

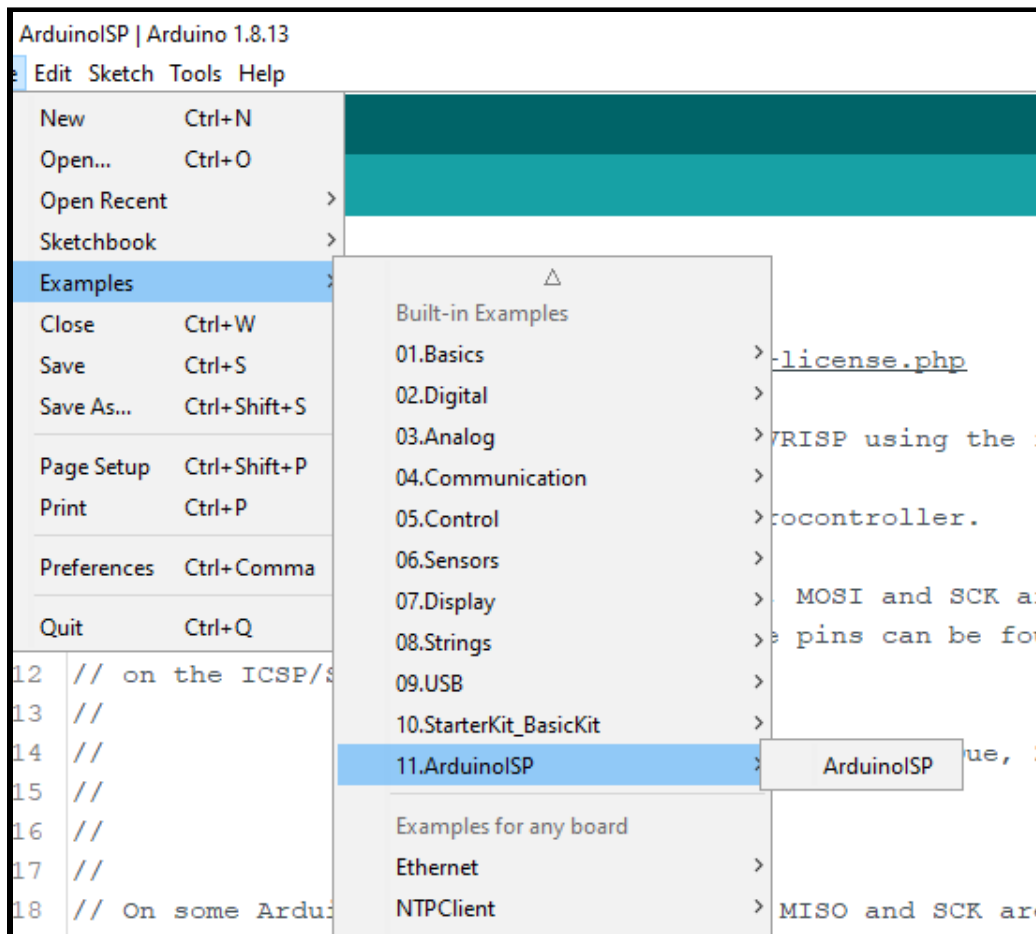
Safmet Programming Guide :

Programming for Atmega328p :

The following are the 3 steps needed to upload code in any **Atmega328p** .

Step 1

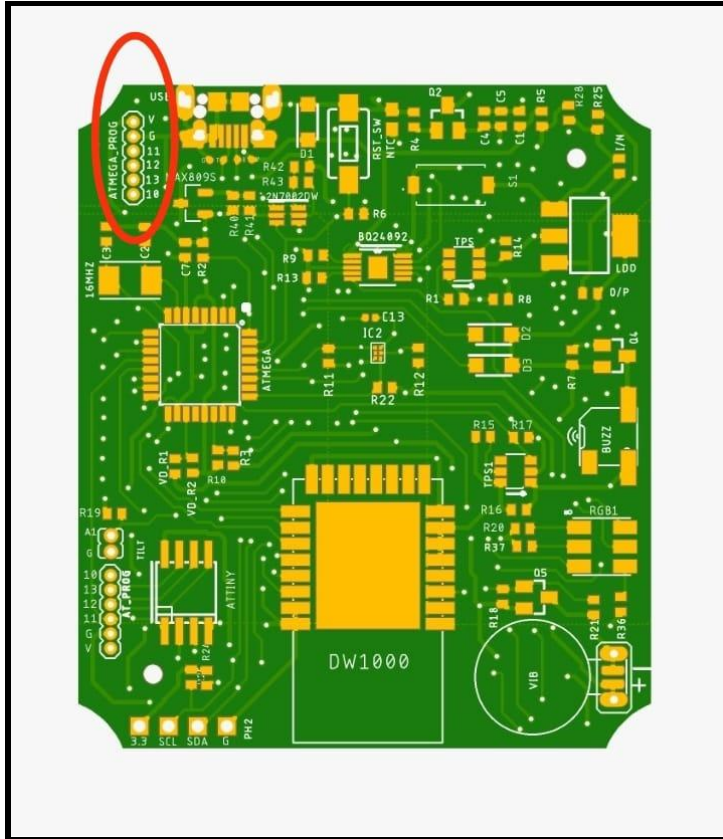
- A. The nano programmer first programmed an atmega328p ic with an ArduinoISP code.



- B. Connect capacitor 220uf 16v between RST and GND pin of arduino nano .
C. Connect spi pins from nano to 6pin molex connector

Step 2

- A. First connect Arduino Nano Programmer to ATMEGA_PROG on SafmetPCB
B. The nano programmer red wire vcc pin should connect to V pin at ATMEGA_PROG on Safmet PCB

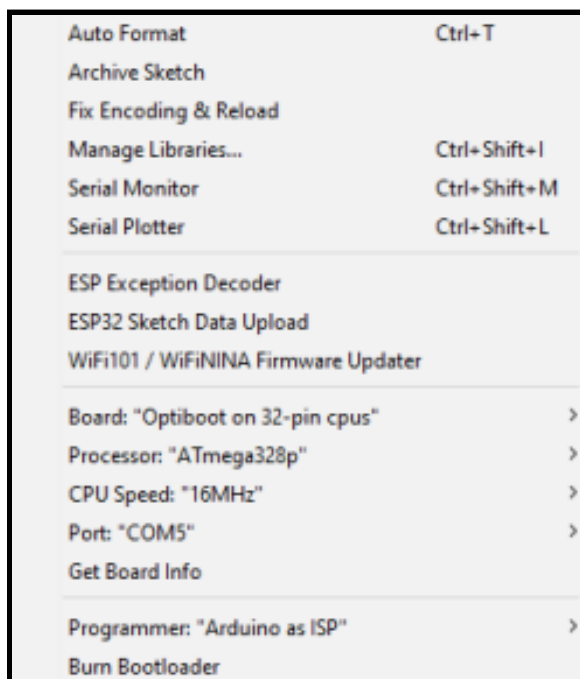


Step 3

a. Burn Bootloader with the following settings

To bootload the atmega328p with the Arduino IDE, first, we need to add the optiboot Support to the Arduino IDE. For that, go to **File > Preferences** and add the below link in the Additional Boards Manager URLs and click 'OK.'

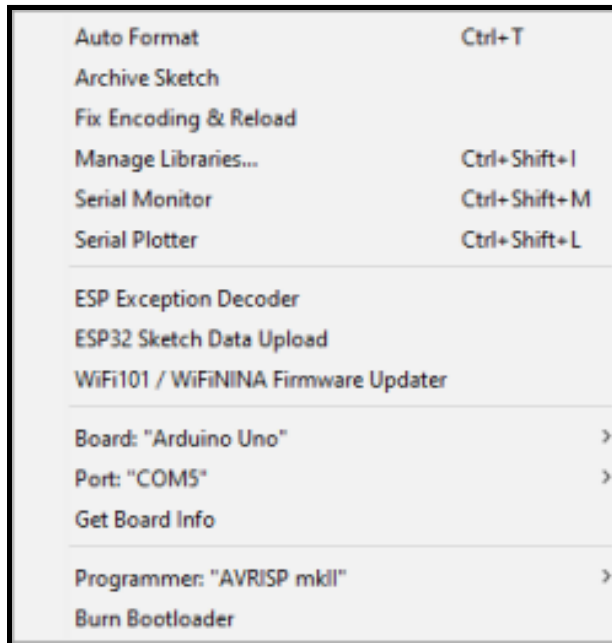
https://github.com/Optiboot/optiboot/releases/download/v8.0/package_optiboot_optiboot-additional_index.json



To burn Optiboot onto an Arduino board

1. Select board type
2. Connect the board to an ISP programmer
My programmer is nano programmer
3. Choose Optiboot board type to upload a sketch-here using "Optiboot on 32-pins cpus"
4. Menu Tools - Burn Bootloader

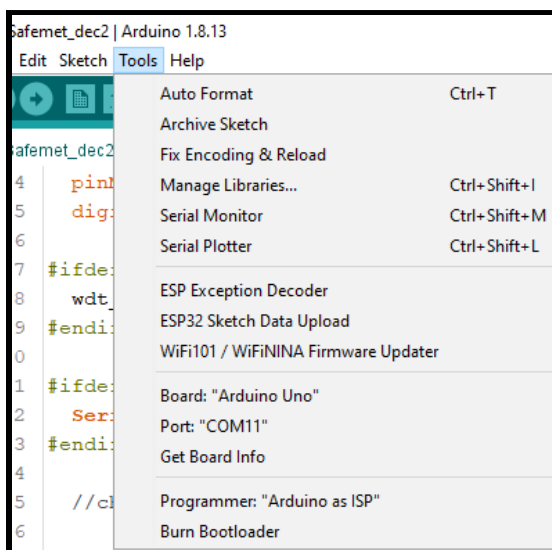
- b. If successful, go to the next step, else check wiring
- c. Now remove the nano programmer, connect USB - TTL with the USB connector to the PCB. If the battery is not charged, R5 should be bypassed and provided with a power supply from USB. In case the battery is charged, don't power the PCB from USB and desolder R5.
- d. Then program with the following settings



- e. Then upload the code and while uploading press reset twice.

Note: If USB - TTL is not working then program with nano programmer only by selecting below options and then upload the code(Safmet.ino) by option upload using a programmer.

Step 1



Programming for ATtiny :

The following are the steps needed to upload code in **Attiny**.

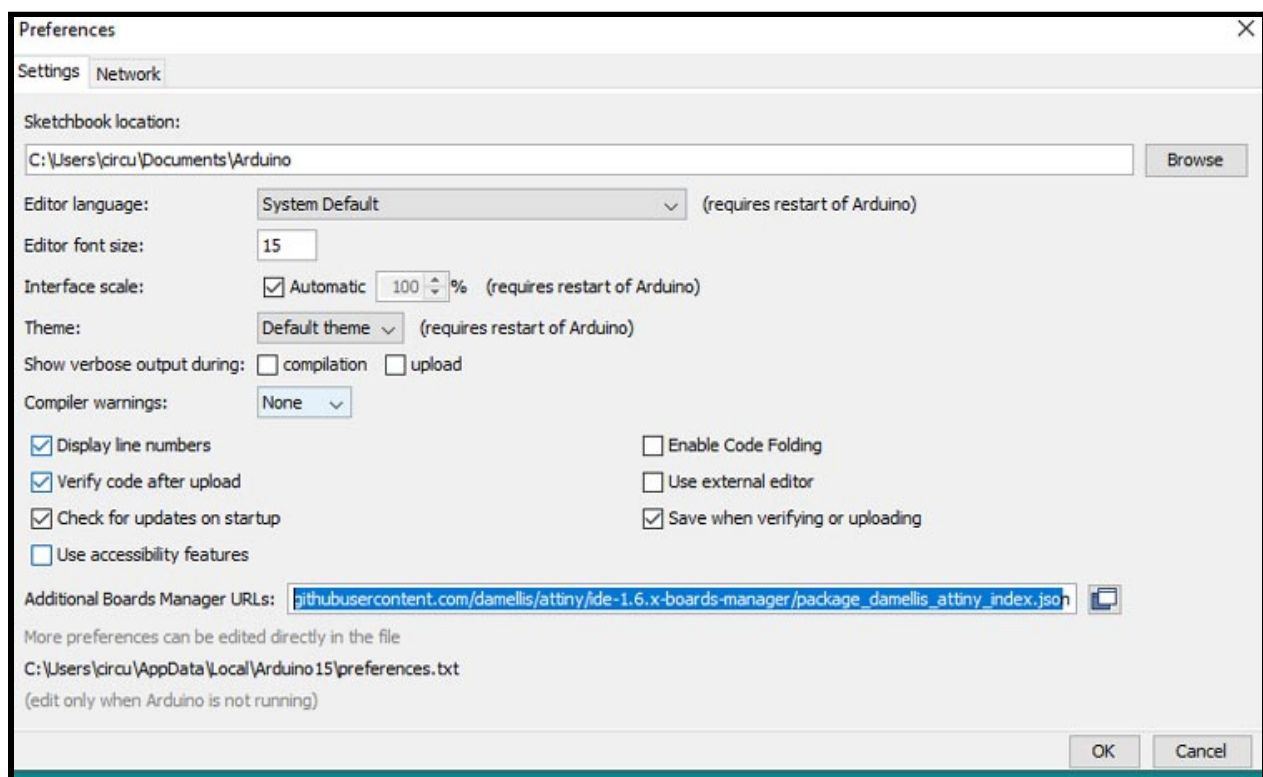
1. For ATtiny85

Step 1 :

Install hardware package for ATtiny85

To program the ATtiny85 with Arduino IDE, first, we need to add the ATtiny85 Support to Arduino IDE. For that, go to **File > Preferences** and add the below link in the Additional Boards Manager URLs and click 'OK.'

https://raw.githubusercontent.com/damellis/attiny/ide-1.6.x-boards-manager/package_damellis_attiny_index.json



Step 2

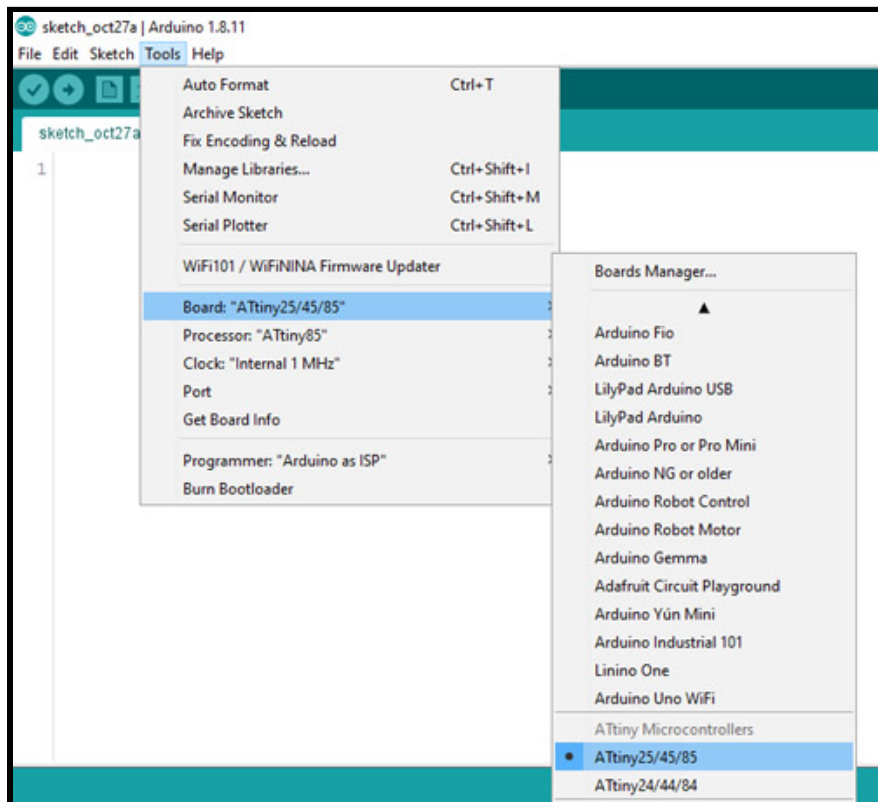
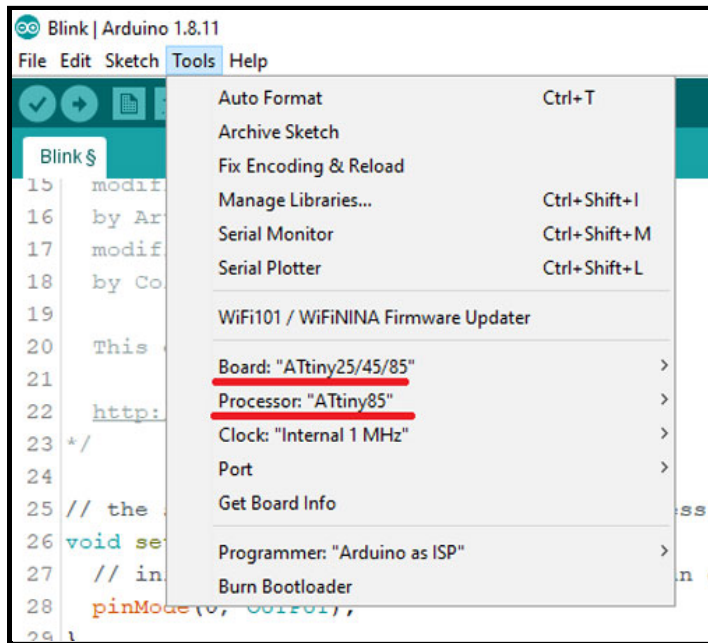
After that, go to **Tools > Board > Board Manager** and search for 'attiny' and install the latest version.



Step 3

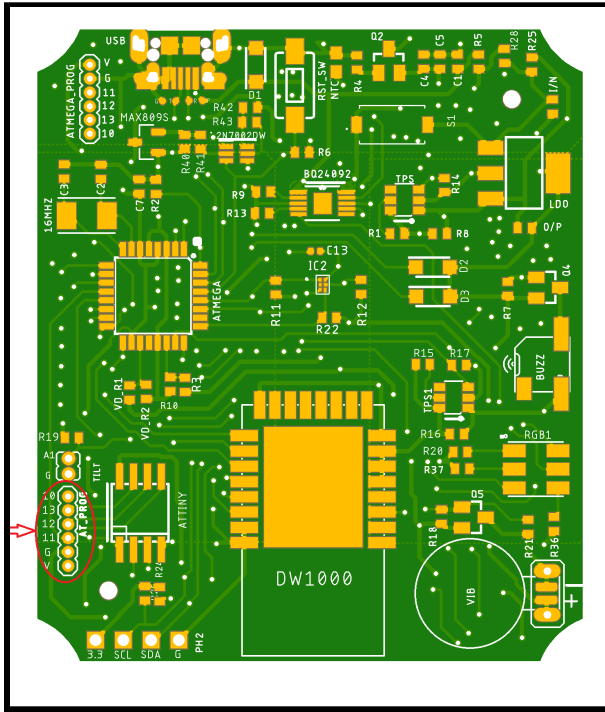
After installing it, now you would be able to see a new entry in the Board menu titled 'Attiny25/45/85'.

Now go to **Tools -> Board** and select "Attiny25/45/85", then select **ATtiny85** under **Tools > Processor**.



Step 4

- First connect Arduino Nano Programmer to AT_PROG on Safmet PCB
- The nano programmer red wire vcc pin should connect to V pin at AT_PROG on Safmet PCB.



Step 5

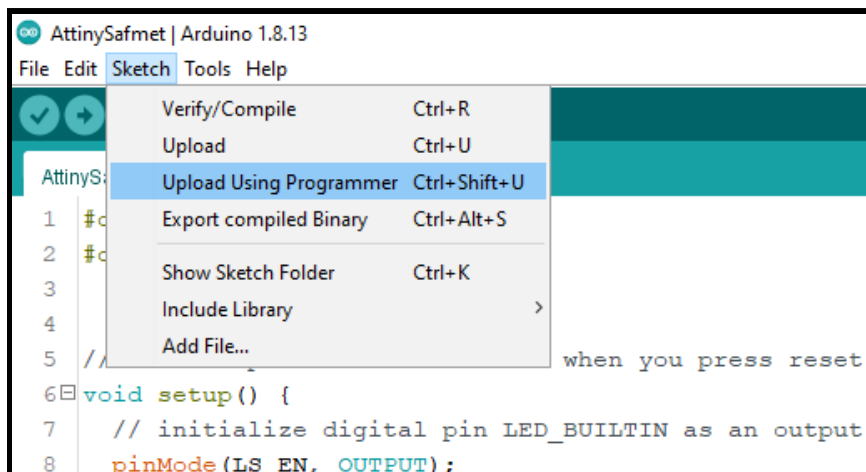
Burn Bootloader

(You only need to do this once per chip)

Go to Arduino IDE -> Tools -> Burn Bootloader

Step 6

Then program with a nano programmer only by selecting the above options and then upload the code(AttinySafmet.ino) by option upload using a programmer

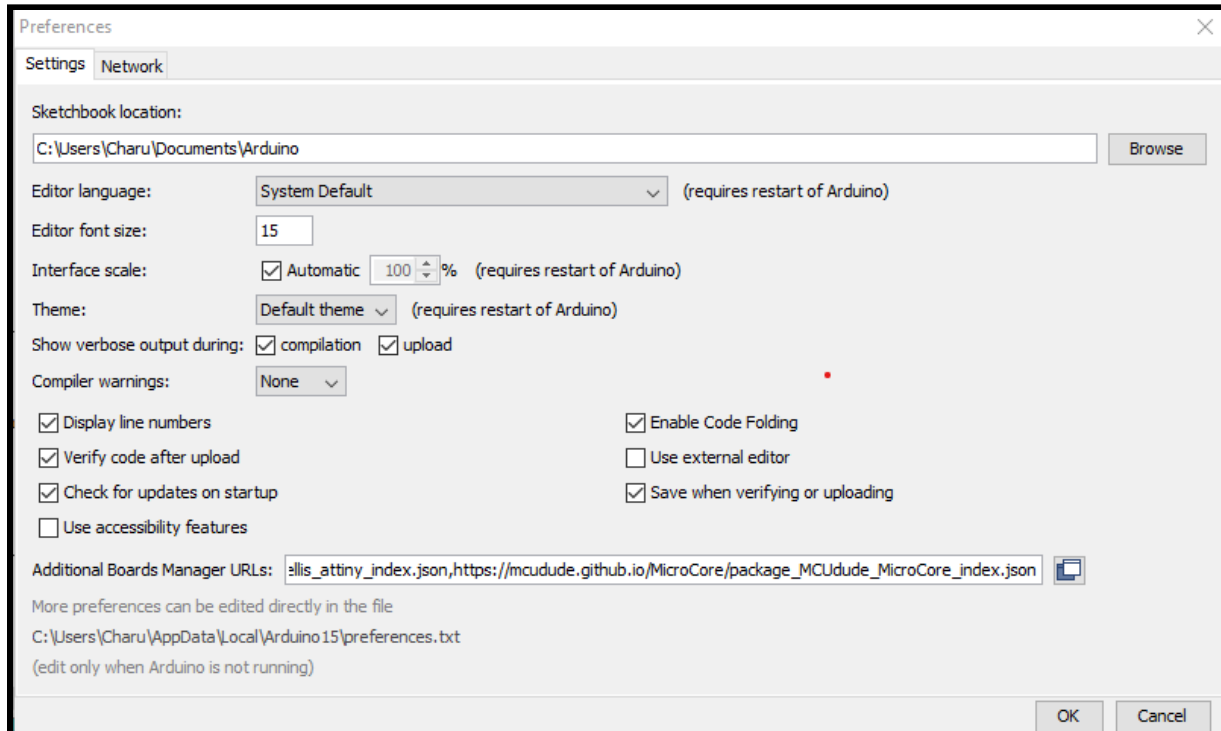


2. For ATtiny13A

Step 1

Install hardware package for ATtiny13A

To program the ATtiny85 with Arduino IDE, first, we need to add the ATtiny85 Support to Arduino IDE. For that, go to **File > Preferences** and add the below link in the Additional Boards

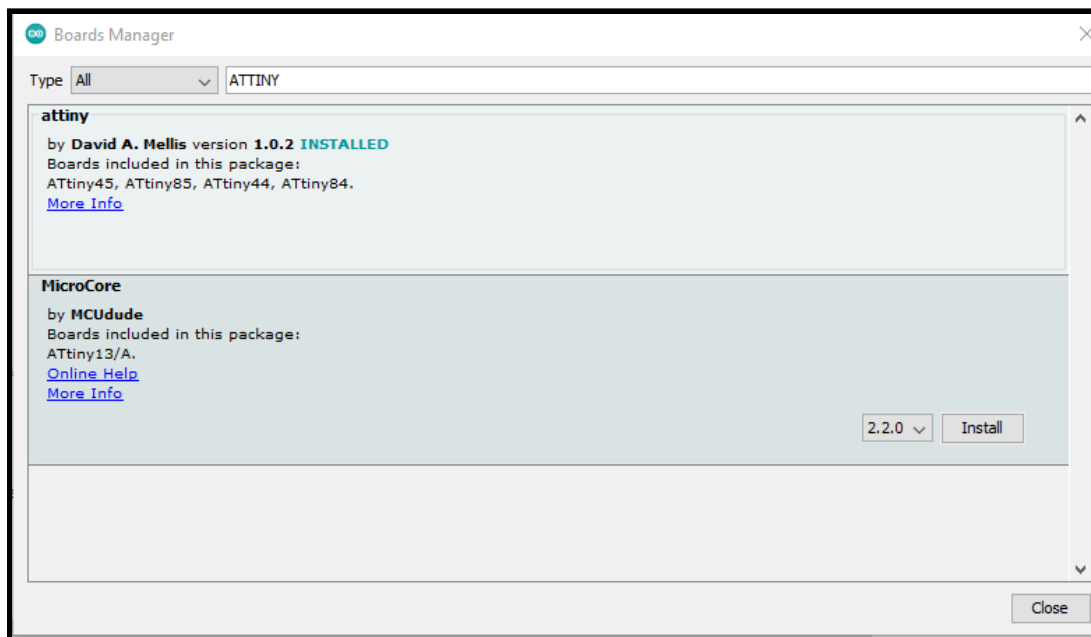


Manager URLs and click 'OK.'

https://mcudude.github.io/MicroCore/package_MCUdude_MicroCore_index.json

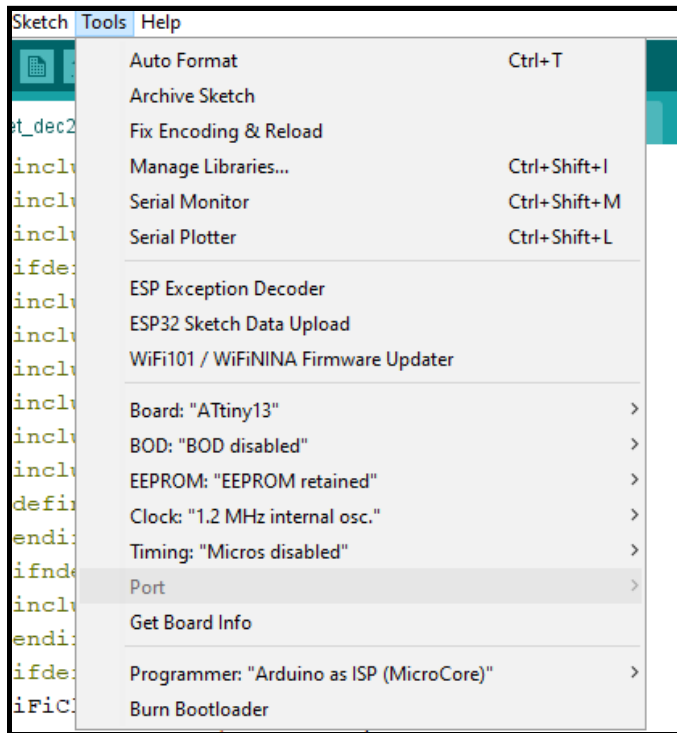
Step 2

Open Arduino IDE -> Tools -> Board -> Boards manager. Find MicroCore and click Install.



Step 3

Go to Arduino IDE -> Tools and select the options below:



Step 4

Burn Bootloader

(You only need to do this once per chip)

Go to Arduino IDE -> Tools -> Burn Bootloader

Step 5

Then program with a nano programmer only by selecting the above options and then upload the code(AttinySafmet.ino) by option upload using a programmer.

