

# Installation and configuration manual

---

*05-05-2018*

*Version 1.4*

*Company Information*

*Copyright and third-party information as required*

## Document Revisions

Date	Version Number	Document Changes
29/01/2017	1.0	Initial Draft
1/02/2017	1.1	Added safety and precautions
08/02/2017	1.2	Added arduino guide, usb driver, fw upgrade, reset to factory
11/03/2017	1.3	Review and corrections
05/05/2018	1.4	NEW version for MQTT remote control architecture

## Table of Contents

Safety & Precaution.....	5
1 SmartPID SPC1000 overview.....	6
1.1 Scope and Purpose.....	6
1.2 SmartPID technical characteristics.....	6
1.3 SmartPID HW description.....	9
2 SmartPID integration.....	12
3 SmartPID remote control.....	15
3.1 WiFi configuration.....	17
4 Arduino configuration.....	18
4.1 Windows USB driver installation – virtual com.....	18
4.2 MAC and Linux USB driver.....	21
4.3 Setting up Arduino IDE.....	21
5 Compiling and uploading sketch.....	25
6 Firmware upgrade.....	26
7 Factory reset.....	27

## Safety & Precaution

Ensure that the product is always used within the specifications

Do not use product close to flammable and explosive gas otherwise injury from explosion may occur

Never disassemble, modify, or touch any of the internal part to avoid electric shock or malfunctions

Do not use the relay over their life cycle and do not exceed the rated load of the outputs

Do not touch the terminals at least while power is being supplied. Doing so may occasionally result in injury due to electric shock.

Do not allow pieces of metal, wire clippings, or fine metallic shaving or filings from installation to enter the product.

Do not allow water or any liquid enter the product. Enclosure is not water proof

The board is sold as a DIY standalone component and people buying should take care of connecting and integrating with their own system. The manula connection diagram and short explanations but minimum expertise in electric circuit is needed.

The board is powered by **High Voltage 220/110V** so you must be very careful and all connections are at your own risk. If you are not familiar with electricity and power please ask a technician to help you. I'm not responsible for any damage or risk you can create



## 1 SmartPID SPC1000 overview

### 1.1 Scope and Purpose

The purpose of this document is to describe in detail the installation and configuration process of the SPC1000 smartPID controller. For the user application feature and functions please refer to proper manuals.

### 1.2 SmartPID technical characteristics

The below table summarize the technical characteristic of SmartPID

FEATURE	SmartPID SPC1000
Control channel	2
Control mode	Heating/Cooling/Thermostatic
PID alghoritm	Y
ON/OFF algorithm	Y
PID autotune	Y
NTC sensor	2
Beta coefficient configuration (NTC)	Y
DS18B20 digital sensor	2
Sensor calibration	Y
Accuracy	0.125C (DS18B20)
Resolution	0.1C
Temperature range	-55C+125C (DS18B20)
250VAC/10A Relay Output	2
SSR Output (PWM)	1
12V 2A direct drive output	2
220v/110v AC power	Y
Box measurement	DIN 36x72x75
OLED graphical display	1,3"
USB port	Y

I2C Expansion port	Grove I2C port (3v3)
PC/Web configuration	Y
Data logging (USB / WiFi)	8Mb
WiFi Connection	Y
Remote management/configuration	Y
Automation process control	Y
Multiple application support	Y
Arduino compatible	Y

### WIFI Module characteristics

IEEE 802.11 b/g/n Wi-Fi

WEP or WPA/WPA2 authentication, or open networks 802.11 b/g/n

Wi-Fi Direct (P2P), soft-AP

Integrated TCP/IP protocol stack

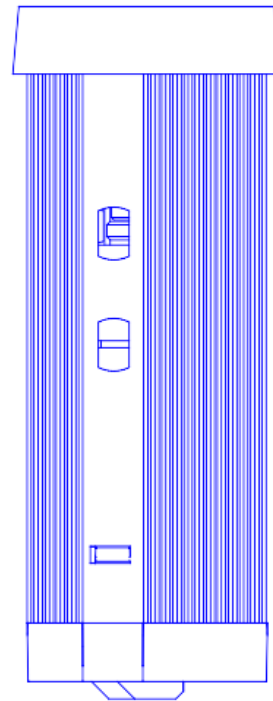
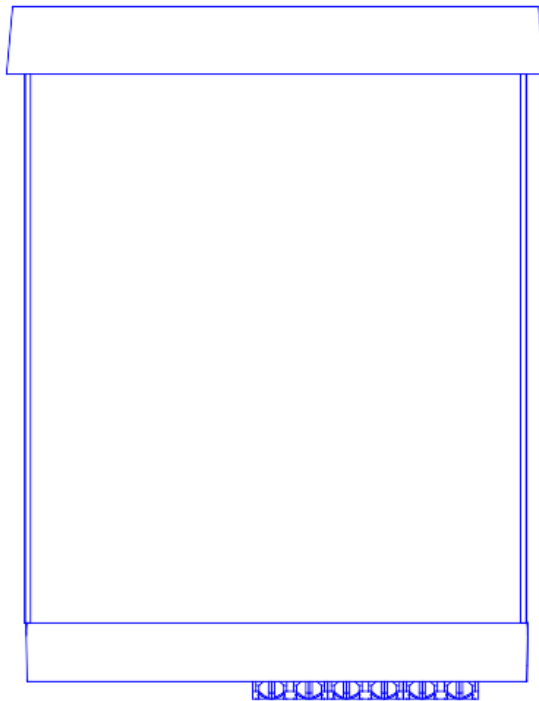
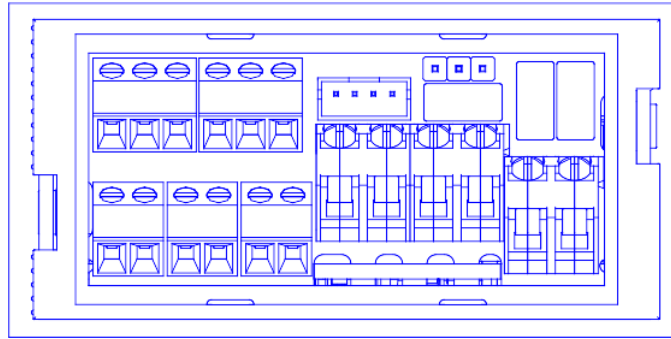
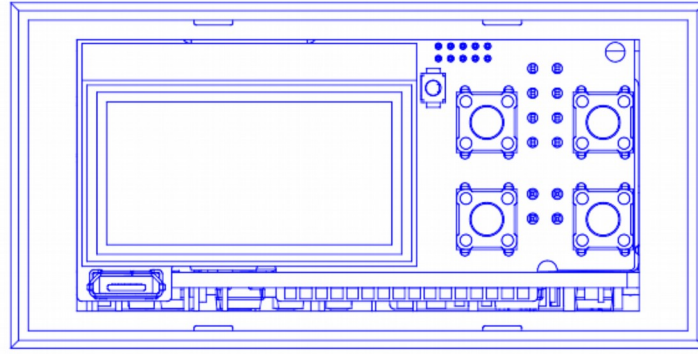
Integrated TR switch, balun, LNA, power amplifier and matching network

+19.5dBm output power in 802.11b mode

Power down leakage current of <10uA

Standby power consumption of < 1.0mW (DTIM3)

### Mechanical Drawings

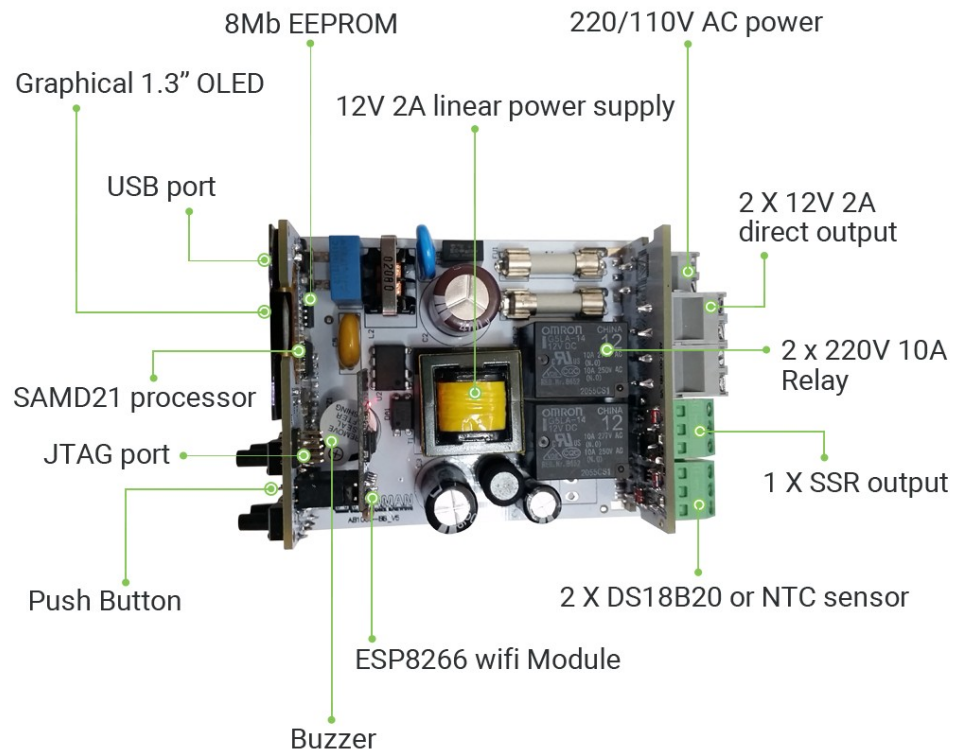
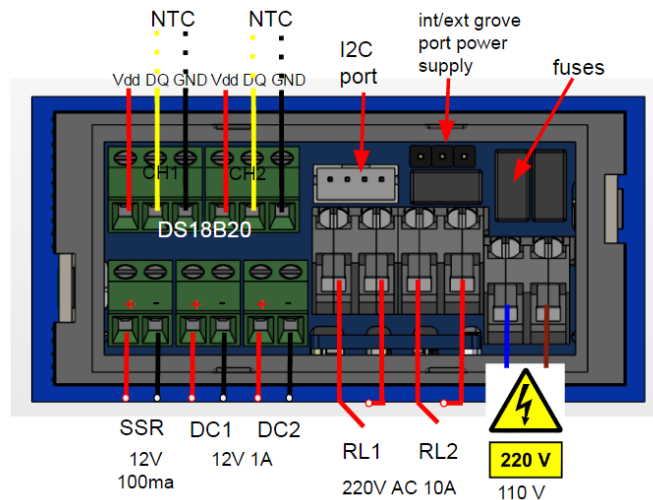




### 1.3 SmartPID HW description

SmartPID is constituted by 3 main boards assembled together and inserted in a DIN small box

On the front panel you have the 1.3" OLED graphical display, the push buttons for interactions and the USB port for data logging/FW upgrade. On the back panel are located the terminal blocks for the temperature probe sensors, the relay outputs, the two 12V outputs, the SSR output, the I2C expansion port as well the 220/110V AC power input.



The schema report the connections of all terminal blocks and connectors

the 3 ports input connectors can be used either for one wire bus with DS18B20 temperature sensor (no parasitic resistor is needed) or for 10K NTC sensors.

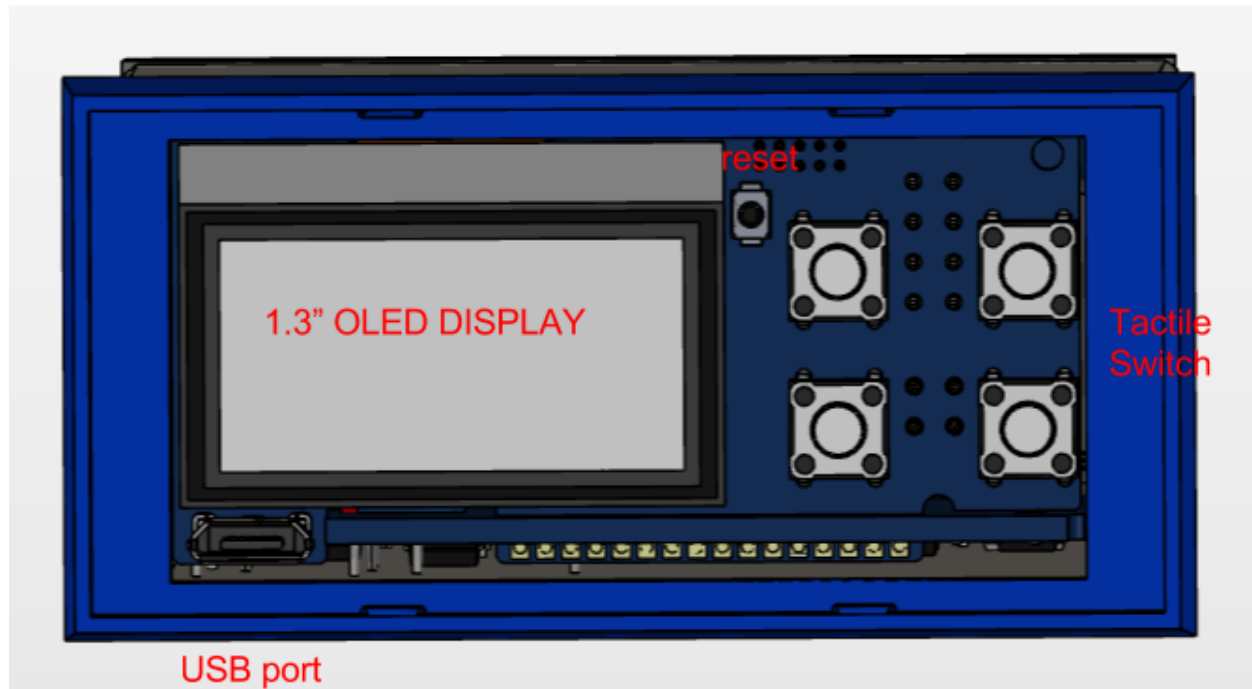
The I2C port has a pinout compatible with **grove** bus with 3V3 power supply configurable (int or ext) via jumper

The Grove I2C connector has the standard layout. Pin 1 is the SCL signal and Pin 2 is the SDA signal. Power and Ground are the same as the other connectors.

pin1	SCL	I2C Clock
pin2	SDA	I2C Data
pin3	VCC	Power for Grove Module, 3.3V
pin4	GND	Ground

There are many types of I2C Grove sensors available. Most are 5V/3.3V devices, but there are a few that are only 3.3V or 5.0V. You need to check the specifications.

for further details see [http://wiki.seeed.cc/Grove\\_System/](http://wiki.seeed.cc/Grove_System/)



The front board 4 tactile switch that are used for multiple functions

	<ul style="list-style-type: none"> <li>a) UP/DOWN <ul style="list-style-type: none"> <li>a. Scroll in configuration menu</li> <li>b. Increase decrease temperature value</li> <li>c. Scroll in value in configuration menu</li> </ul> </li> <li>a) SET <ul style="list-style-type: none"> <li>d. Select/enter a specific menu</li> <li>e. Select/confirm a specific value</li> <li>f. Confirm action upon prompt request</li> </ul> </li> <li>b) Start/Stop <ul style="list-style-type: none"> <li>g. Start process</li> <li>h. Stop Process</li> <li>i. long press exit from the current menu</li> </ul> </li> </ul>
--	---

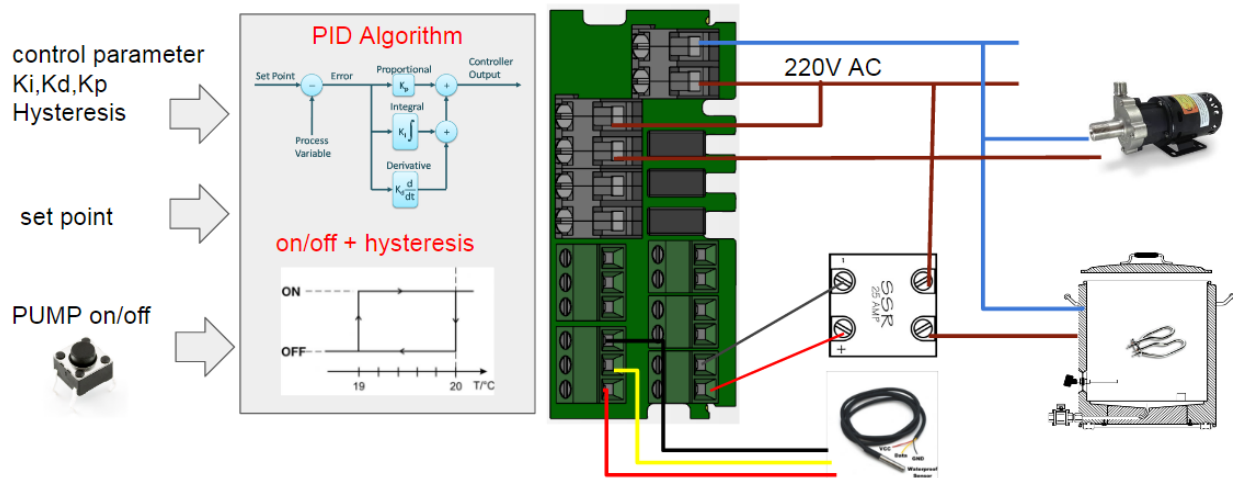
## 2 SmartPID integration

Via HW set up configuration SmartPID can be adapted to manage different applications, the HW configuration allows to assign different processes/logicalchannels to different physical HW resources.

Below some example to illustrate the flexibility and multiple configurations possible (more details in the application manuals)

### Single channel – PID (on/off) control mode – heating + pump

SmartPID reads temperature from 1 probe and drive the heating element via SSR out



Possible heating output configurations

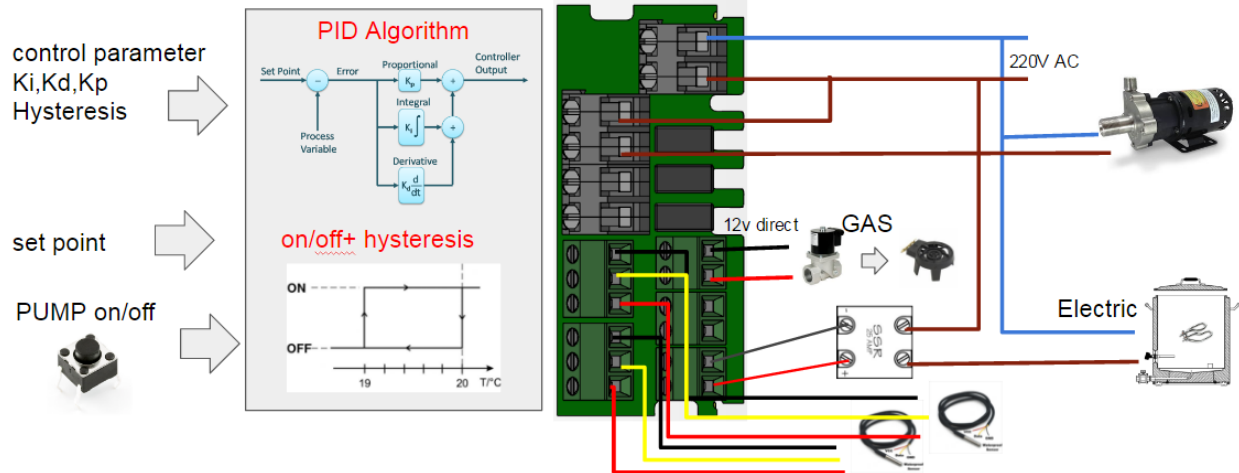
- SSR for electric heating [PID+PWM control algorithm]
- 220V AC relay for electric heating [ON/OFF algorithm]
- Direct solenoid valve drive for GAS heating [ON/OFF algorithm]

Manually drive the PUMP (soft switch) with tow possible configurations

- 220V AC realy pump drive
- 12V DC direct pump drive

### DUAL channel – PID (on/off) control mode – heating (gas or electric)

SmartPID reads temperature from 2 probe and drive 2 heating element interdependently in order to manage two different heating process



Possible heating output configurations

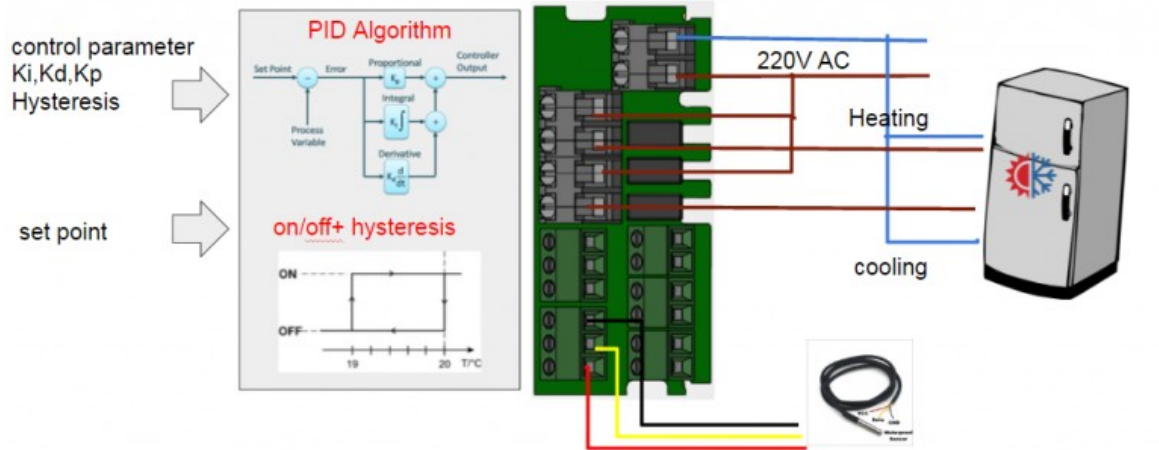
- 1 x SSR for electric heating [PID+PWM control algorithm]
- 1 x 12V DC2 direct solenoid valve drive for GAS heating [ON/OFF algorithm]

Manually drive the PUMP (soft switch) with tow possible configurations

- 220V AC realy pump drive
- 12V DC direct pump drive

#### DUAL channel – ON/OFF control mode – thermostatic

SmartPID reads temperature from 1 probe and drive one heating element and one cooling element in order to keep temperature stable (fermentation chamber)

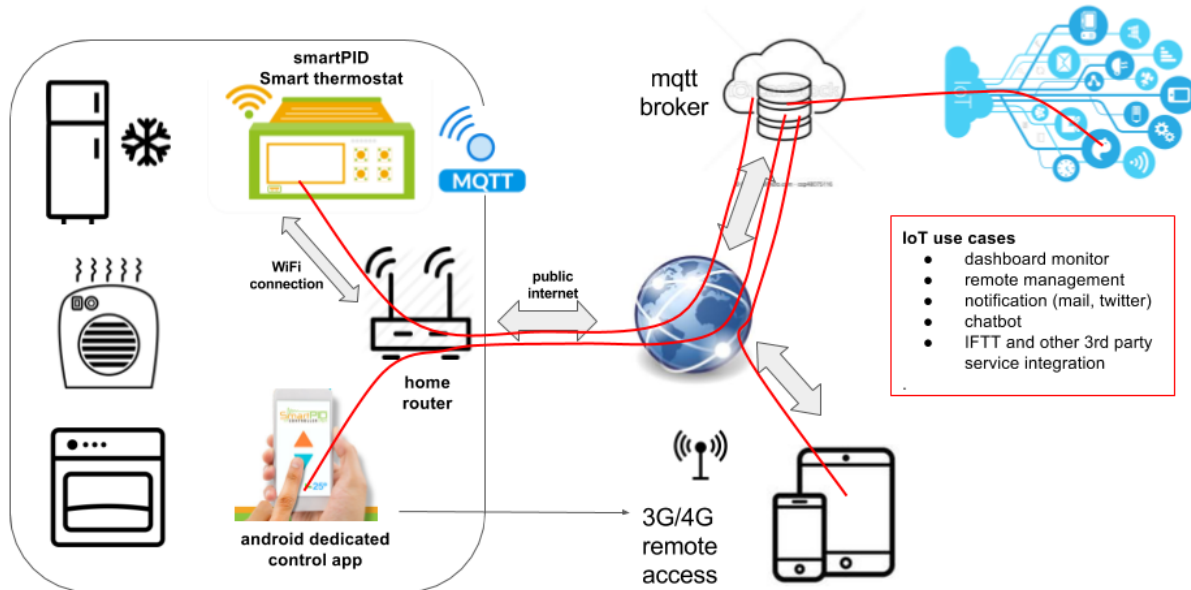


#### Possible heating output configurations

- 220V AC relay for electric heating [ON/OFF algorithm]
- 220V AC relay for electric heating [ON/OFF control algorithm]

### 3 SmartPID remote control

SmartPID controller has on board a WiFi module to connect to home WiFi router. Once connected to the public internet smartPID communicates to a server using standard MQTT protocol and different applications can interact with the controller via server



this type of architecture allow multiple functions and remote mangement

- Process Data Logging and monitor

Main process parameter during status mode or run mode (standard and advanced) are pushed with a configurable frequency to the server and the remote smartphone APP subscribe that data do display on a graphic dashboard

- Set Point temperature
- Current Temperature
- PWM percentage
- Timers

- Events remote notifications

Main process events and alarms are asynchronously notified to the remote application

- process start/stop
- set point reached

- timer expiration
- Remote commands

Via remote APP it's possible to interact with the controller and modify relevant parameters

- set point
  - count down timer
  - start/stop processes
  - pause/resume
- Remote ramp/soak profile definition

Profile definition can be performed using the graphical interface on the smartphone app and then saved on the smartPID controller for execution



### 3.1 WiFi configuration

In order to connect SmartPID controller to home WiFi network and to remote server and to the smartphone app 3 basic operation are need

1. SmartPID account creation (sign-up via app)
2. WiFi configuration and provisioning
3. SmartPID pairing with the app using serial number

Following data are need to complete the 3 steps

**Make sure to write correctly and avoid typo errors (pay attention to capital letters, unwanted space, O and 0 etc..)**

WiFi home network

- SSID (WiFi network name)
- Password

SmartPID account

- Login
- Password

SmartPID serial number

- 14 char unique string [visible on boot screen and INFO menu]

**For a detailed step by step procedure please refer to the specific guide**

## 4 Arduino configuration

SmartPID platform is 100% compatible with the arduino ecosystem since SmartPID processor is the SAMD21G (arm cortex M0+) used in arduino zero and MKR1000 platform.

In order to program your SmartPID using the Arduino IDE it's necessary to install some components and drivers.

### 4.1 Windows USB driver installation – virtual com

In order to install the USB driver and create a virtual serial com you need to follow below steps

After plugging the board in, Windows will try – and fail – to search the Internet for drivers. Click the button below link to download the driver.

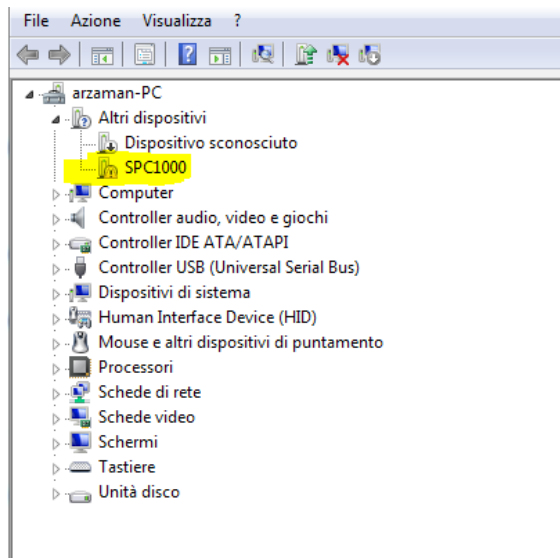
#### DOWNLOAD THE SAMD21 WINDOWS DRIVERS

1.After downloading copy down the location of the araman.inf files.

2.Open your Device Manager

- In Windows 8 or 10, simply search for "Device Manager" and select the Windows app
- See Window's resources for Windows 7.
- (Start > Run > "devmgmt.msc", should work on almost any version of the OS.)

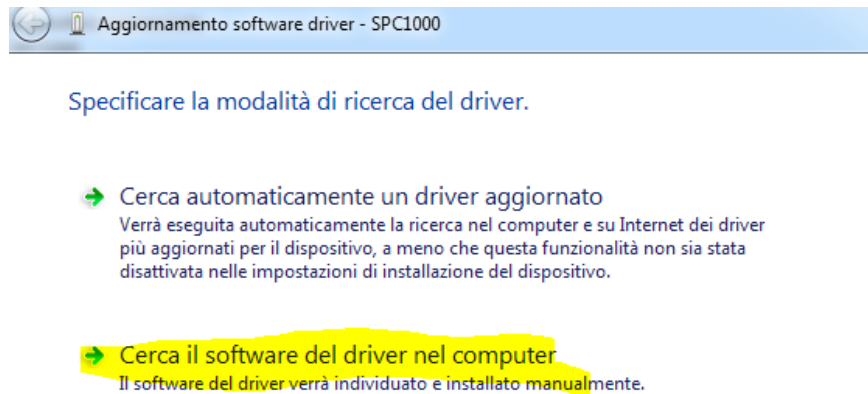
3.In the Device Manager, expand the "**Other devices**" tree-- you should see an entry for



*Screenshot: Navigating the Device Manager*

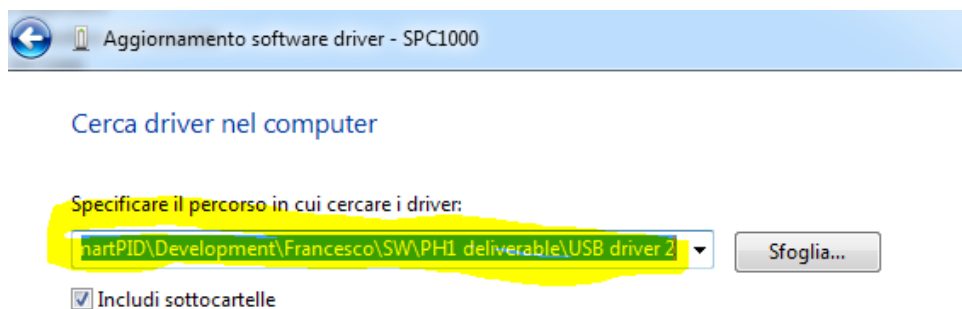
4. "Unknown Device", **Right-click on SPC1000 and select Update Driver Software...**

5. Select **Browse my computer for driver software**. On the next screen



*Screenshot: Browse your computer for the driver*

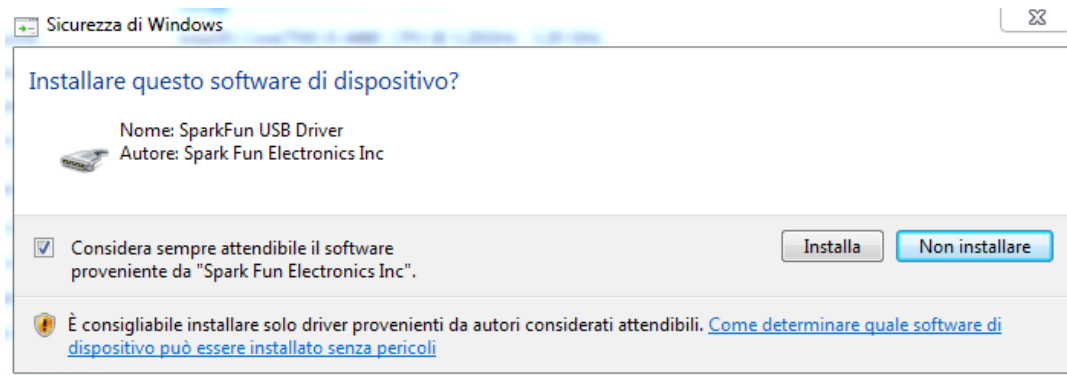
6. **Paste the directory location** of your arzaman.inf and files into the search location. Then hit "Next".



*Screenshot: Setting the driver location*

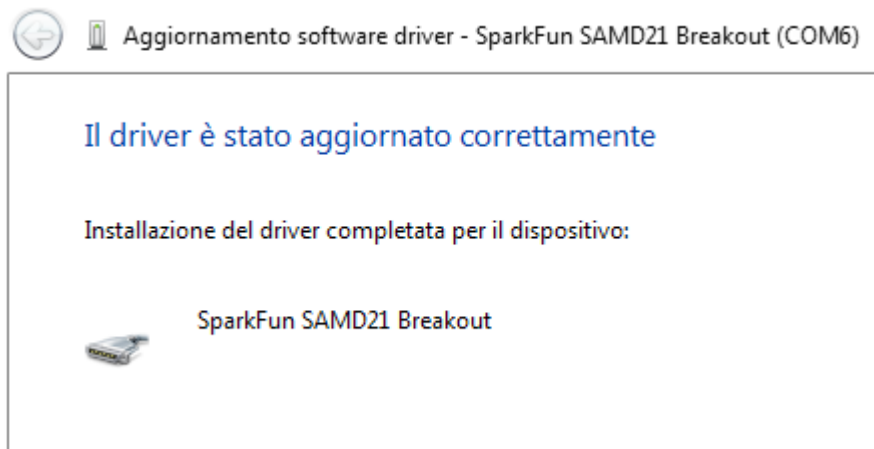
7. Click **"Install"** when the next pop-up questions if you want to install the driver.

8. If you get any security warning please proceed with installation

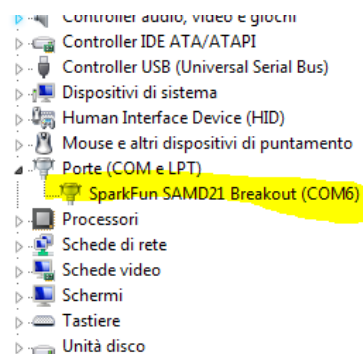


*Screenshot: security warning*

9. The driver installation may take a moment, when it's done you should be greeted with a "successfully updated your driver software" message!



10. Once the installation is completed you should see the serial com created



## 4.2 MAC and Liux USB driver

Mac and Linux users shouldn't need to download any drivers. The device should show up as a serial port as soon as it's plugged in to your computer.

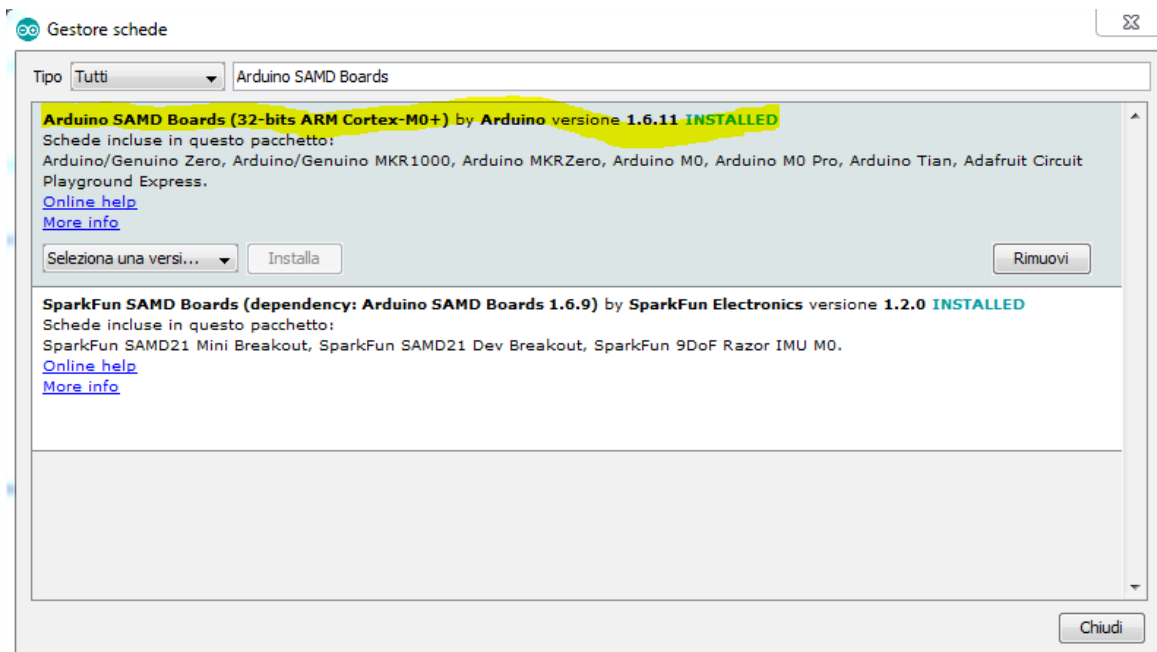
## 4.3 Setting up Arduino IDE

SAMD21 MCU is fully supported by Arduino IDE you just need to add ARM Cortex-M0+-support to your Arduino IDE. This page will list every step required for getting SmartPID SAMD21 controller support into your Arduino IDE.

**Update Arduino!** This setup requires at least Arduino version 1.6.4 or later. We've tested it on IDE 1.6.13 . If you're running an older version of Arduino, consider visiting [arduino.cc](http://arduino.cc) to get the latest, greatest release.

First, you'll need to install a variety of tools, including low-level ARM Cortex libraries full of generic code, arm-gcc to compile your code, and bossa to upload over the bootloader. These tools come packaged along with Arduino's SAMD board definitions for the Arduino Zero.

To install the Arduino SAMD board definitions, navigate to your board manager (Tools > Board > Boards Manager...), then find an entry for Arduino SAMD Boards (32-bits ARM Cortex-M0+). Select it, and install the latest version (recently updated to 1.6.13)

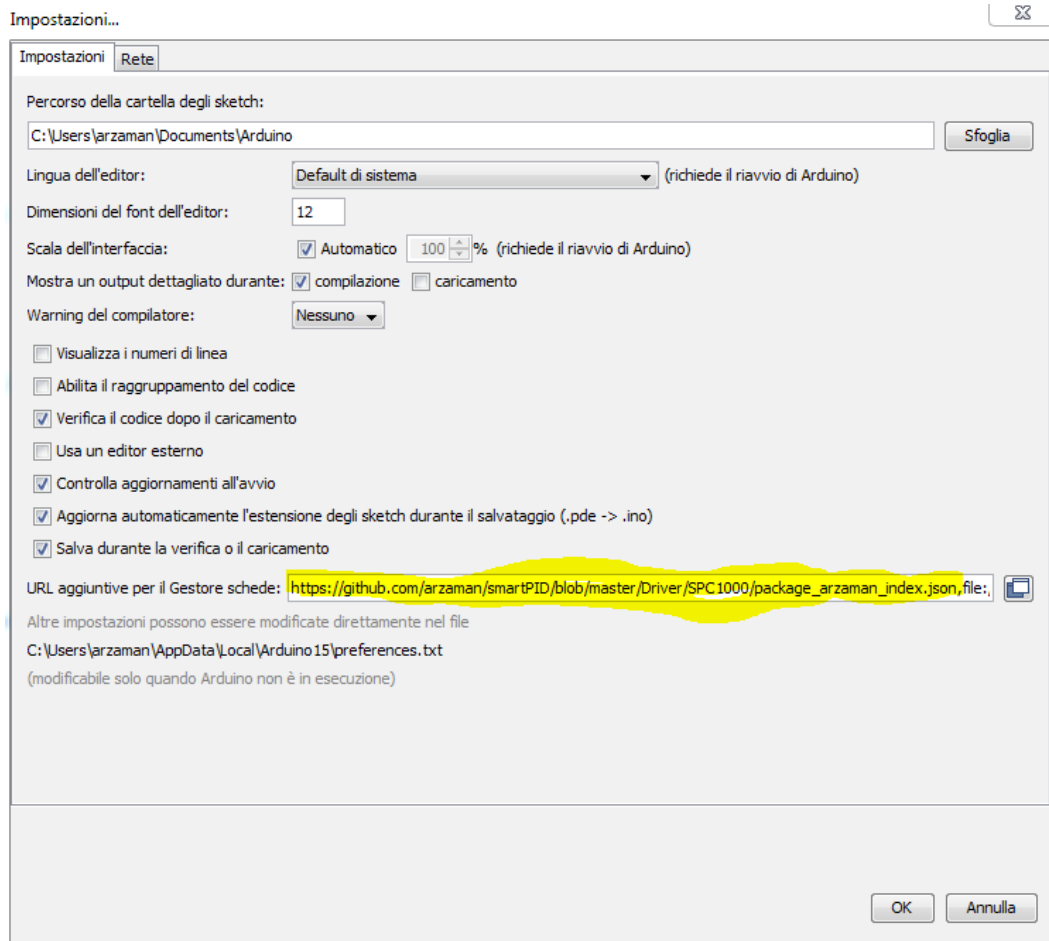


Downloading and installing the tools may take a couple minutes – arm-gcc in particular will take the longest, it's about 250MB unpacked.

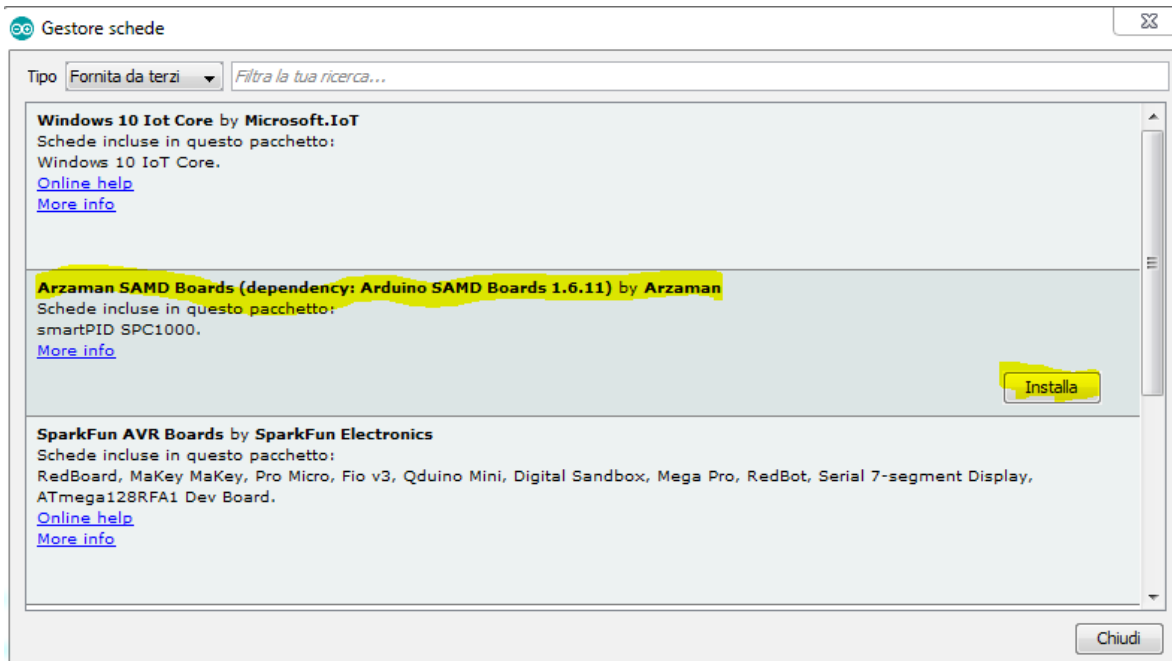
Once installed, Arduino-blue "Installed" text should appear next to the SAMD boards list entry.

Now that your ARM tools are installed, one last bit of setup is required to add support for the SmartPID board definition. First, open your Arduino preferences (File > Preferences). Then find the Additional Board Manager URLs text box, and paste the below link in:

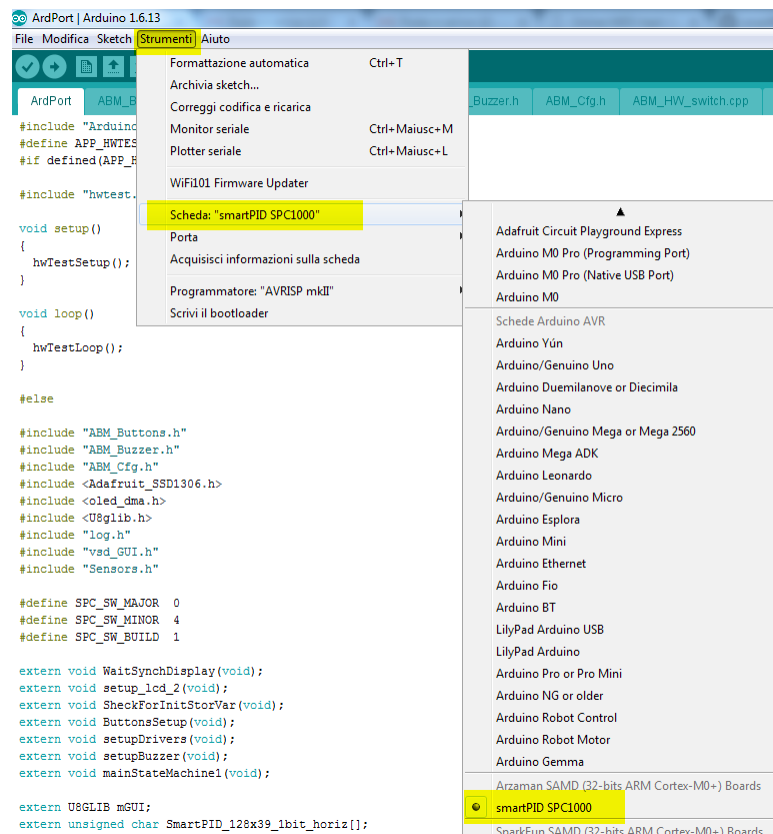
[https://raw.githubusercontent.com/arzaman/smartPID/master/Driver/SPC1000/package\\_arzaman\\_index.json](https://raw.githubusercontent.com/arzaman/smartPID/master/Driver/SPC1000/package_arzaman_index.json)



Then hit “OK”, and travel back to the Board Manager menu. You should (but probably won’t) be able to find a new entry for Arzaman SAMD Boards. If you don’t see it, close the board manager and open it again.



Once you have installed the SmartPID – SPC100 board definition you should see two one entries in your Tools > Board list: Select your **SmartPID - SPC1000** Board.





## 5 Compiling and uploading sketch

Once you have install the SAMD core components and the SmartPID – SPC1000 board definition you are ready to develop and compile your sketch as any ordinary arduino code.

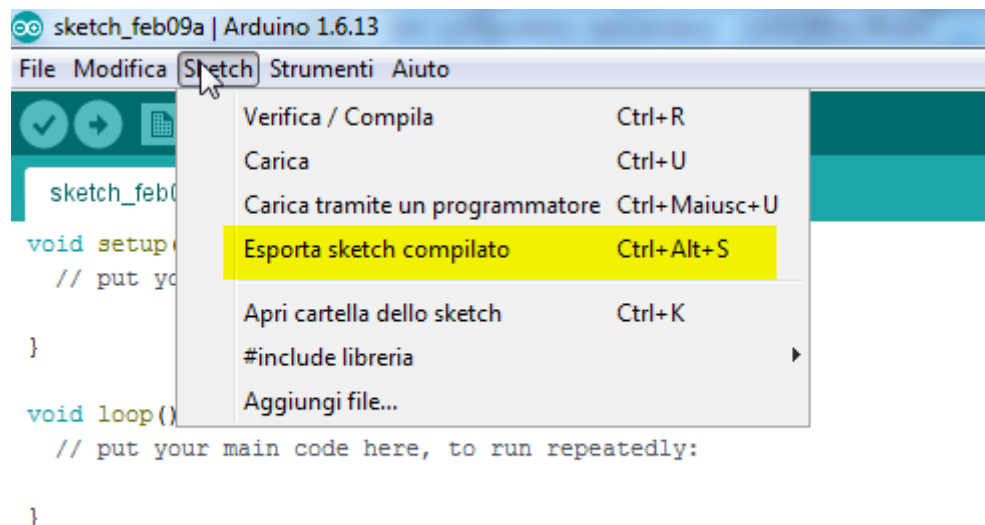
In order to support some basic functions you need to install locally the library that are available on github

<https://github.com/arzaman/smartPID/tree/master/Libraries>

To upload the binary file a special boot-loader has been developed based on **Mass Storage device** (MSD) technology. The bootloader replace the SAM-BA original atmel bootloader

The binary file is NOT loaded via the serial interface using BOSSA tool insiude the IDE but is simply copy pasted in the smartPID board that is mapped as a USB memory drive [see chapter FW upgrade]

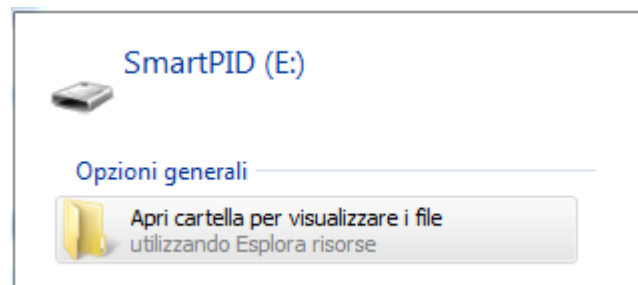
Once you have compiled successfully the sketch you can export the binary file in your sketch folder



Than simply copy-paste the <sketch name>.bin file in the USB memory drive and than **safely remove** the drive and reboot the controller

## 6 Firmware upgrade

The firmware upgrade can be performed very easily using the **MSD mass storage device** bootloader technology that allow you to “copy paste” the binary executable file in the smartPID controller connected via USB and mapped as a USB memory



**To upgrade FW you DON'T need any driver or SW installation, just simply follow below steps**

Programming an application via the bootloader

1. power up the board while keeping the S/S button pressed: the OLED display will be lit with a solid white color
2. connect the board to a USB port, it will appear as a mass storage device and the OLED display will alternate between a black and a white screen
3. remove the FLASH.BIN file present in the mass storage device
4. copy the your executable .bin file of your application to the mass storage device
5. **safely remove the USB** mass storage device from the PC
6. power-cycle the board and reboot

## 7 Factory reset

If you need restoring application settings to factory default settings (internal EEPROM erase) you perform via the bootloader following operation

**power up** the board while keeping the **DOWN button** pressed: the OLED display will report the "Init EEPROM" message indicating that application settings have been reset to factory defaults; after a few seconds, the application will start with default settings