registers(right part of the tab)

The values can be either set as a byte, or bit by bit.

The set button will send the three byte values to picmic

The get button will read back the three value of picmic register and display them.

Registers Tab: VPulse switch



For the column selection (top of the tab):

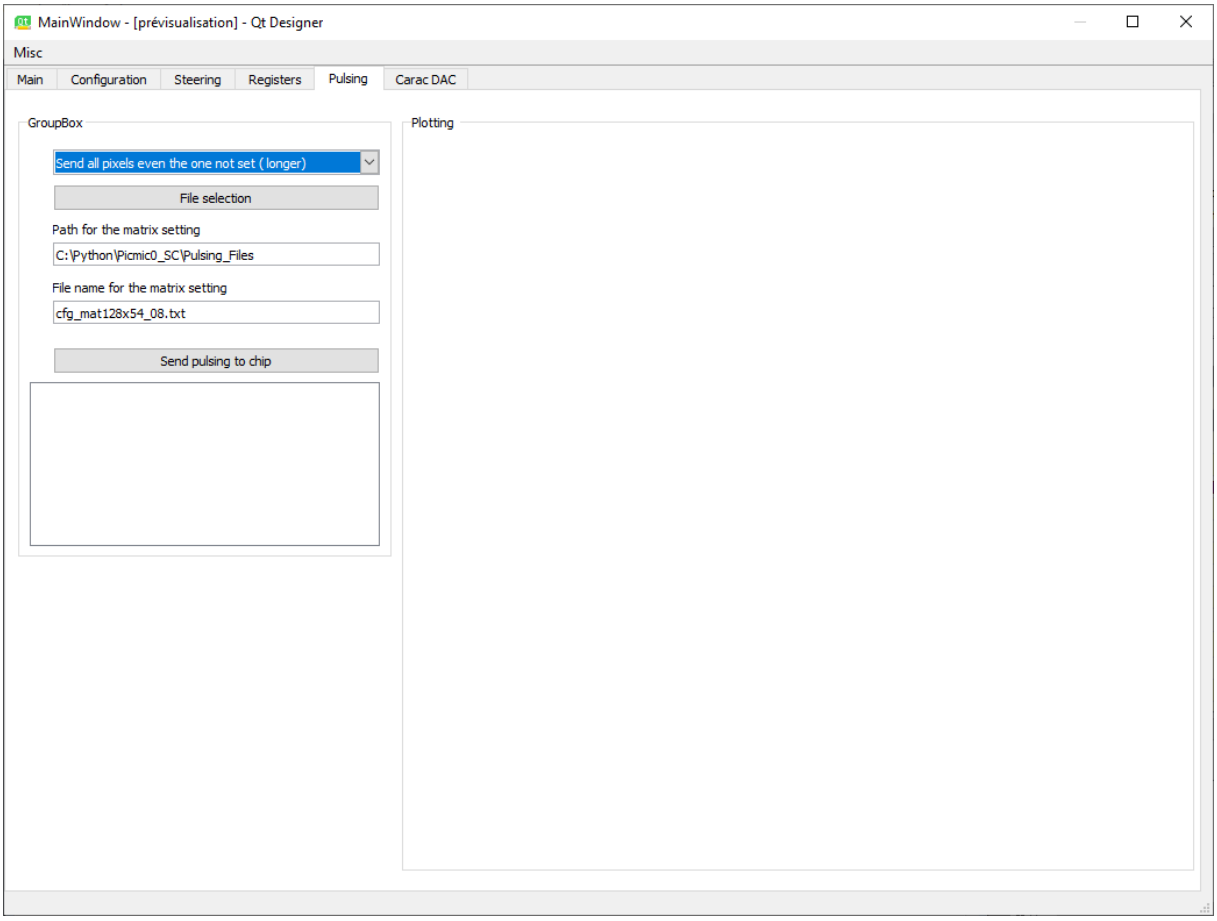
You can select the first column, and the number of column targeted and either set or reset them.

Then the choice you make will be displayed in the byte state field.

The send button (1) will send the values displayed in the field to the picmic chip

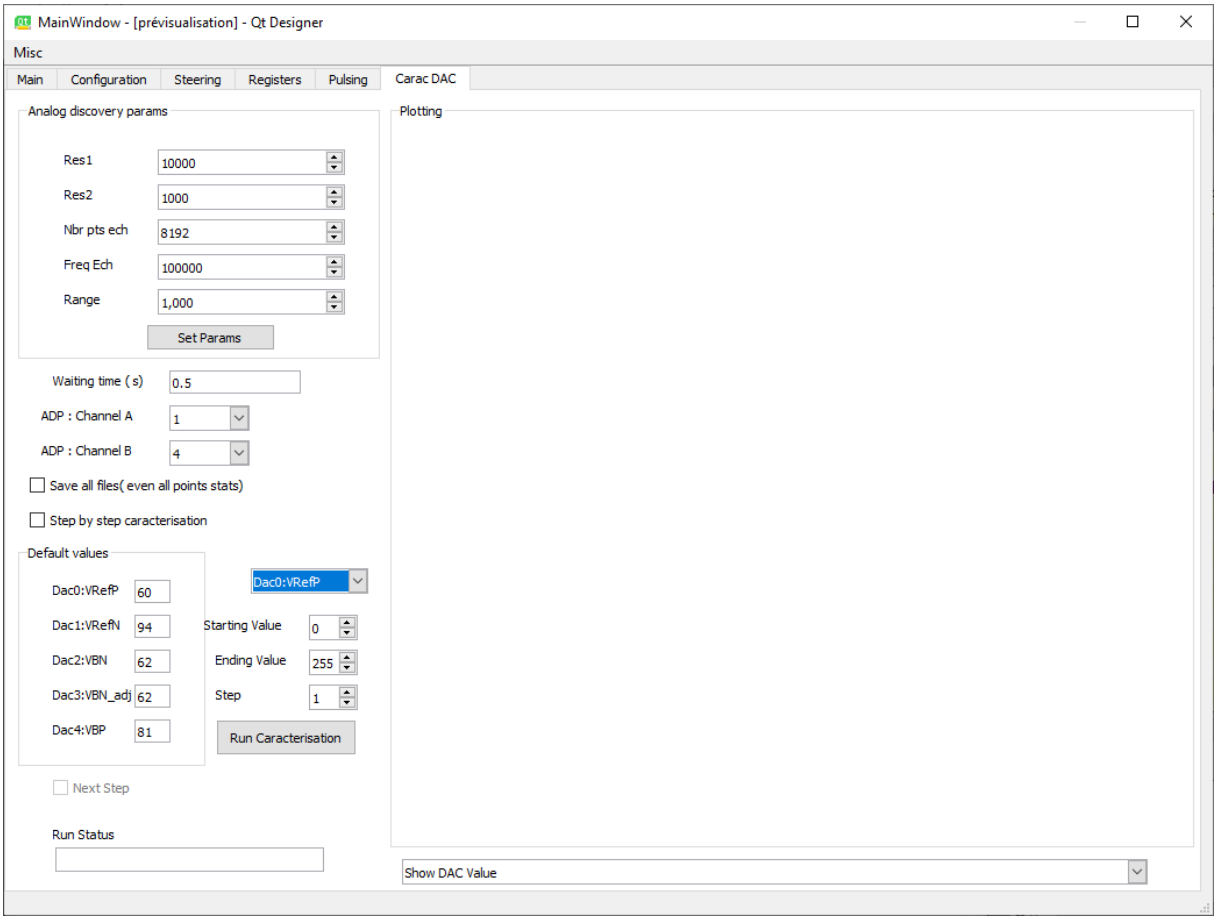
The get button will read the picmic values and display the m in the field

Pulsing Tab:



- 1 : File selection : when you select a file for the pulsing, the params of the file will be displayed in the text field
- 2 : Select pixel to configure: either you can :
    - Send all the pixels even the pixels not set. This will take more time to complete
    - Send only the pixels to be set
- 3 : Send pulsing to chip:
    - The selected pixels will be sent to the chip, it can take several seconds to complete
    - The resulting pulsed matrix will be shown on the right side of the tab

Dac caracterisation Tab:



Analog discovery params:

- Res1 : external resistor value put on the dac :0,1,3,4
- Res2 : external resistor value put on .the dac 2
- Nbr Pts Ech : number of values taken for one dac value
- Frequ ech : sampling frequency for the ADC of the analog discovery board
- Range : Range of the analog data (in volts)

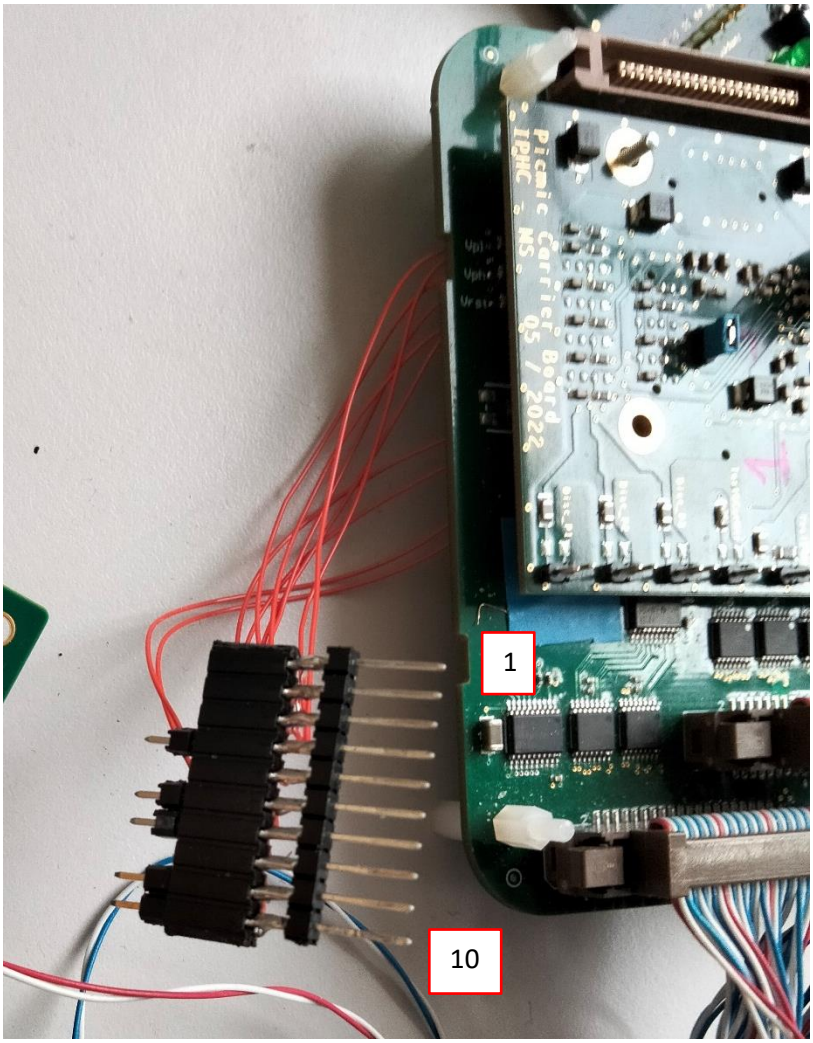
Once the params have been changed, do not forget to push the set params button.

Waiting time: time between the setting of the dac and the starting of the sampling

Once the caracterisation is completed, the DAc value will be shown in the right field.

Note that on windows XP system, the Result will be shown in an external window.

Dac caracterisation Hardware:



Pinout of the connector:

- 1 : VRefP(GND)
- 2 : VRefP (signal)
- 3 : VRefN (signal)
- 4 : VRefN(GND)
- 5 : VBN(signal)
- 6 : VBN(GND)
- 7 : VBN-adj(GND)
- 8 : VBN_adj (signal)
- 9 : VBP (signal)
- 10 : VBP (GND)