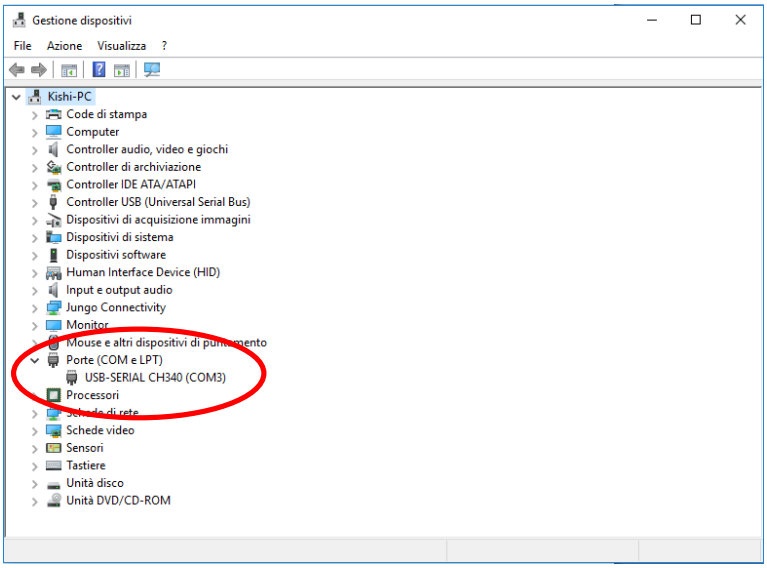
Davide Cavaliere
www.monocilindro.com
dadez87@gmail.com
11th March 2017

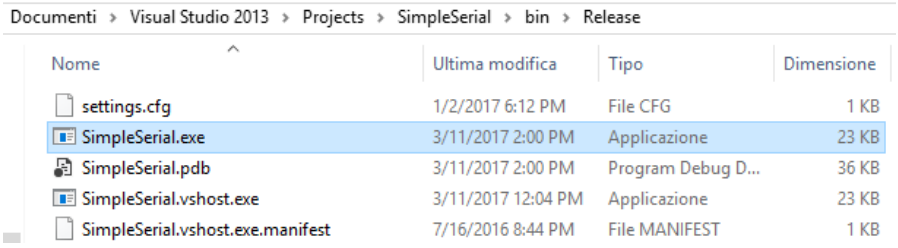
Fuelino Calibration Tool Guide

This guide will explain how to use the *Fuelino Calibration Tool*.

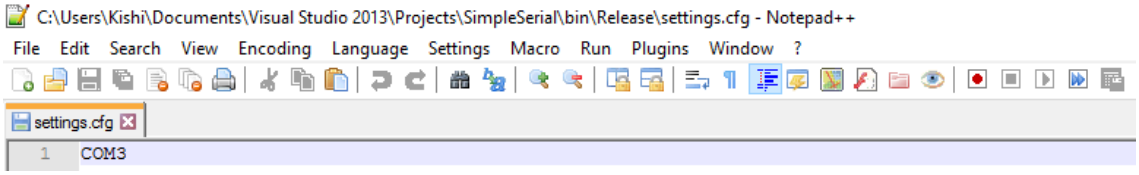
First of all, connect Fuelino to the PC using a USB cable. Then, in *"Device Manager"*, you should check the Serial Port number assigned to Fuelino (Arduino Nano). In my case, it was assigned *COM3*, as shown below.



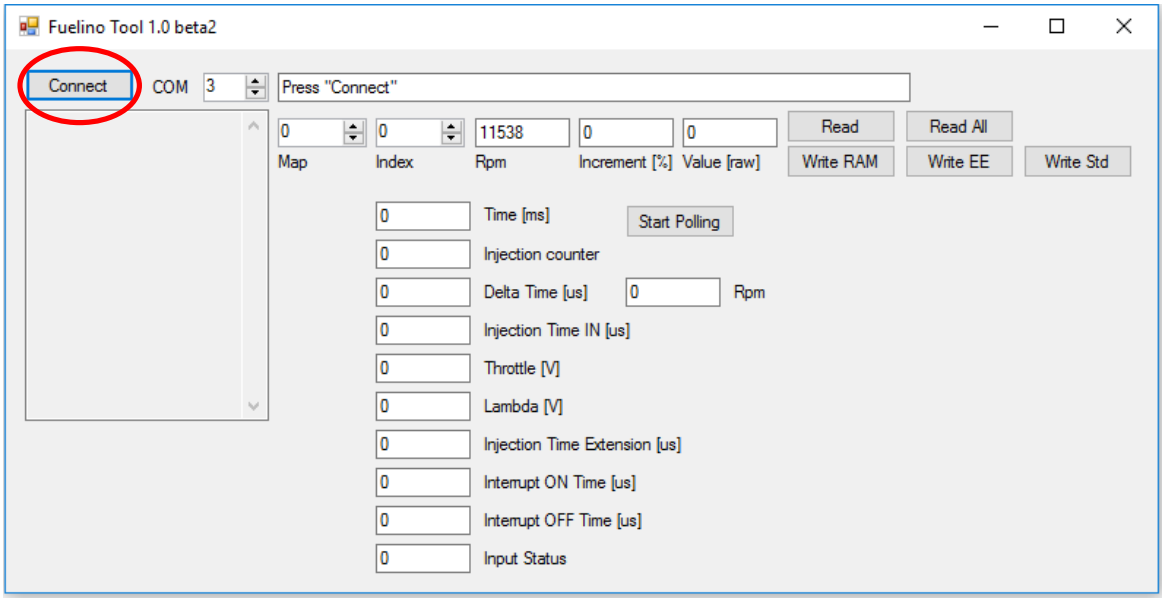
Open the file *"settings.cfg"* using Notepad (or Notepad++ or similar).



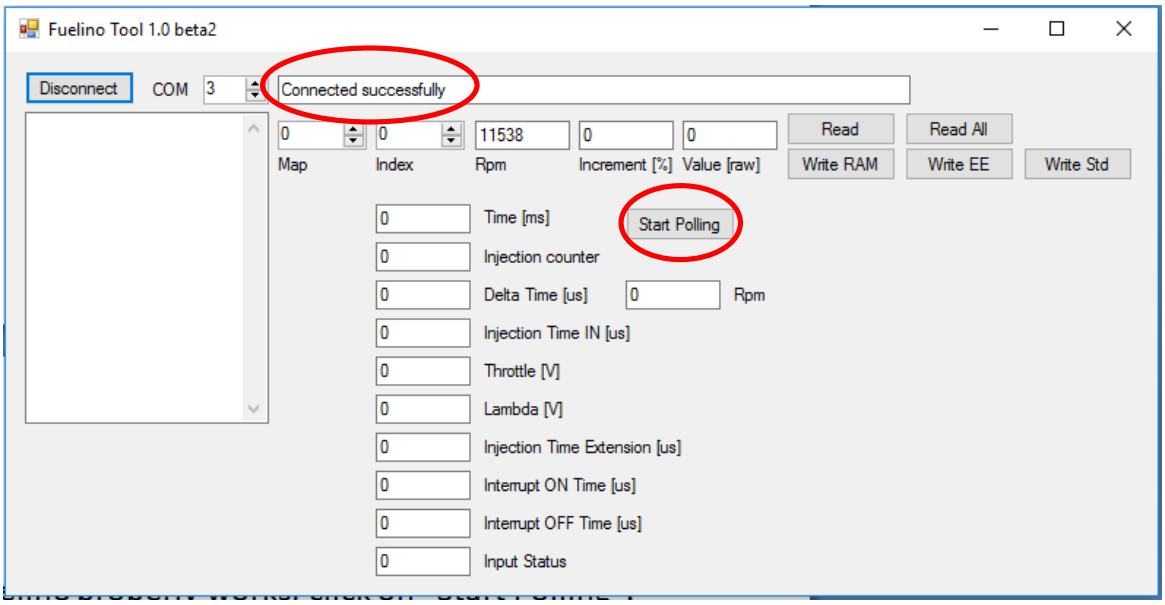
Set the Serial Port number of the previous step (*COM3* in my case). Save and exit.



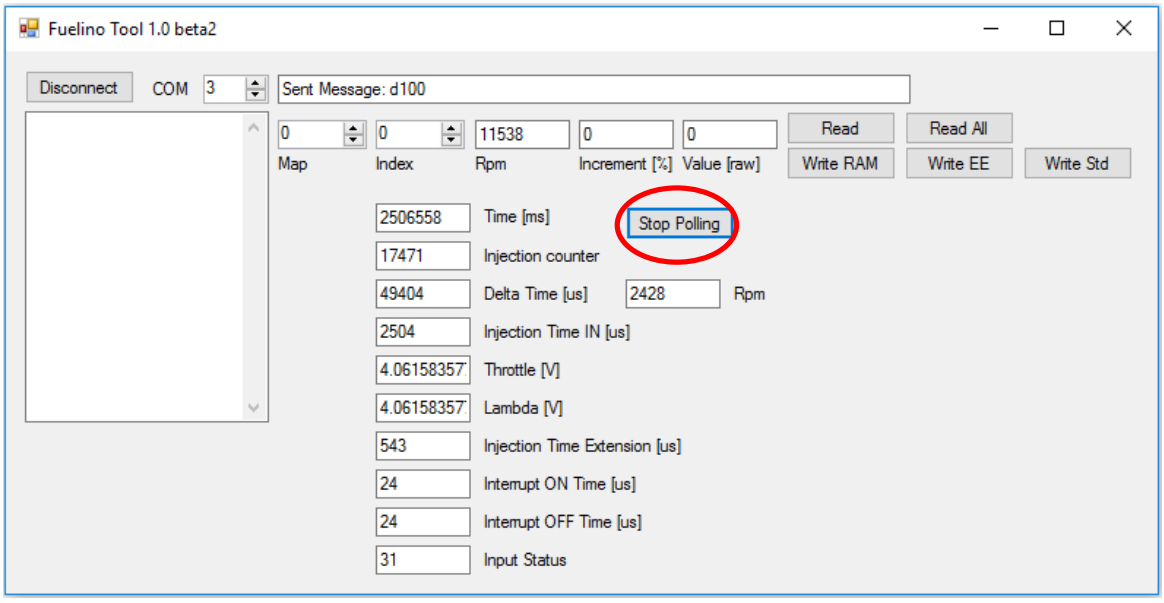
Execute the file *"SimpleSerial.exe"*. The following screen appears.



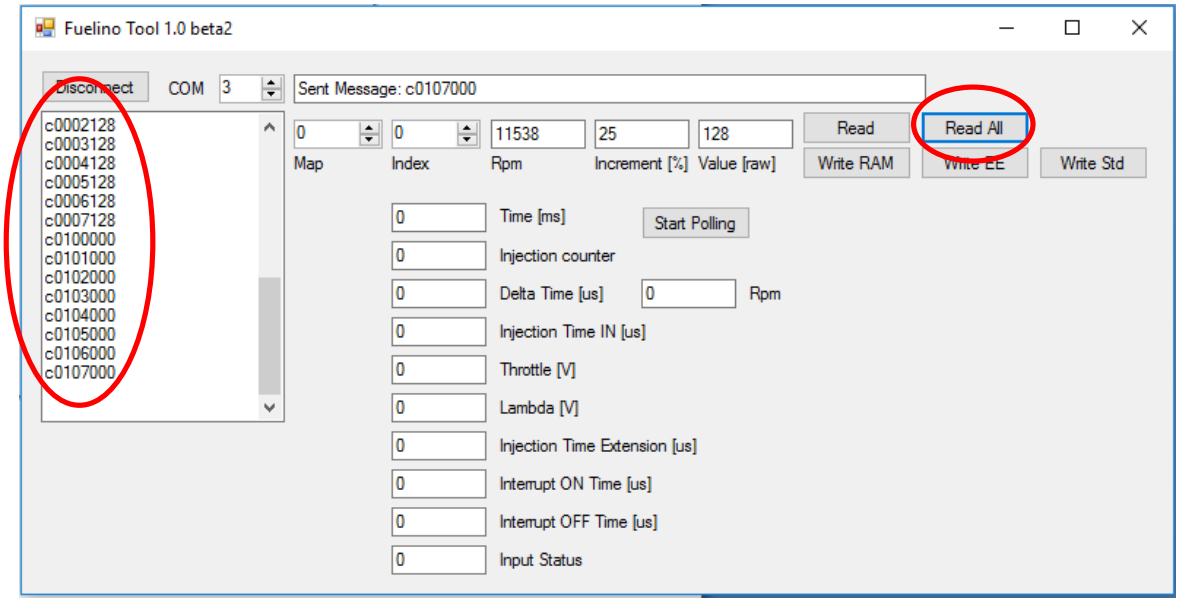
Make sure that the Serial Port is correct (*COM3* in my case). Then, click on "*Connect*". In case the Serial Port exists, it is not busy, and the tool can correctly reserve it, the message *"Connected successfully"* will be shown.



If you want to make sure that the communication with Fuelino properly works, click on “Start Polling”. The tool will start showing the data acquired in real time, such as the *Time[ms]* from last Power On. In case you see that the time is properly updated, it means that the serial connection with Fuelino is properly working (i.e. the tool can read data from Fuelino, in real time). In case it is not working, try to push the *“reset”* button on the Arduino Nano.



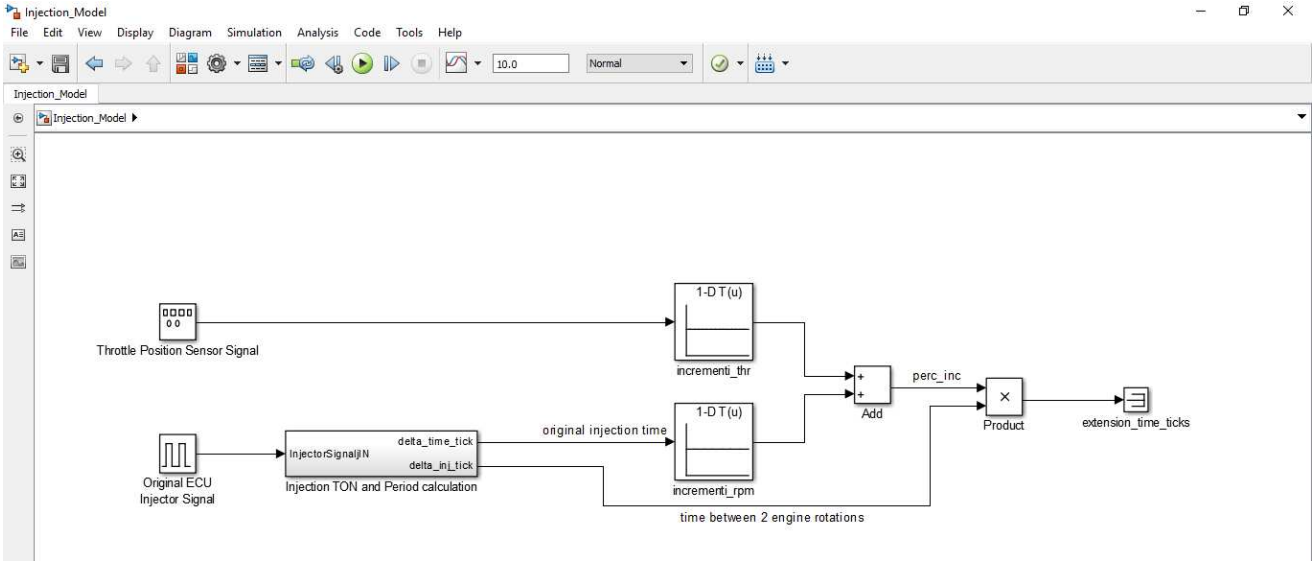
If you want to read the Calibration maps values stored in Arduino, first of all stop polling. You should at first click on *“Stop Polling”*. Then, click on *“Read All”*. The tool is reading the values of the calibration maps stored inside Fuelino. On the left side, you can see the raw messages received from Fuelino. On the right side, you can check the calibration values of each map and index.



There are 2 maps:

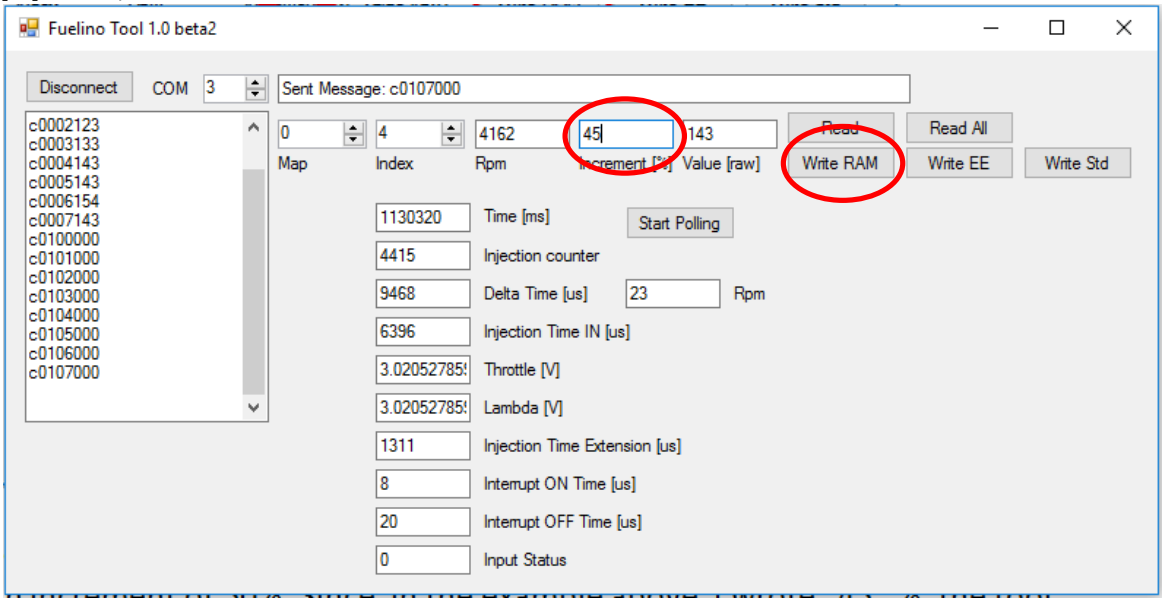
- **Incrementi_rpm[]**, “map 0”. 8 values (index 0-7). This 1-D map defines the percentage of increment (%), in terms of injection time percentage, depending on the engine speed (rpm). There are 8 breakpoints (rpm) which are defined inside the source file “INJmgr.cpp”.
- **Incrementi_thr[]**, “map 1”. 8 values (index 0-7). This 1-D map defines the percentage of increment (%), in terms of injection time percentage, depending on the throttle voltage (V). The throttle voltage acquired goes from 0V to 5V. There are 8 breakpoints (index 0-7). Each one corresponds to a voltage equal to (5V/8)*index.

The injection time extension is calculated based on the following logic. At first, Fuelino checks the values of engine speed (rpm) and throttle position sensor (V). Then, for each signal, checks which is the increment calibrated in the map. The total increment programmed is the sum of the 2 increments (increment depending on rpm, plus increment depending on throttle). If you are not using the throttle signal, you should set the calibration values of “map 1” indexes to 0.
For rpm and throttle values between 2 breakpoints, Fuelino performs the “linear interpolation” to calculate the proper extension percentage (see Appendix).

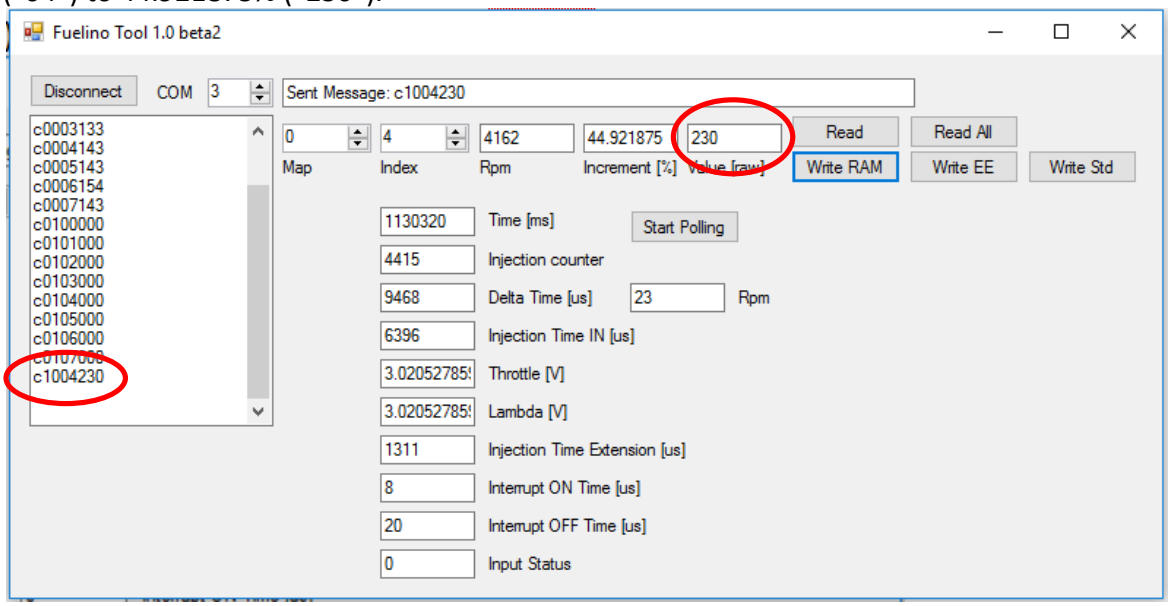


How to calibrate the maps

In order to set a desired increment percentage, you select a Map, and the Index that you want to modify. Then, write the Increment [%]. Please notice that the increment should be within “0” and “49.9” [%]. Then, click on *“Write RAM”* button.

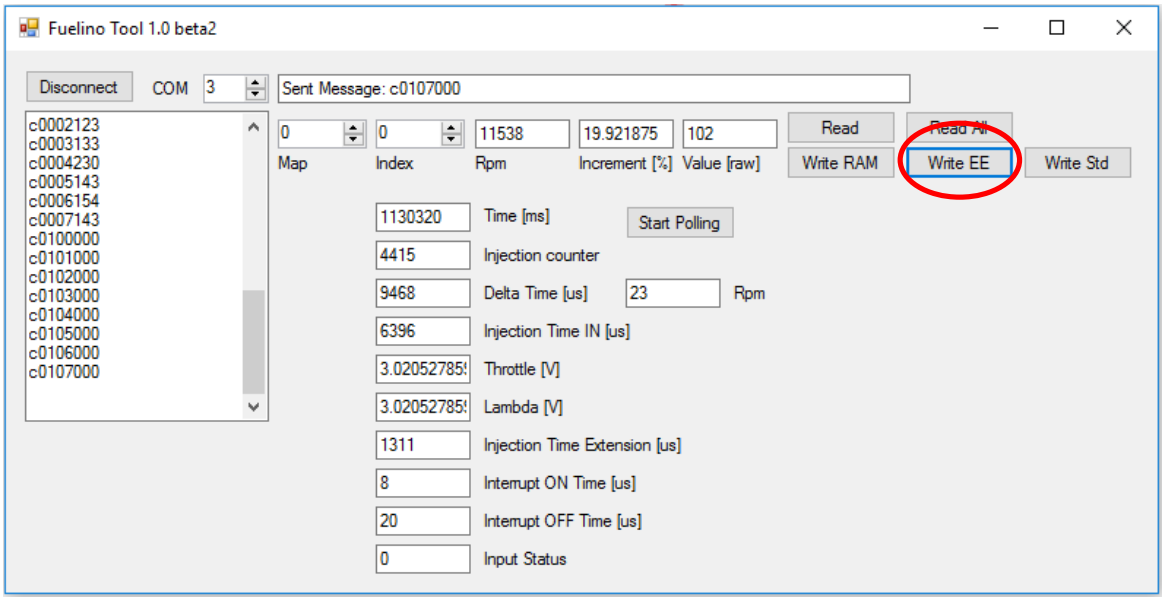


Fuelino uses <u>1 byte</u> to encode each increment value. Therefore, the value can be from 0 to 255. “256” would correspond to an increment of 50%. Since, in the example above, I wrote “45” %, the tool calculated a raw value of “230”, which corresponds to a practical increment of “44.921875” %. This is the value which is actually used by Fuelino. On the left side, Fuelino replied *“c1004230”*. This raw message means that Fuelino properly set the calibration (“c”). Fuelino set (“1”) the Map 0 (“0”) Index 4 (“04”) to 44.921875% (“230”).



How to save the calibrated values into EEPROM

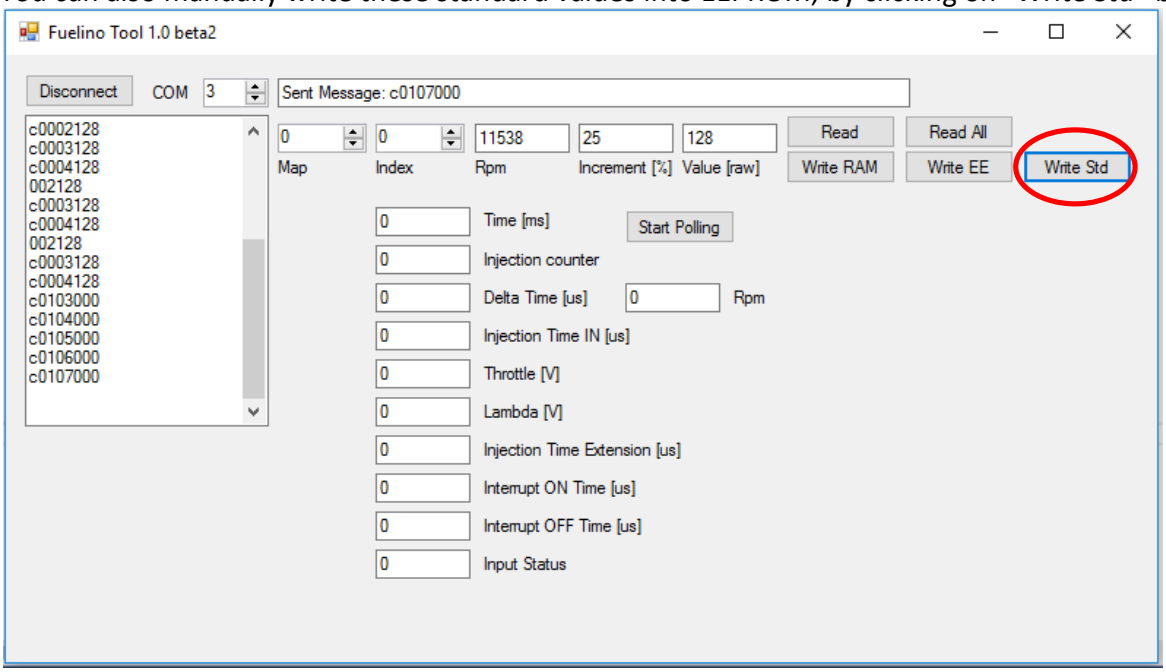
Please notice that when clicking to “Write RAM”, the calibration value is at first stored into RAM memory (volatile memory) and not in EEPROM (permanent memory). The reason why it is not suddenly written into permanent memory (EEPROM) is that, EEPROM memory cannot be written infinite times. After writing it some thousands of times, the memory corrupts. For this reason, the maps are at first saved into RAM memory. This allows you to test new calibrations at the fly (in run time). But in case you turn OFF the Fuelino (Power Off), all calibration will be lost. Therefore, if you want to keep your calibration values even after Power Off, make sure that you click *“Write EE”* after having calibrated all your maps.
In other words, you should at first save your calibration on the RAM memory *(“Write RAM”)* as explained on the previous chapter. After performing all changes, you should click on *“Write EE”*. Basically, when running this command, Fuelino copies the content of the RAM calibrations and saves it into the EEPROM memory, so that the same calibration maps will be available also at the next Power On.



Calibration maps: standard values

Fuelino calibration maps are saved into EEPROM memory. In order to be safe against data corruption, Fuelino is using “redundancy”, which means that the same maps are saved 2 times. At each Power On, Fuelino loads calibration maps from the EEPROM, and performs a checksum check to determine if the data is reliable. In case of no reliability (checksum check fails), Fuelino writes the standard values in the EEPROM. These values are defined as “*INJ_INCREMENT_RPM_STD*” and “*INJ_INCREMENT_THR_STD*” inside the C++ source code file *“INJmgr.h”*.

You can also manually write these standard values into EEPROM, by clicking on *“Write Std”* button.



In latest SW, the standard values are as following:

```
#define INJ_INCREMENT_RPM_STD (uint8_t)128 // Injection increment standard (50/256 %)
#define INJ_INCREMENT_THR_STD (uint8_t)0 // Injection increment standard (50/256 %)
```

Appendix 1: Example of Map 0 calibration (*incrementi_rpm[]*)

The following picture shows an example of calibrated Map 0 (increment % depending on engine speed rpm), in which I calibrated the 8 indexes with the following values [%]:

0. 20
1. 22
2. 24
3. 26
4. 28
5. 28
6. 30
7. 28

The theoretical increments are shown as red points in the image below. The blue circles are the measured points. They have been measured using a validation tool (*Pico DrDaq*). As shown in the picture below, Fuelino uses interpolation to calculate the injection % to be used, between 2 breakpoints (blue line between 2 red points).