Installation and configuration manual

11-03-2017

Version 1.3

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

Table of Contents

Safety & Precaution 5
1SmartPID SPC1000 overview 6
1.1Scope and Purpose *6*
1.2SmartPID technical characteristics *6*
1.3SmartPID HW description *8*
2SmartPID integration 11
3 SmartPID wifi configuration 13
3.1Thingspeak server configuration *14*
3.2WiFi Configuration *17*
SmartPID configuration 17
SmartPID connection to WiFi router 20
Process parameter logging 21
3.3WiFi configuration via browser *23*
4Arduino configuration 25
4.1Windows USB driver installation – virtual com *25*
4.2MAC and Liux USB driver *28*
4.3Setting up Arduino IDE *28*
5Compiling and uploading sketch 31
6Firmware upgrade 32
7Factory reset 33

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 prof

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

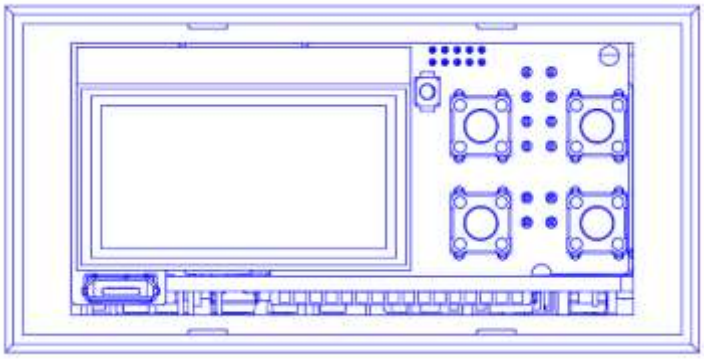
Integrated TR switch, balun, LNA, power amplifier and matching network

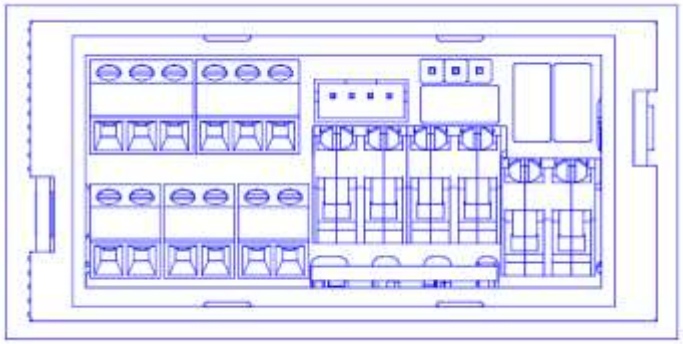
+19.5dBm output power in 802.11b mode

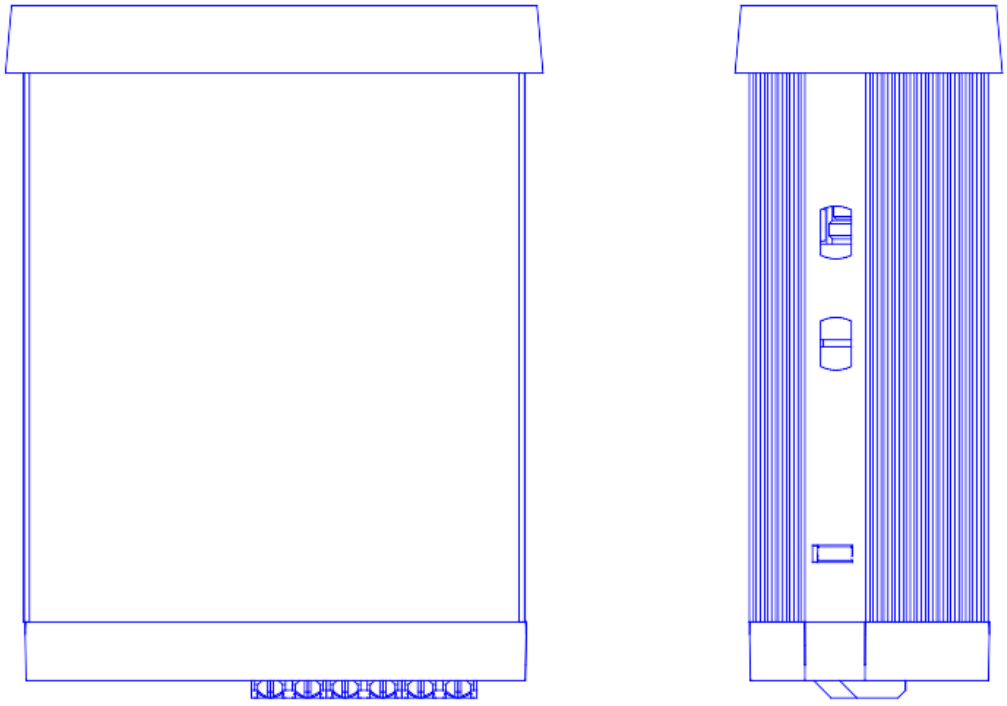
Power down leakage current of <10uA

Standby power consumption of < 1.0mW (DTIM3)

Mechanical Drawings



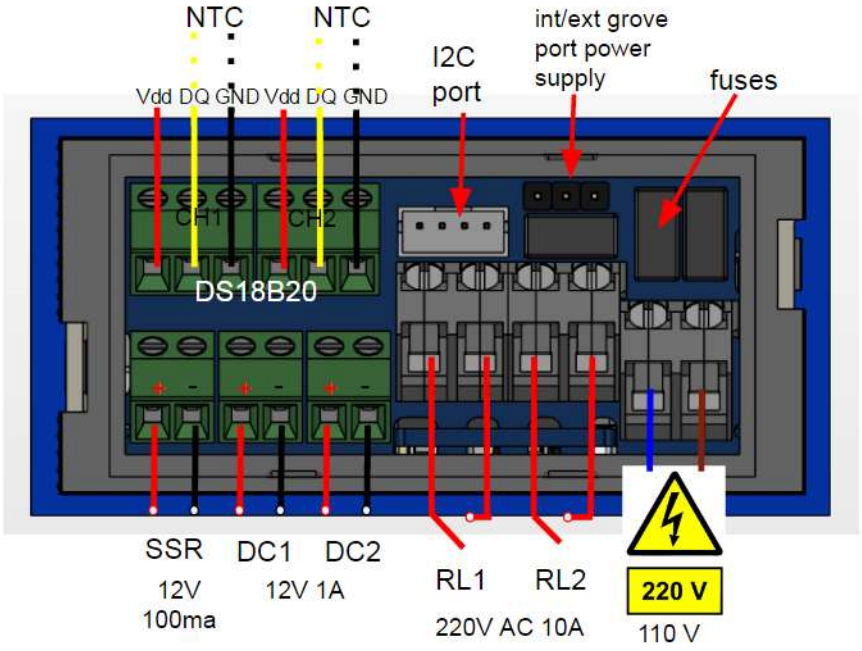


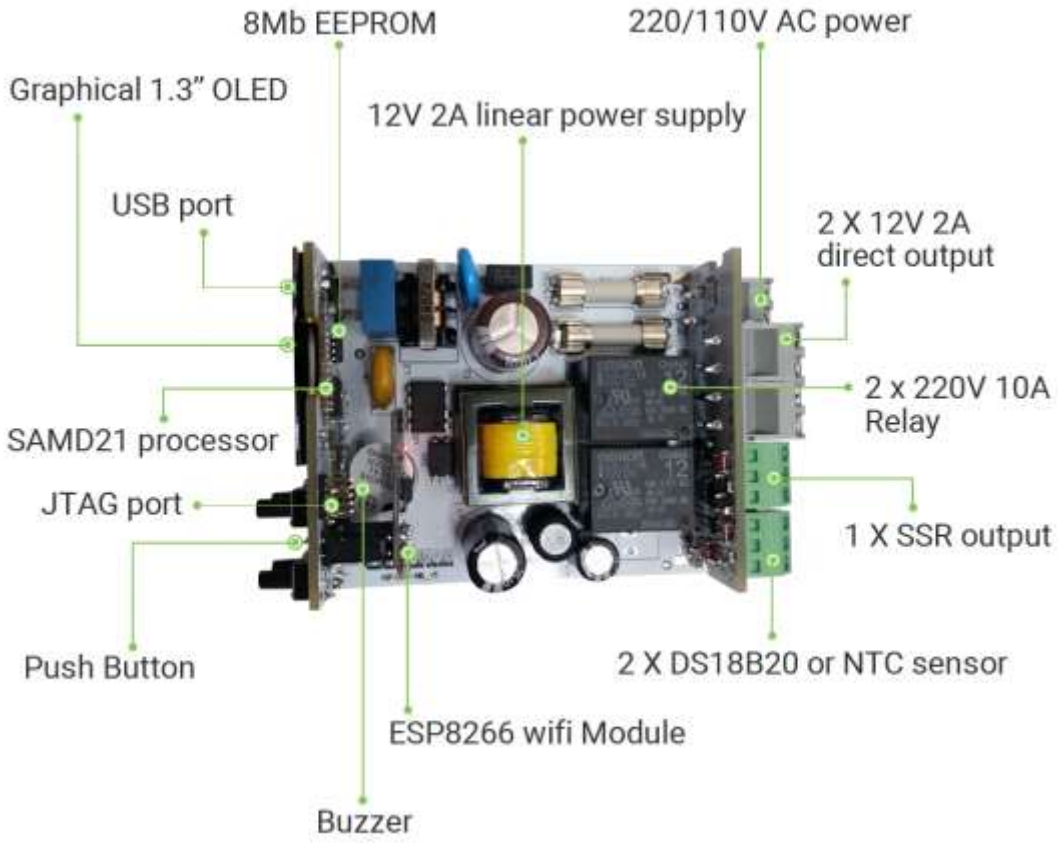


1.3 SmartPID HW description

SmartPID is constituted by 3 main boards assembled together and inserted in 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.

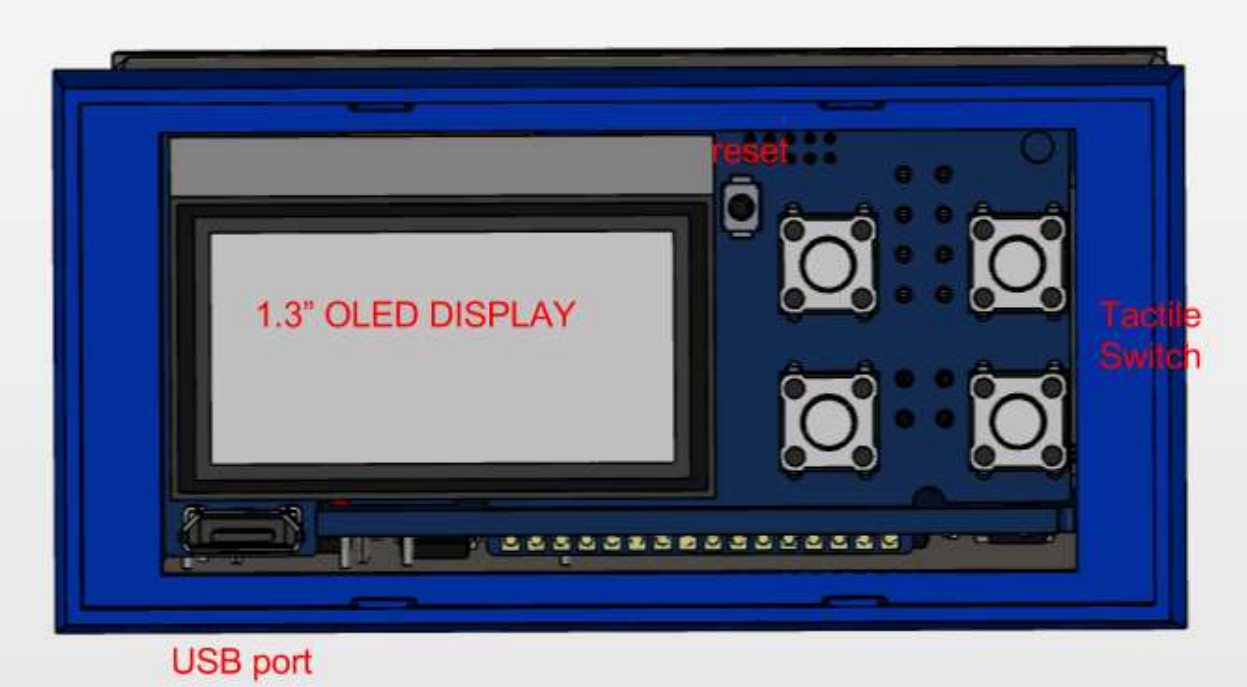
The I2C port has a pinout compatible with **grove** bus with 3V3 power supply configurable (int or ext) via jumper

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.

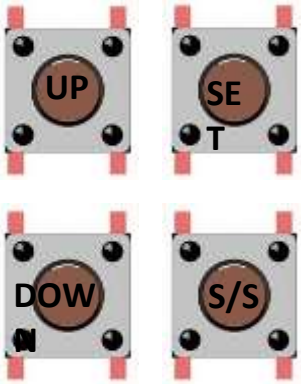
pin1	SCL	I2C Clock
pin2	SDA	I2C Data
pin3	VCC	Power for Grove Module, 3.3V
pin4	GND	Ground

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/



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

	a) UP/DOWN a. Scroll in configuration menu b. Increase decrease temperature value c. Scroll in value in configuration menu a) SET d. Select/enter a specific menu e. Select/confirm a specific value f. Confirm action upon prompt request b) Start/Stop g. Start process h. Stop Process i. long press exit from the current menu

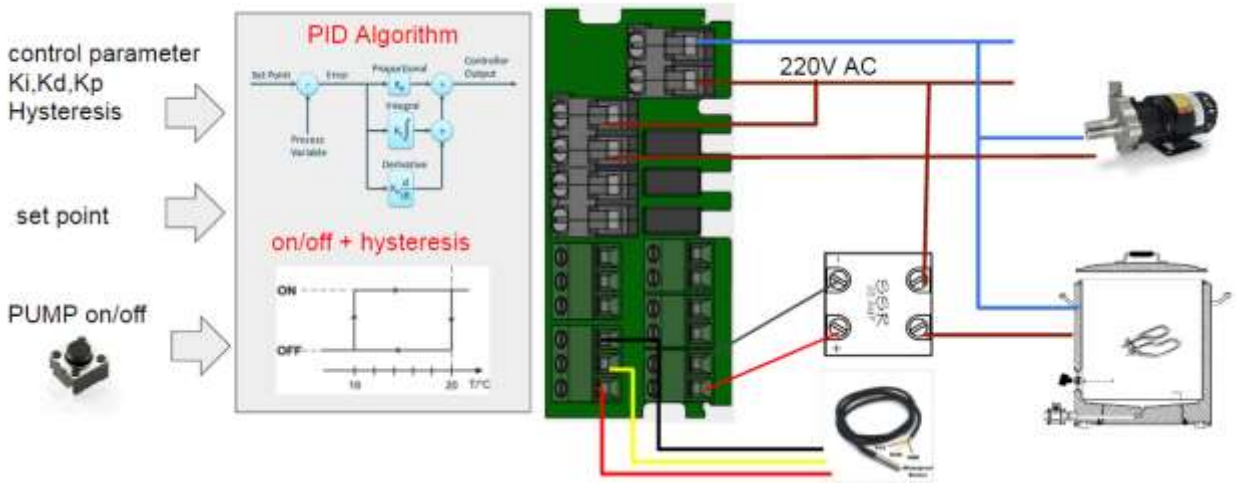
2 SmartPID integration

Via HW set up configuration SmartPID can be adapted to mange 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

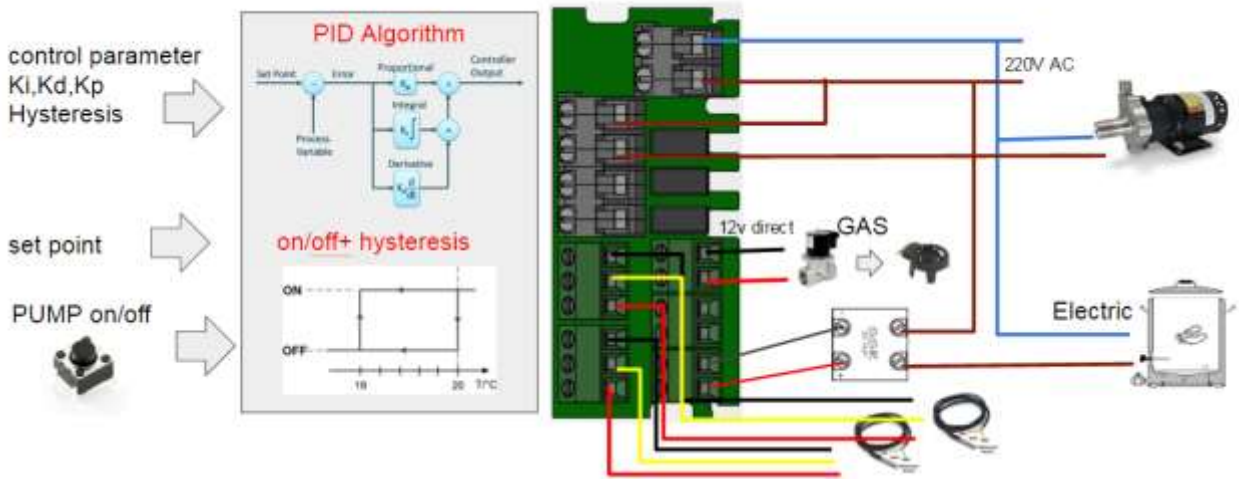
- SSR for electric heating [PID+PWM control algorithm]
- 220V AC relay for electric heating [ON/OFF algorithm]
- Direct solenoid valve drive for GAS heating [ON/OFF algorithm]

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 mange two different heating process



Possible heating output configurations

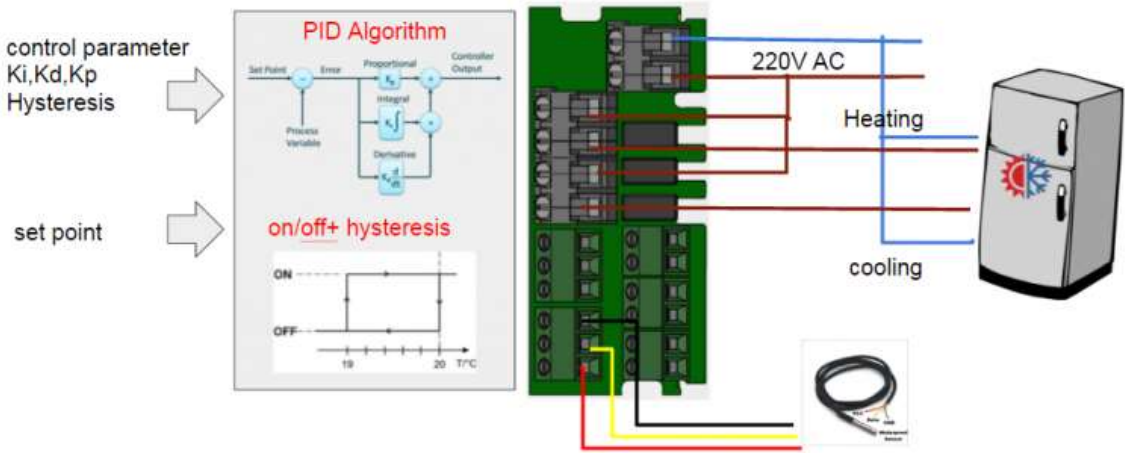
- 1 x SSR for electric heating [PID+PWM control algorithm]
- 1 x 12V DC2 direct solenoid valve drive for GAS heating [ON/OFF algorithm]

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

3 SmartPID wifi configuration

SmartPID has an internal wifi module that allow the application to communicate to an external server and push all log data and process parameters. The wifi module implement also a web server and via a smartphone dedicated app or a browser it's possible to :

1. perform the initial setup and configure once the relevant parameters
2. retrieve and visualize the log data from the cloud server
3. Modify from remote the set point [in run mode]

the SmartPID architecture for data logging is client-server and in order to manage and store data in

the cloud public **thingspeak** service is used

For more details on powerful thingspeak features refer to https://thingspeak.com

Each smartPID is associated to a thingspeak **channel** that is uniquely identified by a channel ID. Each thingspeak account can have multiple channels and so you can manage multiple smartPID from one thingspeak account. The channel is the logical entity that contains process data generated by smartPID and has a set of predefined field corresponding to process data

In order to complete the wifi configuration and thingspeak logical channel associations 4 mains logical step must be executed

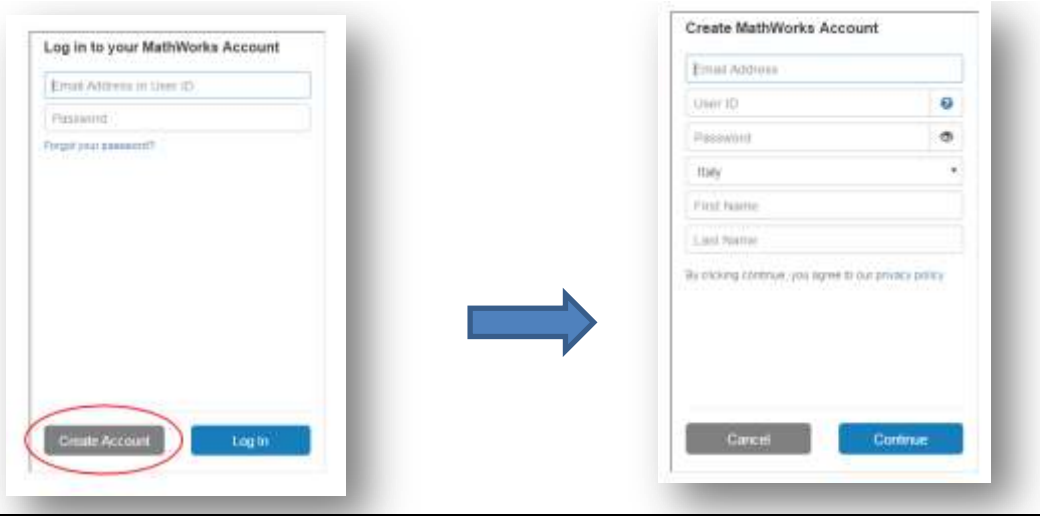
1. Create a thingspeak account and create the channel with proper structure
2. Create a smartPID account using the smartphone app and associate to the thinsgspeak account. A unique “user API KEY” is used to adress the proper thingspeak channel
3. Activate WiFi on smartPID and connect the smartphone (or laptop) with direct connection
4. Configure the smartPID via smartphone app (or web browser) updating following values
   a) Wifi SSID and password
   b) Channel ID to associate to the smartPID
   c) web server port / baud rate [can be left to default value]

3.1 Thingspeak server configuration

This operation is done on thingspeak server and is independent from smartPID configuration

1. Connect to https://thingspeak.com/
2. Select Sign Up:

3. Select Create account and fill the form

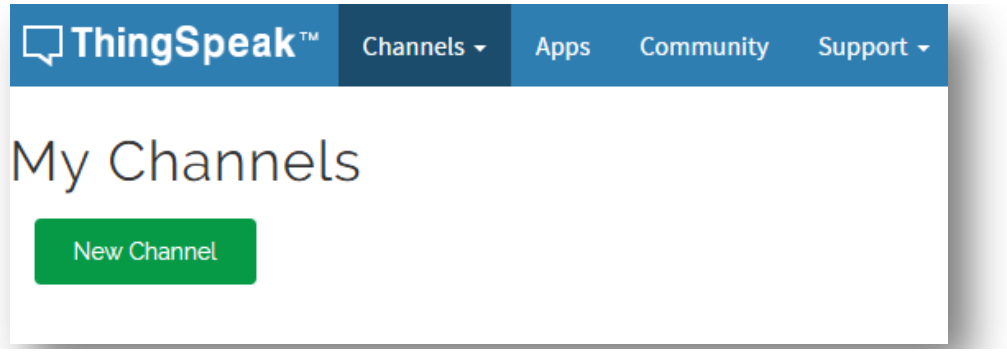


4. Verify Your account: go to your inbox and click the link in the received email

5. Press Continue and Sign in with your credentials and agree to terms.

Now you need to create a channel to feed the data from SmartPID, each channel can be public or private depending on your needs

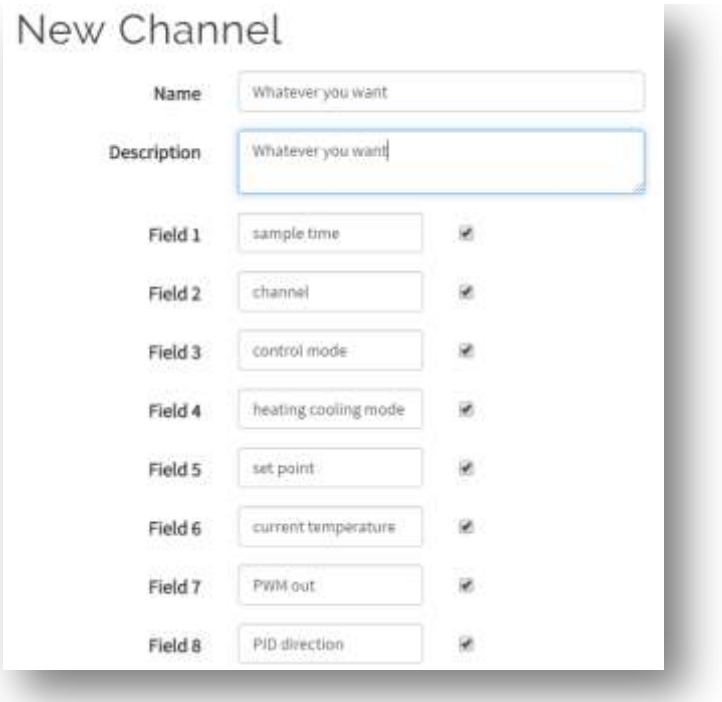
6. Press on New Channel:



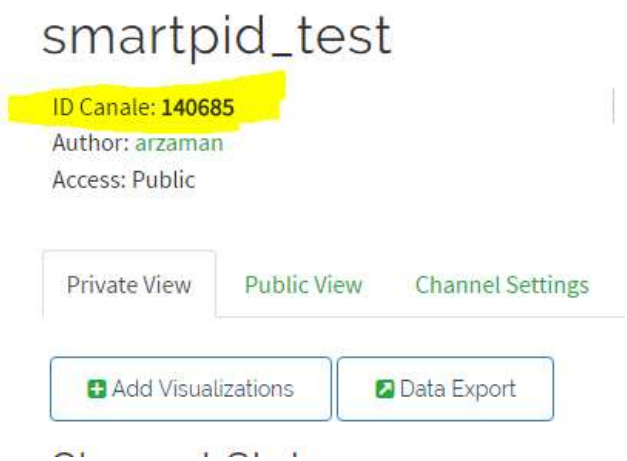
7. Fill the form for channel definition

Note that **Name** and **Description** can be filled as you prefer, the other fields **must be filled as suggested , same order and labels**

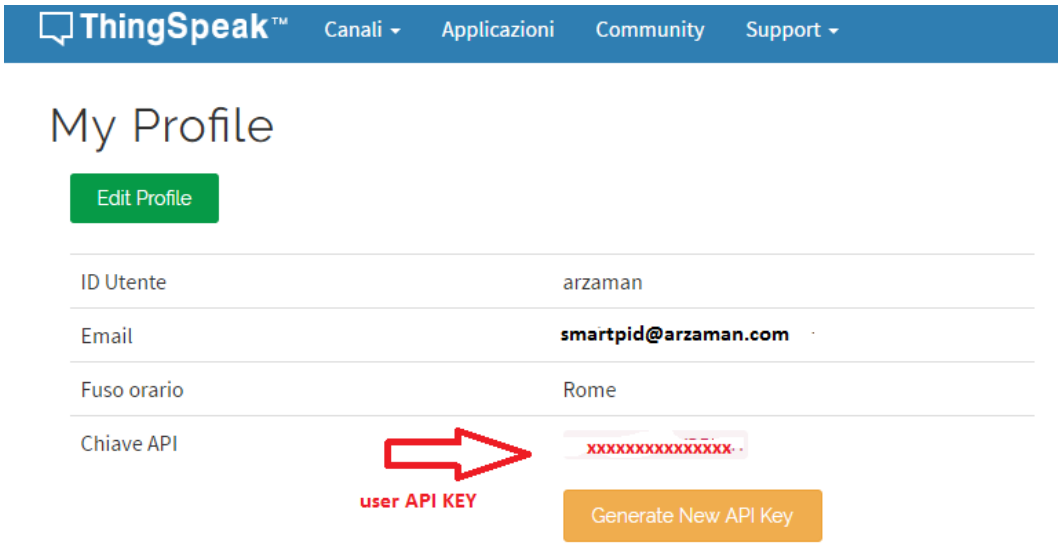
All the other fields can be left empty.



8. Press Save Channel and take note of your channel ID that should be uniquely assigned to your smartPID



9. Select ACCOUNT→ my profile and take note of the **USER API KEY**



The user API key is the way smartPID connects to your channel in a secure and trusted way and you have to insert it in the SmartPID configuration phase via the smartphone app.

3.2 WiFi Configuration

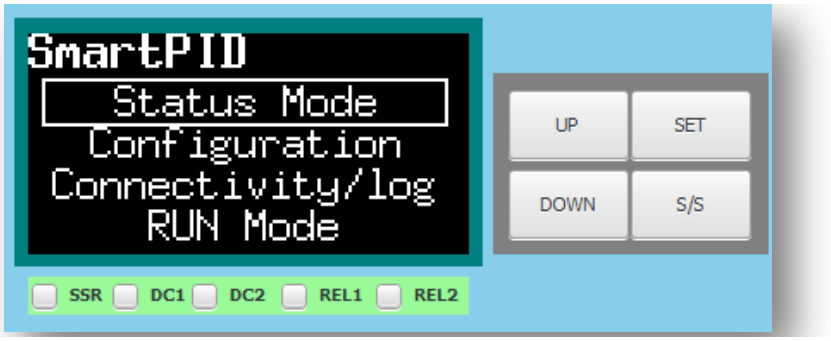
In this paragraph it is explained how to configure the SmartPID WiFi connection provided the internal ESP8266 WiFi module

In order to input in an easy way all the relevant data (SSID, WIFI PWD , user API KEY etc…) you can use your smartphone and a dedicated app [**android 5.0 or greater**] that connects directly to the SmartPID or in alternative a laptop PC using any browser

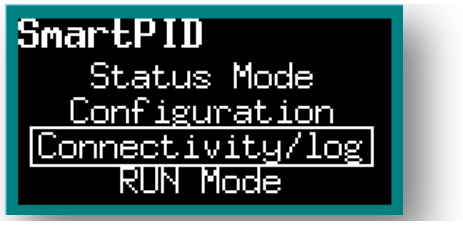
SmartPID configuration

The wifi module can be configured in various mode 0=off, 1= Client Mode , 2= AP mode, 3= Auto this mode will be used for the parameter configuration over WiFi

1. Take your smartPID device and take familiarity with the interface:



2. Select Connectivity/log on the smartPID device and press **SET** (upper right button):



3. Select Wi-Fi and press the **SET** button:

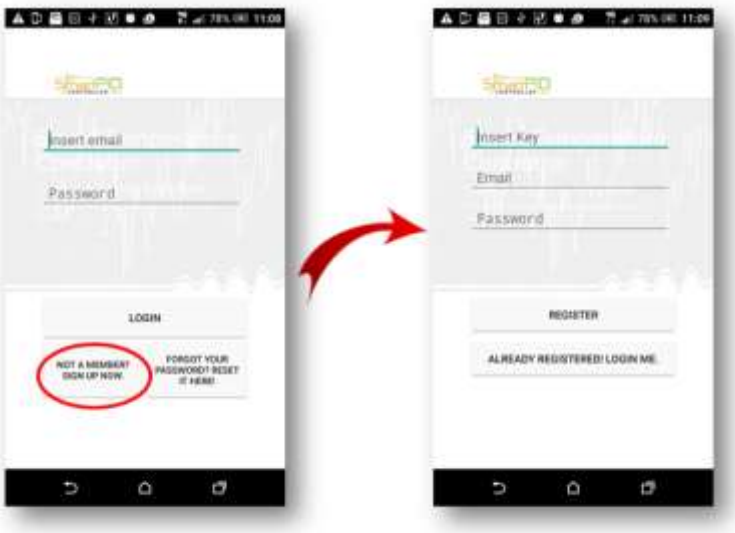
4. Set Wi-Fi mode to Auto and press the **SET** button
   in this configuration the wifi module broadcast a wifi signal SPC1000_xxxxxxx where xxxxxxx is your device serial number

Smartphone APP configuration

5. Go to the Play Store and install SmartPID app on your Android smartphone

6. Launch the app and sign up in order to link the app with the thingspeak profile

   You should register your SmartPID account using a valid email address , your password and insert the **USER API KEY** that you note down from your thingspeak profile [see chapter 3.2 , step 9]



7. Choose in the App SETUP-> Wi-Fi setup

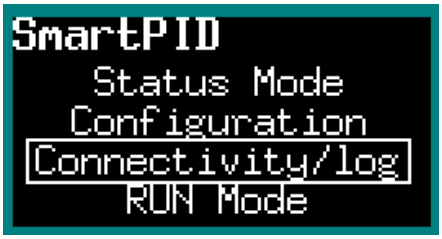
8. The app scans all available WiFi netwoks , keep you smartphone close to SmartPID. Choose SPC1000_xxxxxxxx network (xxxxxxxx is the serial number of your smartPID) and insert the following default password: ***smartpid!*** and press the **Next** button:

9. Choose your home Wi-Fi network, insert the password of your router and all the other parameters if necessary like the **channel ID** that you want to associate to your smartPID, the port and the baud rate and press the **Next** button

SmartPID connection to WiFi router

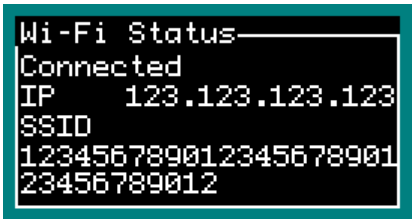
Once you have configured the smartPID it will reboot automatically and immediately after it try to connect to your WiFi router. If all the values are correct you should see the smartPID connected to your home WiFi router. In order to verify the proper configuration on smartPID follow below steps

10. Select Connectivity/log on the smartPID device from the top menu and press the **SET** button



11. Select Logging and set it to Wi-Fi and press the **SET** button

12. Select status menu and your IP address assigned by the router and your SSID wil be reported. I**P address** is useful if you want to control your smartPID from remote via app



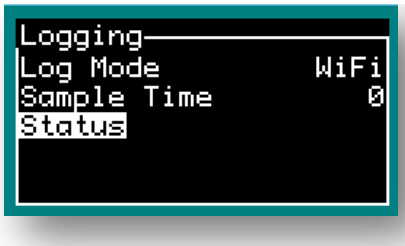
Process parameter logging

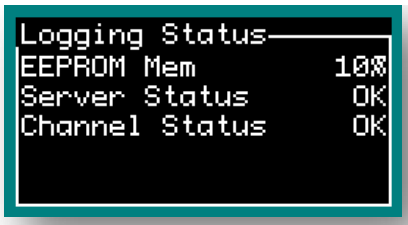
Now you can activate your logging via wifi and verify the thingspeak server connection

13. Got to the connectivity and log menu and select logging

14. activate the log via WiFi

15. Select the status and press the **SET** button





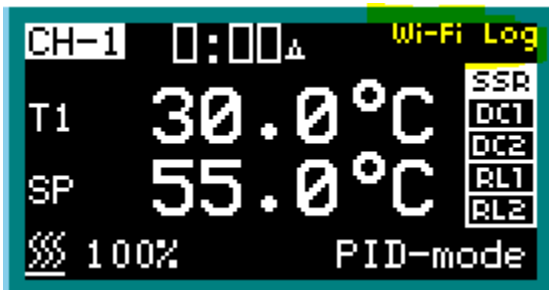
Now every time that SmartPID is in **RUN MODE** the process data for each channel (set point, current temperature and PWM percentage) are pushed to the things peak server.

You can visualize your data either on your channel on thingspeak account or on the smartphone app

For details on channel visualization or data manipulation/analysis/export on thingspeak server please refer to the thingspeak web site

https://thingspeak.com/

In the top right corner of the run mode screen you can see both indication of WiFi connection and logging activation



3.3 WiFi configuration via browser

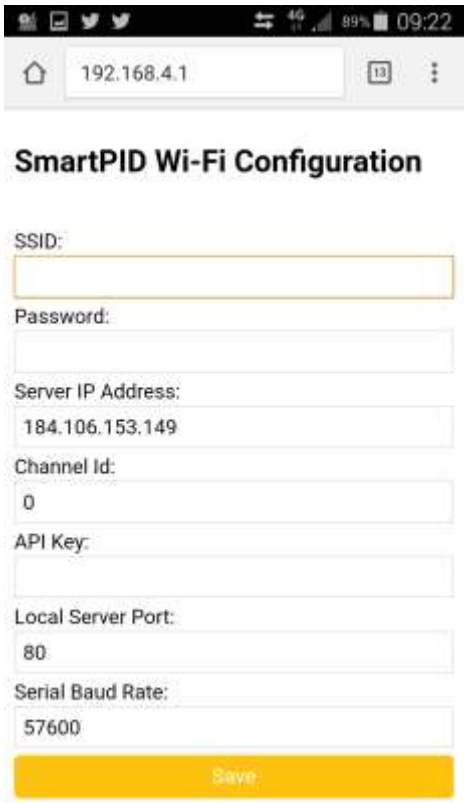
In case it's not possible to use an android (5 or greater) smartphone it's possible to complete the Wi-Fi configuration using a web browser on a laptop of even on the smartphone browser

The thingspeak configuration steps and smartPID steps are exactly the same, what changes is that instead using the app to input the parameter a web form inside a browser can be used

connect to the SPC1000_xxxxxxxxxxx wifi network using your wifi connection manger and open the browser at the following url

http://192.168.4.1

the browser will present a form to be filled



the value that are needed are

1. SSID of the Wi-Fi network to connected
2. PWD of the wifi network
3. Server IP address is the thingspeak server **184.106.153.149**

4. Channel ID is the value associated to the channel during the thingspeak setup
5. API KEY is the **user API KEY**
6. Serial baud rate is the communication speed between Wi-Fi module and the controller

once you have saved the value the smartPID reboot and the next steps are same as for the configuration via smartphone

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 seril 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.

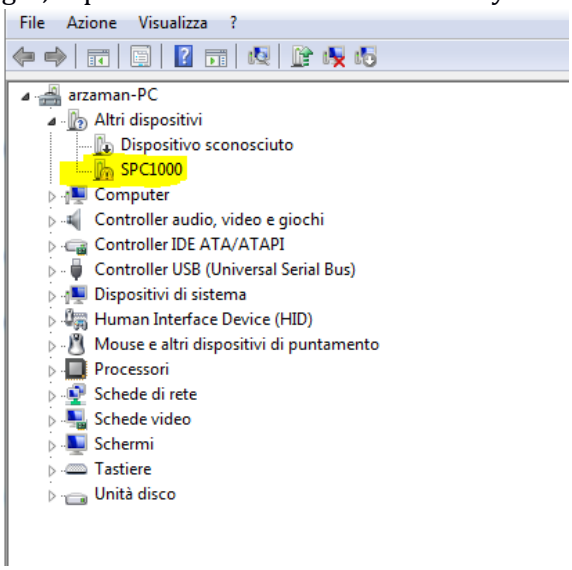
DOWNLOAD THE SAMD21 WINDOWS DRIVERS

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.)

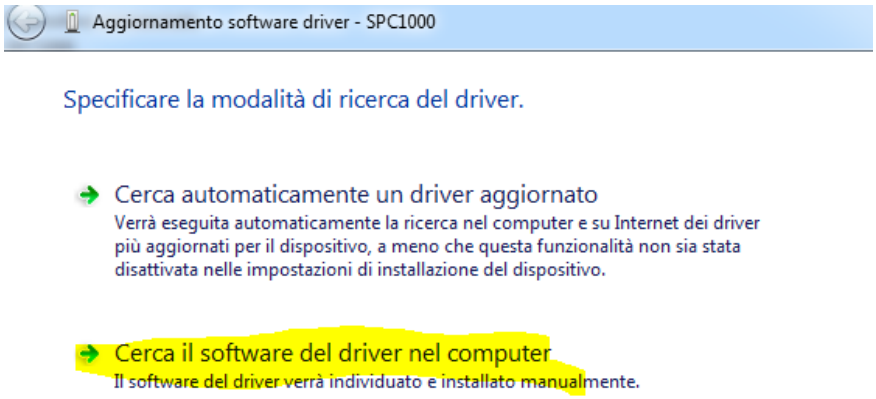
3.In the Device Manager, expand the**"Other devices" tree**-- you should see an entry for



Screenshot: Navigating the Device Manager

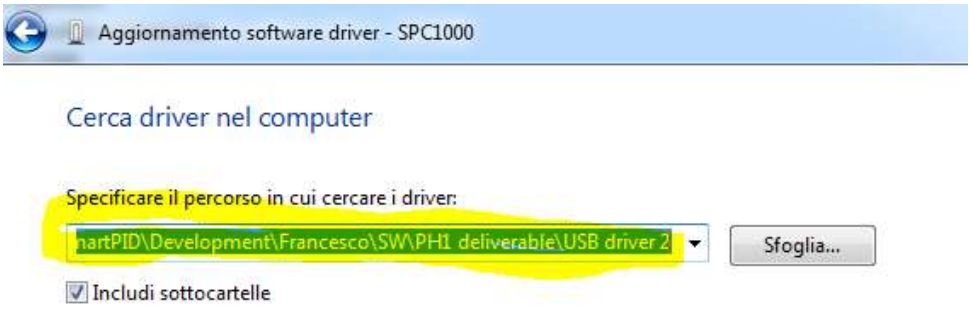
4."Unknown Device",**Right-click on SPC1000 and select Update Driver Software...**

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



Screenshot: Browse your computer for the driver

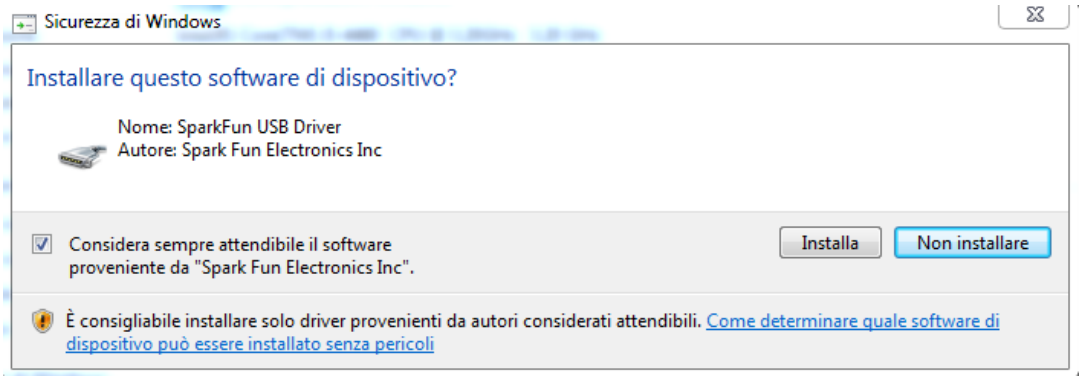
6.**Paste the directory location** of your arzaman.inf and files into the search location. Then hit "Next".



Screenshot: Setting the driver location

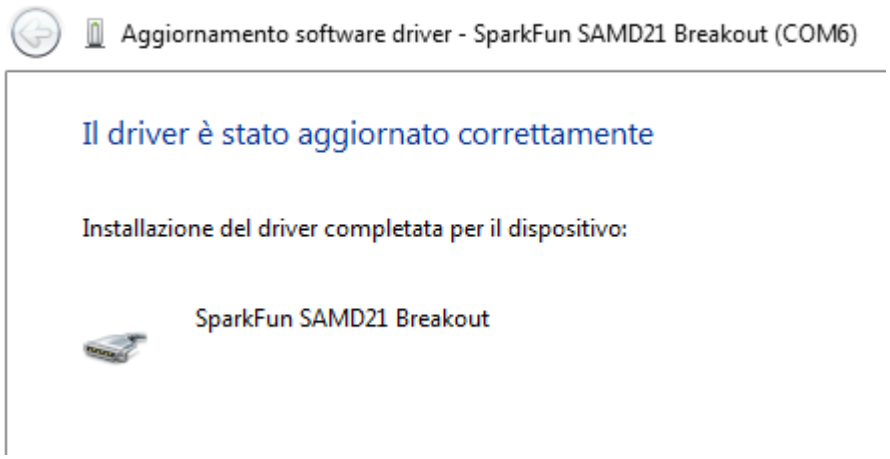
7.**Click "Install"**when the next pop-up questions if you want to install the driver.

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

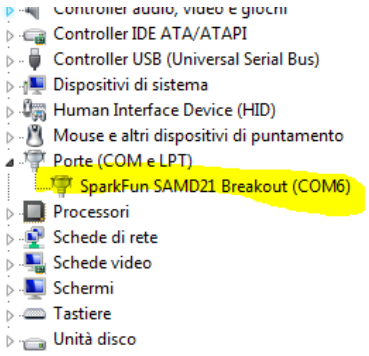


Screenshot: security warning

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.

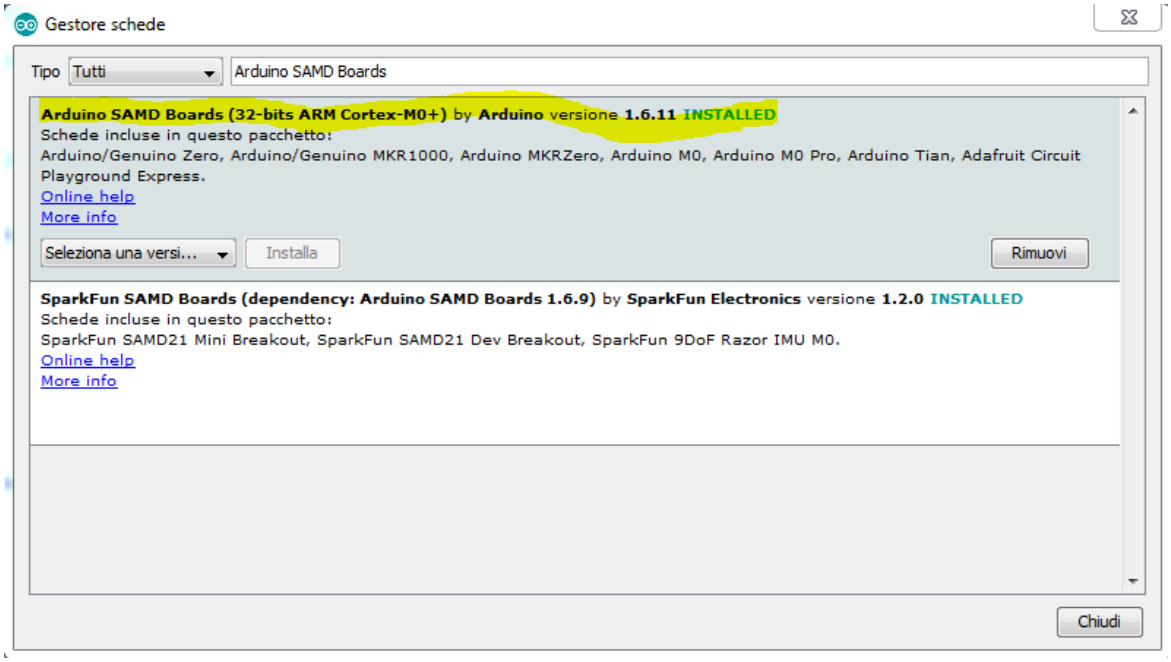
4.3 Setting up Arduino IDE

SAMD21 MCU is fully supordet 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 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)

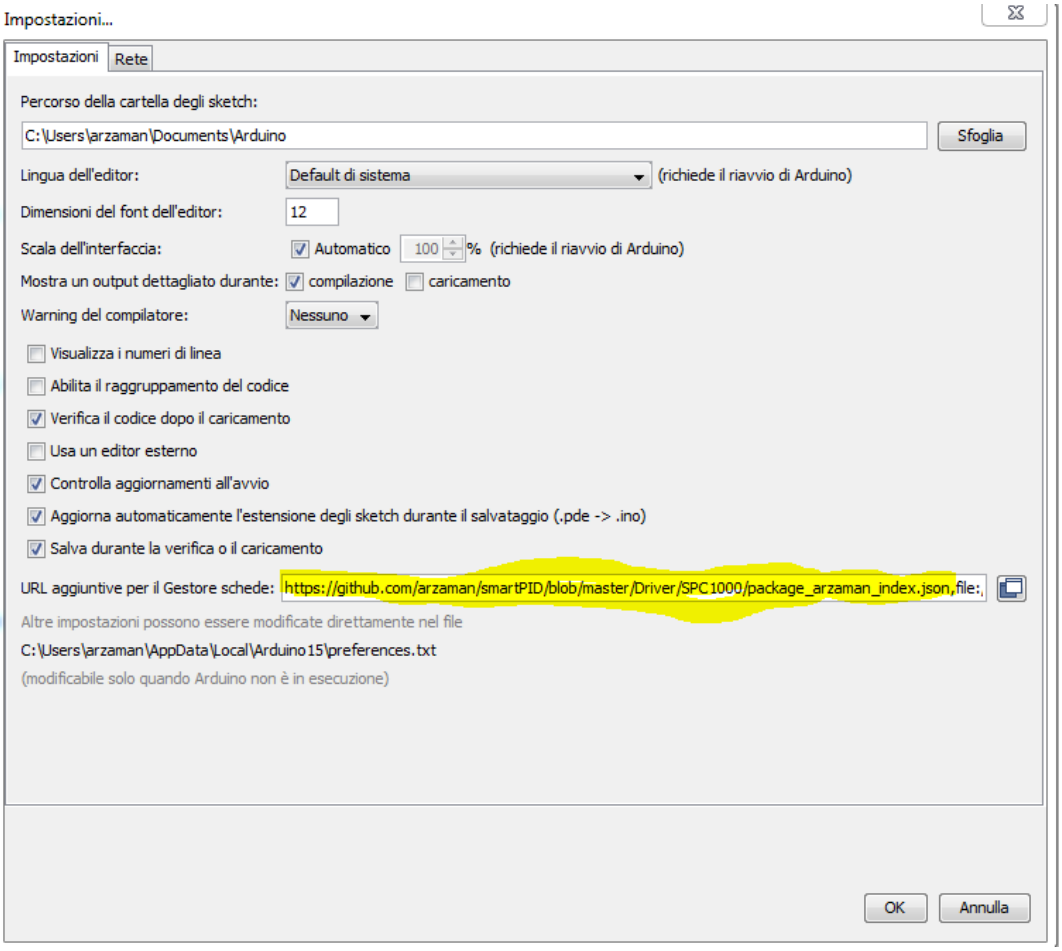


Downloading and installing the tools may take a couple minutes – arm-gcc in particular will take the longest, it's about 250MB unpacked.

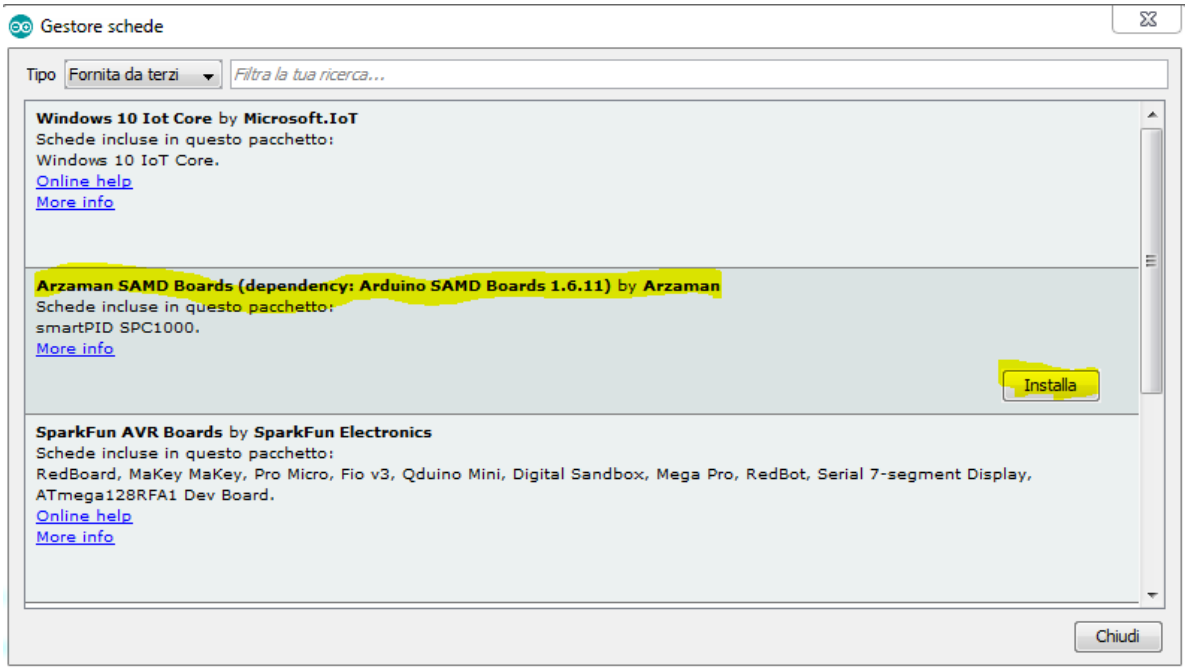
Once installed, Arduino-blue "Installed" text should appear next to the SAMD boards list entry.

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:

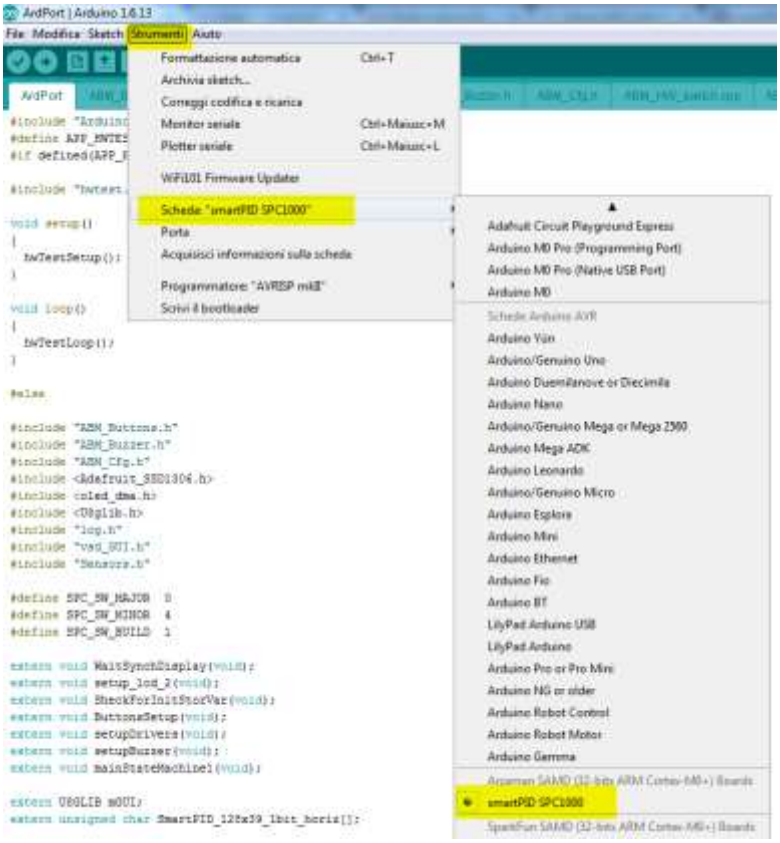
https://raw.githubusercontent.com/arzaman/smartPID/master/Driver/SPC1000/package_arzaman_index.json



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 bord defintion you should see two one entries in your Tools > Board list: Select your **SmartPID - SPC1000** Board.


ArdPort | Arduino 1.6.13
File Modifica Sketch Strumenti Aiuto
Formattazione automatica Ctrl+T
Archivia sketch...
Correggi codifica e ricarica
Monitor seriale Ctrl+Maiusc+M
Plotter seriale Ctrl+Maiusc+L
WiFi101 Firmware Updater
Scheda: "smartPID SPC1000"
Porta
Acquisisci informazioni sulla scheda
Programmatore: "AVRISP mkII"
Scrivi il bootloader
Adafruit Circuit Playground Express
Arduino M0 Pro (Programming Port)
Arduino M0 Pro (Native USB Port)
Arduino M0
Schede Arduino AVR
Arduino Yún
Arduino/Genuino Uno
Arduino Duemilanove or Diecimila
Arduino Nano
Arduino/Genuino Mega or Mega 2560
Arduino Mega ADK
Arduino Leonardo
Arduino/Genuino Micro
Arduino Esplora
Arduino Mini
Arduino Ethernet
Arduino Fio
Arduino BT
LilyPad Arduino USB
LilyPad Arduino
Arduino Pro or Pro Mini
Arduino NG or older
Arduino Robot Control
Arduino Robot Motor
Arduino Gemma
smartPID SPC1000

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.

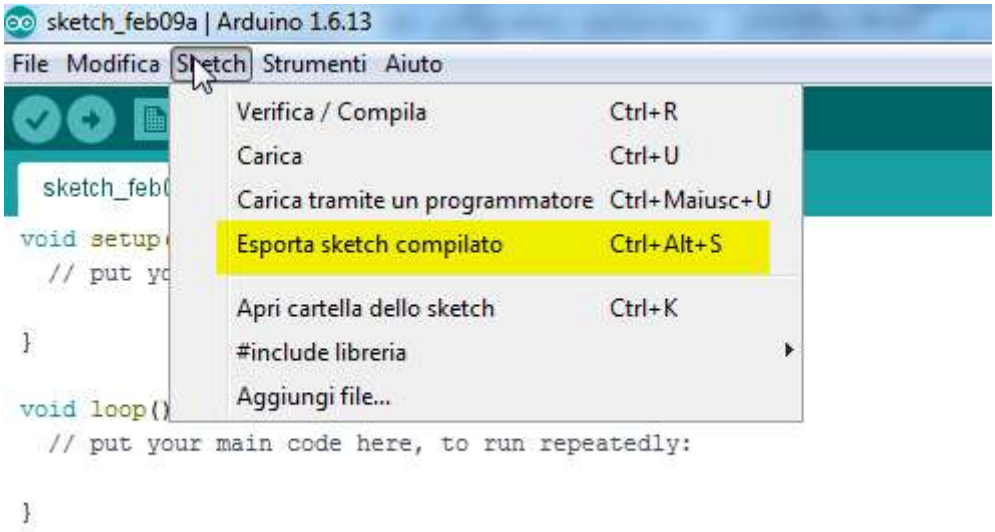
In order to support some basic functions you need to install locally the library that are available on github

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 uing BOSSA tool insuide 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 complied 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