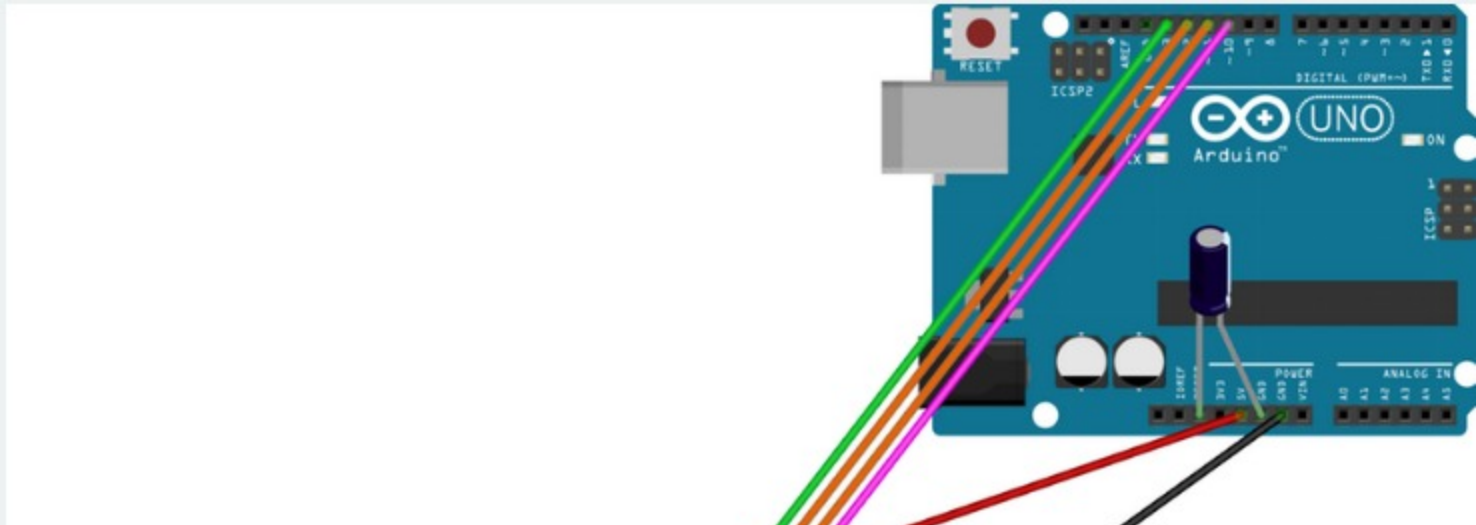


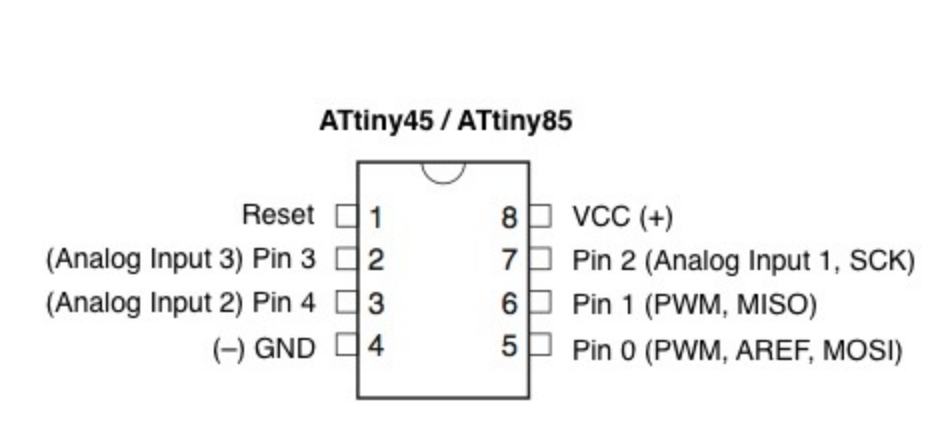
Programming ATtiny85 with Arduino Uno

Step by step on how to program a ATtiny85 microcontroller with Arduino Uno development board.

attiny85 arduino

ABOUT THIS PROJECT

I am working on a project which requires reading multiple sensor data on different locations. These require only few PWM pins so using multiple Arduino Uno would be expensive and unnecessary. So I decided to use ATtiny85 microcontroller in place of Arduino Uno development boards. ATtiny85 is a cheap and powerful alternate when u don't need too many PWM pins. Since ATtiny85 is just a microcontroller we need a Arduino Uno to program it. In this project I will explain how to do it. Below is the pin configuration of ATtiny85 and the datasheet can be found here http://www.atmel.com/images/atmel-2586-avr-8-bit-microcontroller-attiny25-attiny45-attiny85_datasheet.pdf.



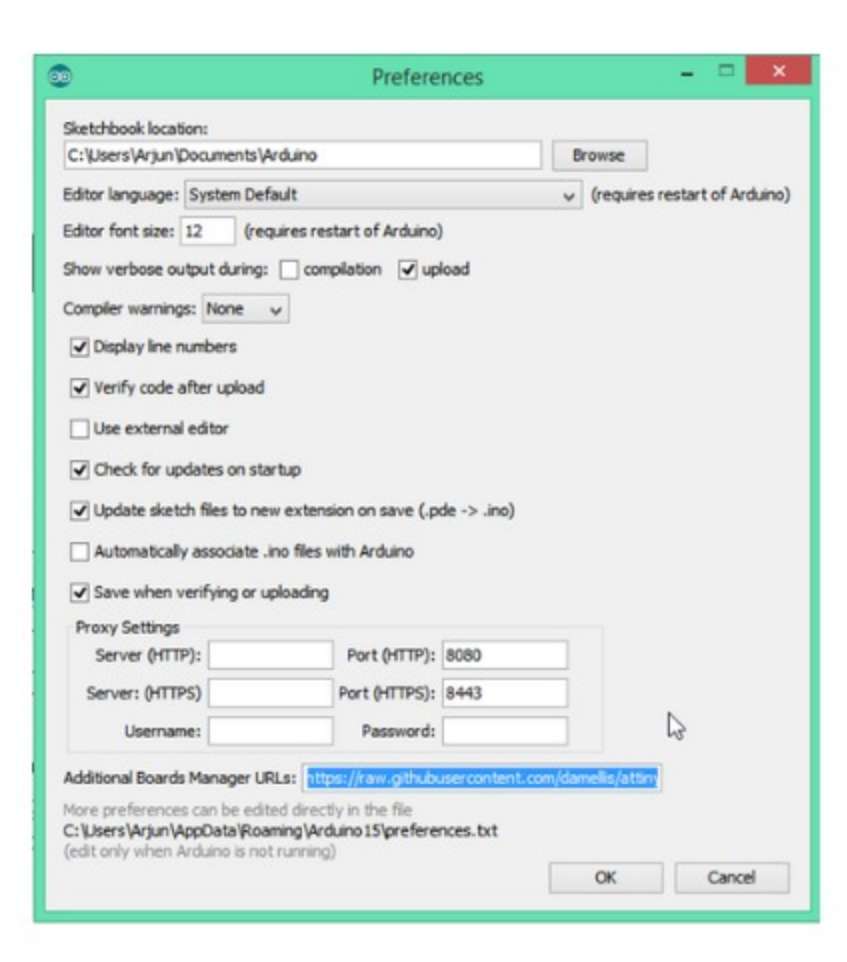
ATtiny85 Pin Configuration

Configuring Arduino Uno as a ISP (In-System Programming)

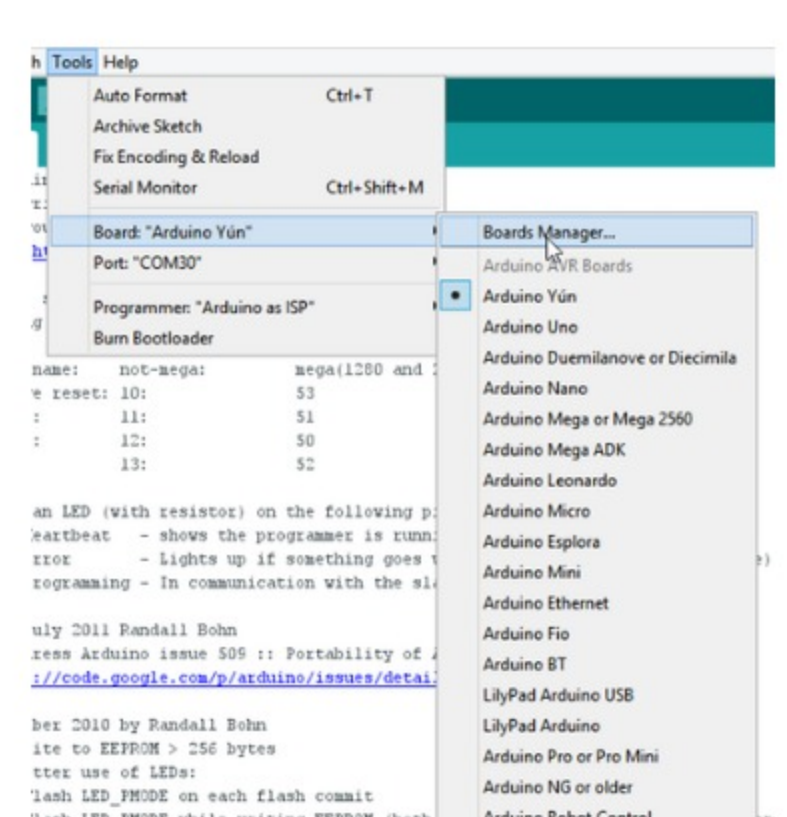
To program the ATtiny85 we need to first set Arduino Uno in ISP mode. Connect your Arduino Uno to the PC. Open Arduino IDE and open the ArduinoISP example file (File -> Examples -> ArduinoISP) and upload it.

Adding ATtiny85 Support to Arduino IDE

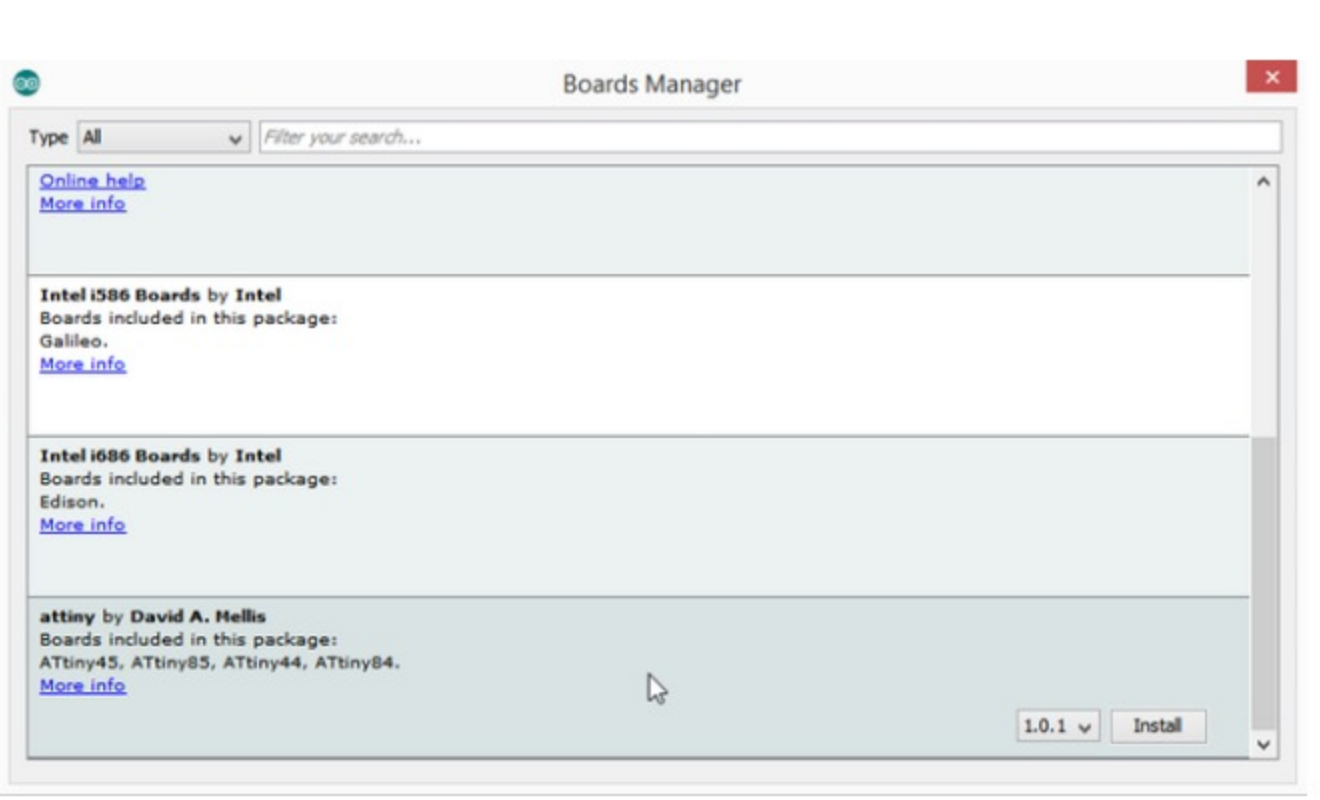
By default Arduino IDE doesn't support ATtiny85 so we should add ATtiny boards to Arduino IDE. Open File -> Preferences and in the Additional Boards Manager URLs give this url https://raw.githubusercontent.com/damellis/attiny/ide-1.6.x-boards-manager/package_damellis_attiny_index.json.



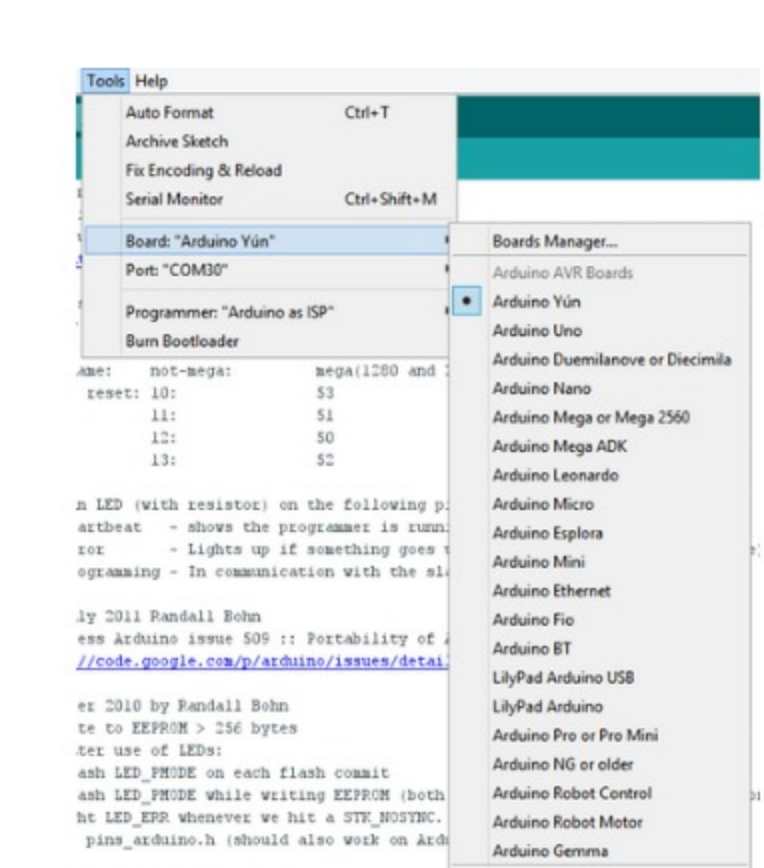
After this is done open Tools -> Board -> Board Manager



After opening Board Manager scroll down the list where it says "attiny by Davis A. Mellis". Click on that and install it.



After installing now you would be able to see a new entry in the Board menu

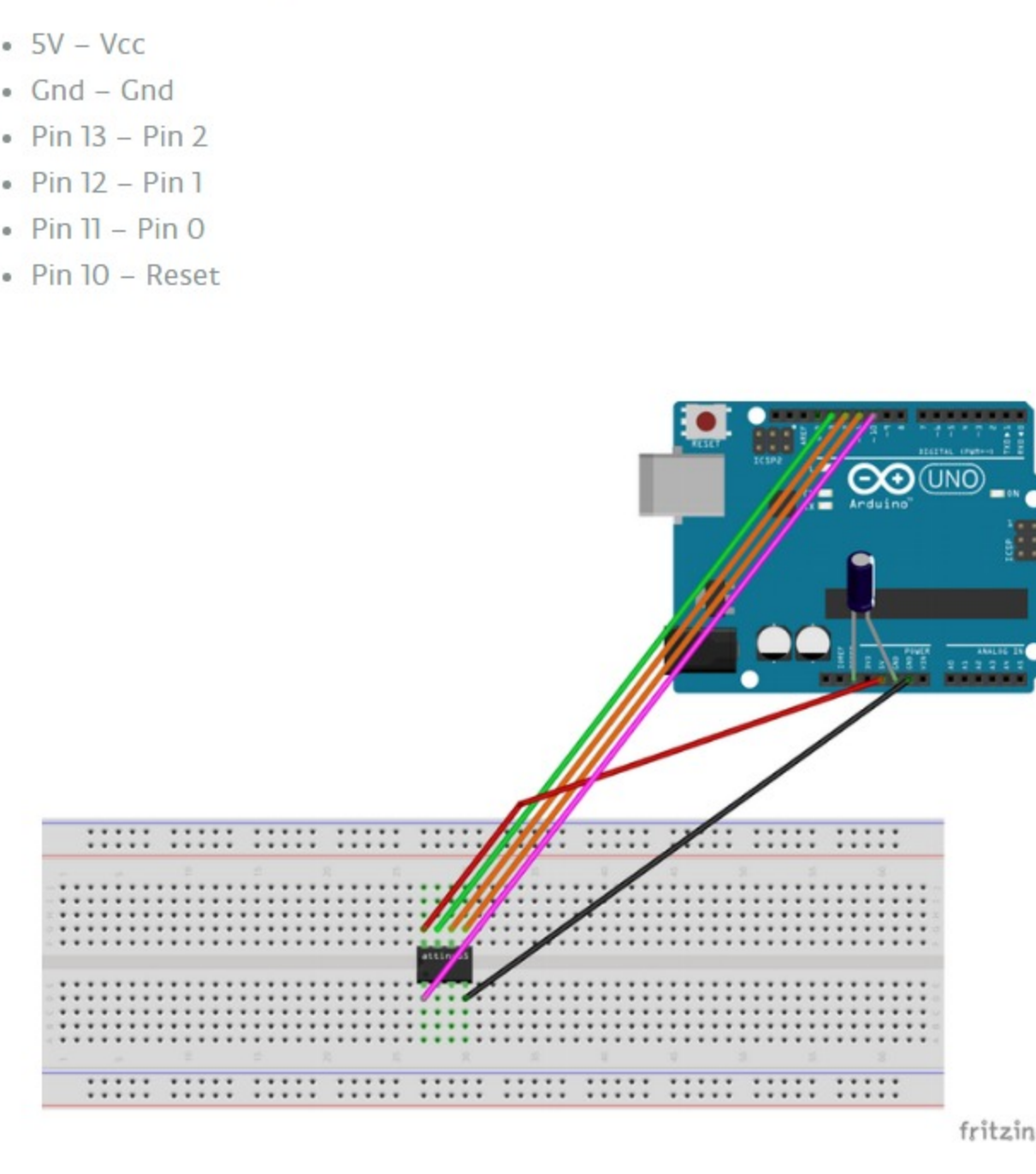


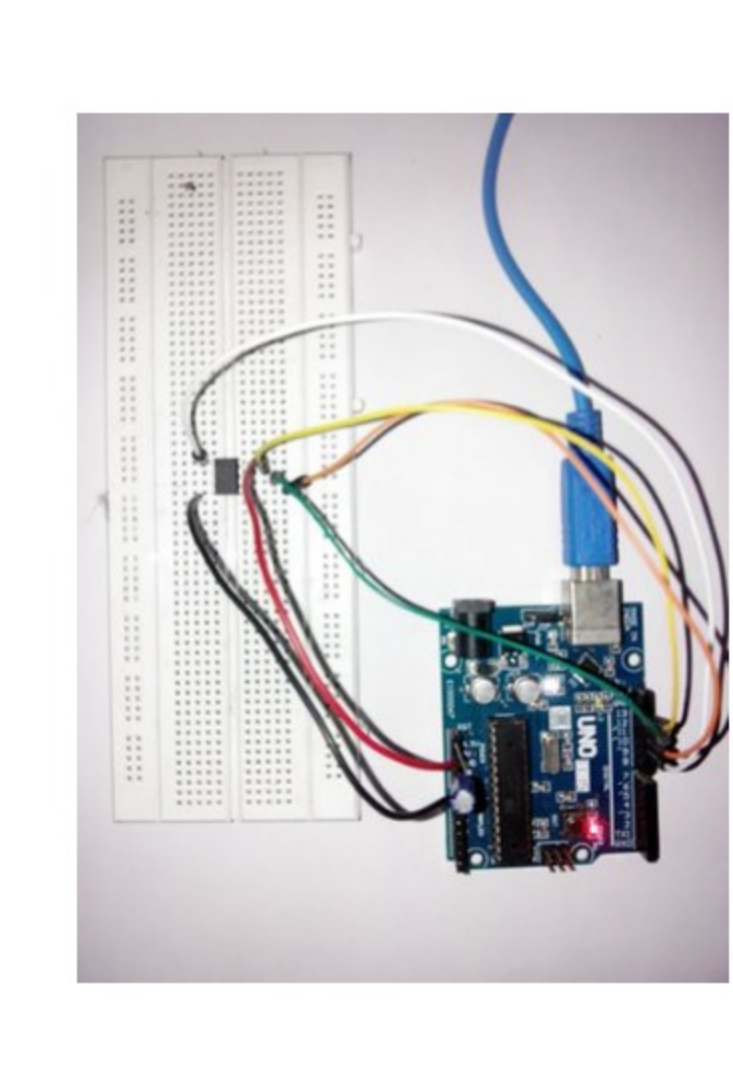
Connecting ATtiny85 with Arduino Uno

Now with all the above things ready we will start programming the attiny85. Connect the arduino uno to attiny85 using breadboard as below.

Arduino Uno – ATtiny85

- 5V – Vcc
- Gnd – Gnd
- Pin 13 – Pin 2
- Pin 12 – Pin 1
- Pin 11 – Pin 0
- Pin 10 – Reset

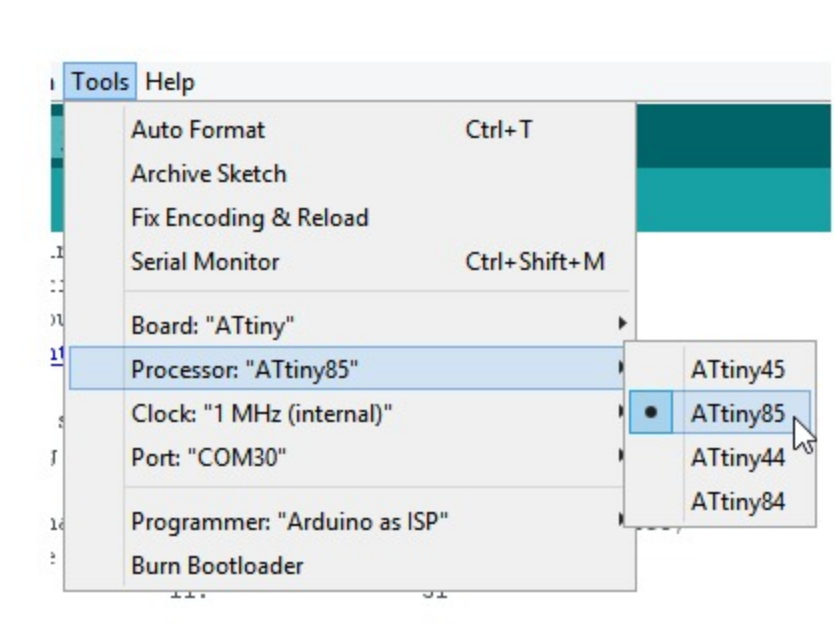


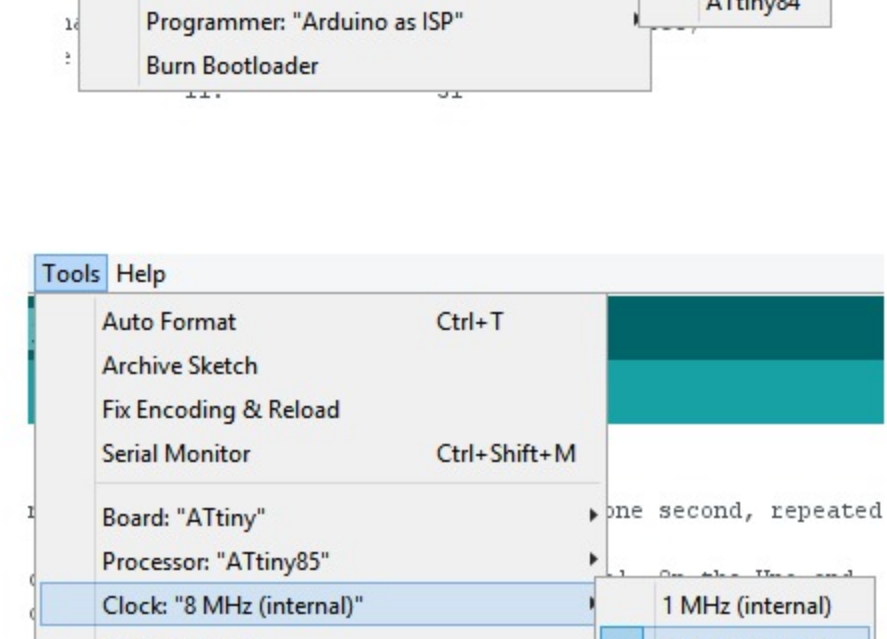


Add a 10uF capacitor between RESET and GND in arduino. This is to avoid arduino from being auto reset when we upload the program to attiny85. If you are using a electrolytic capacitor make sure the anode goes in GND of uno.

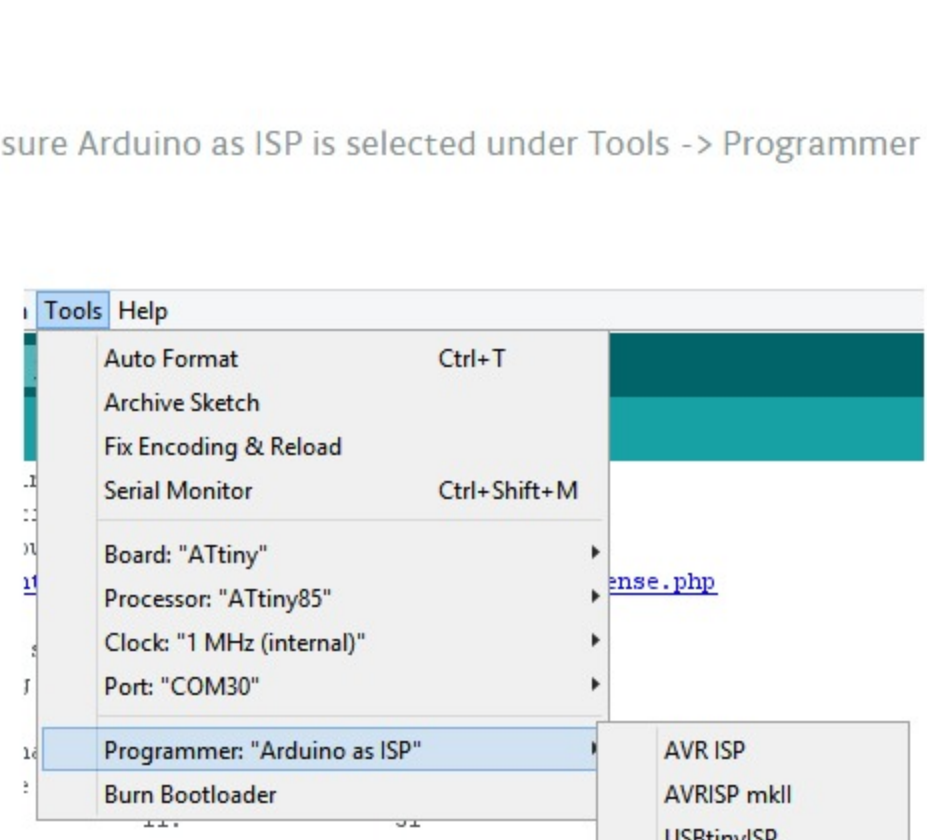
Uploading program to ATtiny85

Now back to Arduino IDE. Select ATtiny under Tools -> Board. Then select ATtiny85 under Tools -> Processor. And select 8 MHz (internal) under Tools -> Clock.

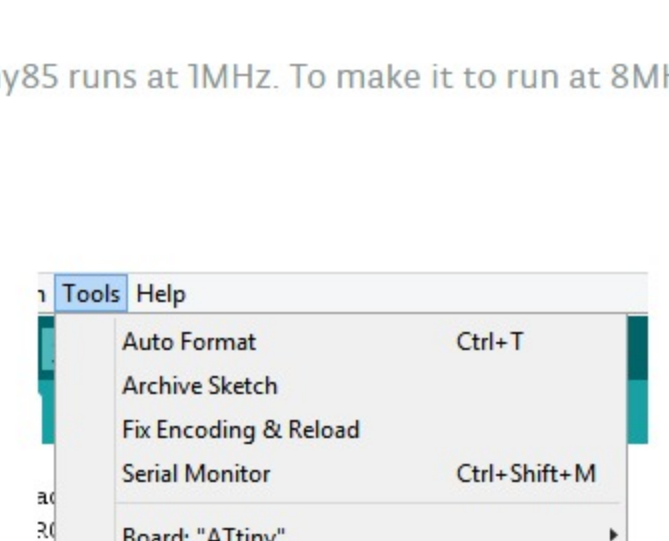


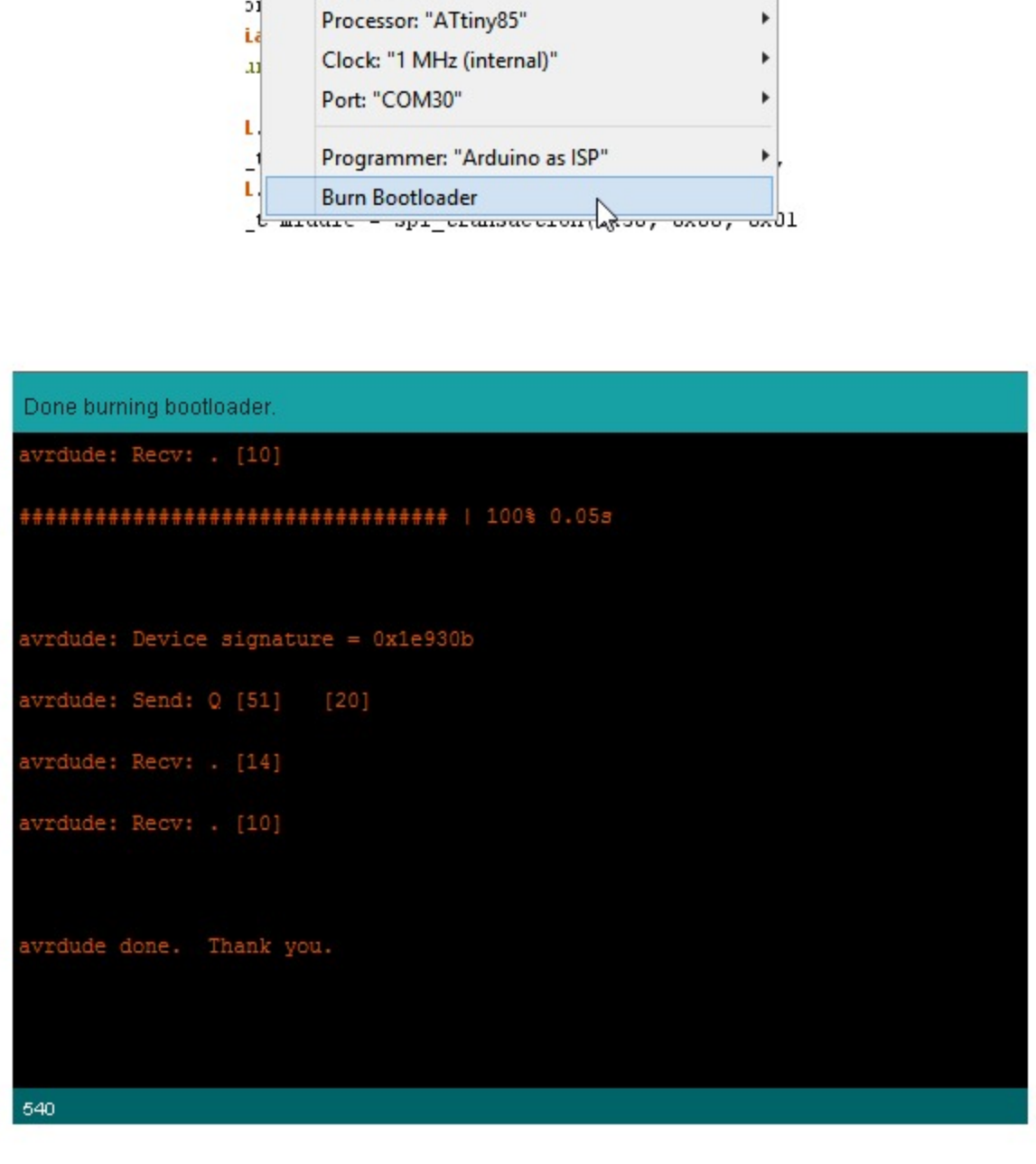


Then make sure Arduino as ISP is selected under Tools -> Programmer



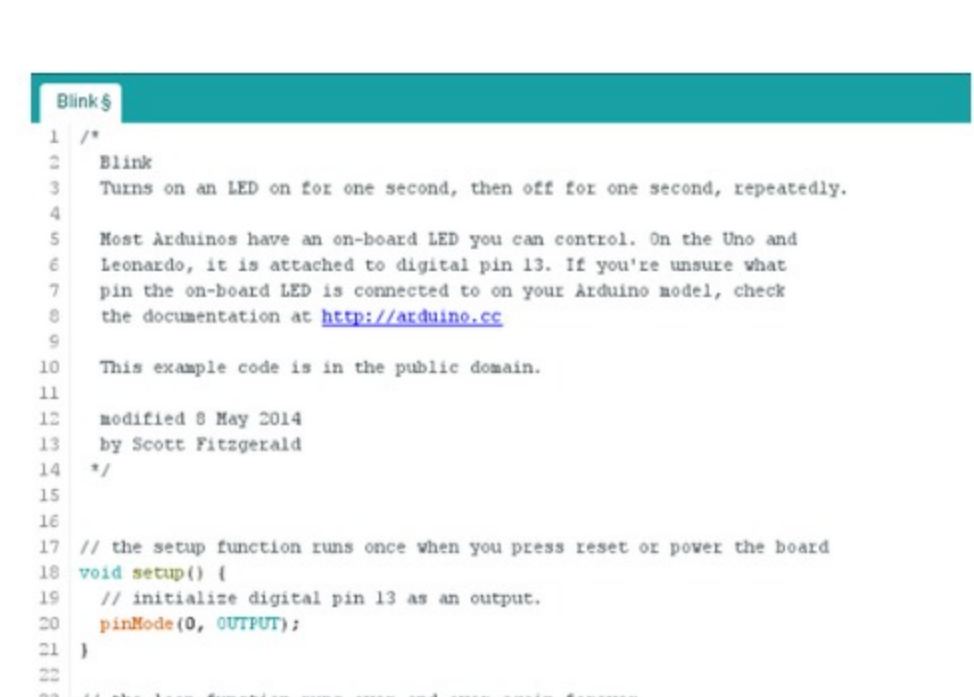
By default the ATtiny85 runs at 1MHz. To make it to run at 8MHz select Tools -> Burn Bootloader.

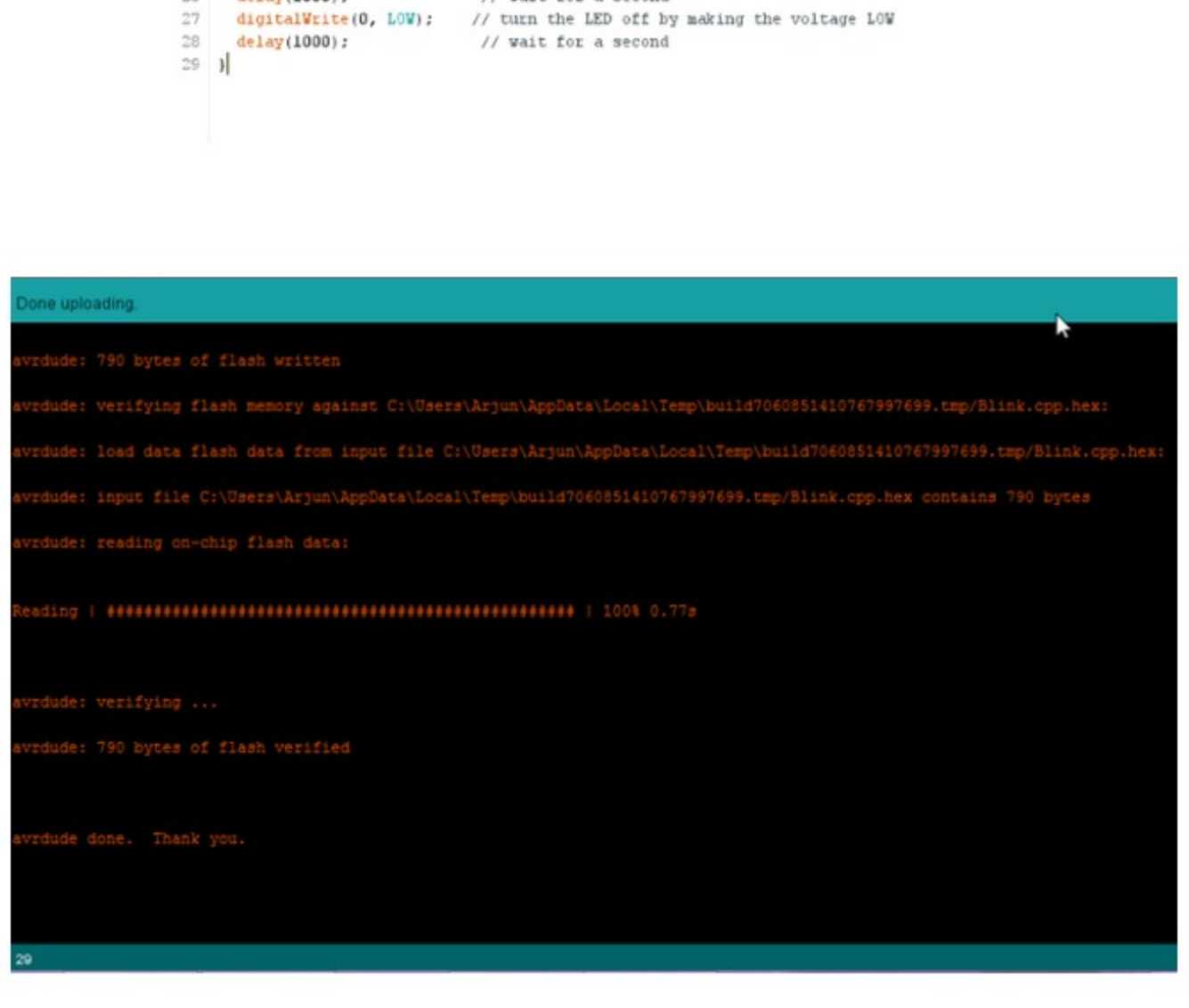




You will get the above message if burning bootloader was successful.

Now open the Blink example from arduino examples and change the pin number from 13 to 0 and upload.





You can see the above message if everything was successful. Now we have upload the blink program to ATtiny85 and now lets test it out.

Testing ATtiny85 Blink

Now its time to test. Remove all connections from Arduino and take a power source. Here I will use a button cell to power ATtiny85.