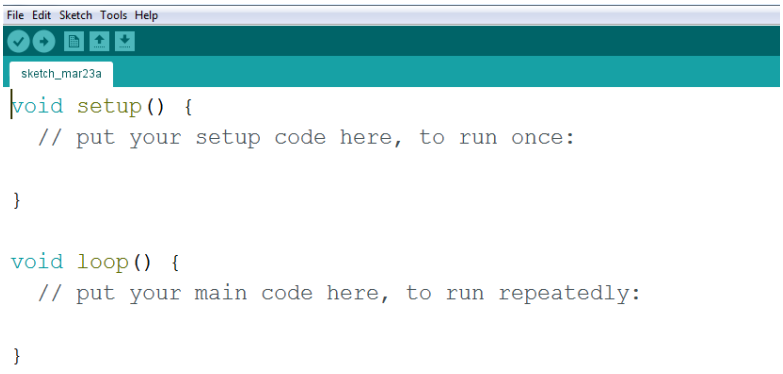
Programming Your Arduino

1. Connect your Arduino to a USB port. When you do this the Operating System will configure the drivers (software that allows you to communicate with the Arduino) and assign a COM port to the Arduino (e.g. COM7)

1. Start the Arduino IDE (Integrated Development Environment)

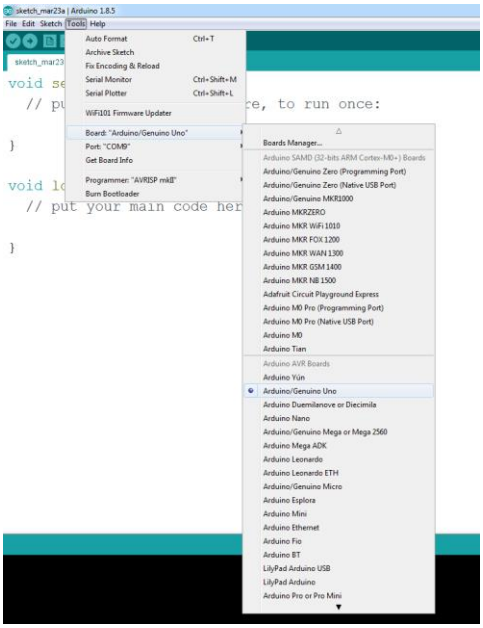


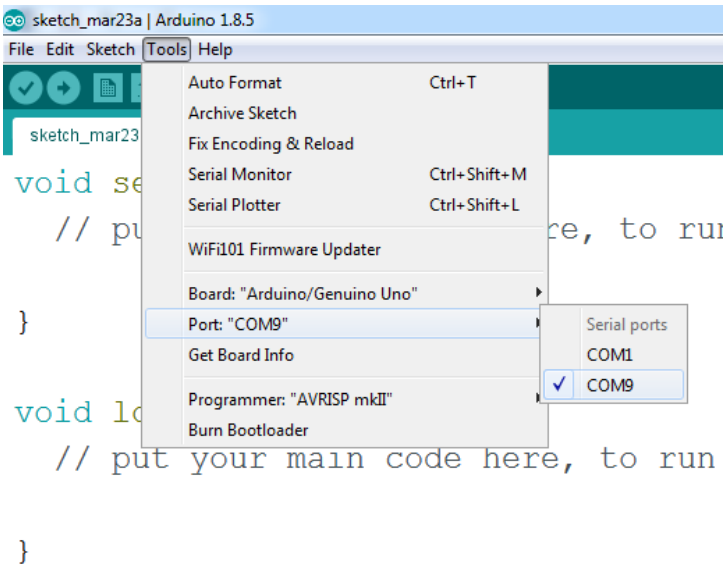
This will create a Sketch (an Arduino program) which contains 2 functions. The setup function will execute only once and that is where you put any initialization code. The loop function will continuously execute. As you get more advanced in you programming knowledge, you will learn to make your own functions.

2. You need to check two thing from the Tools menu:

    a. Board – to make sure you are configured to correct type of Arduino

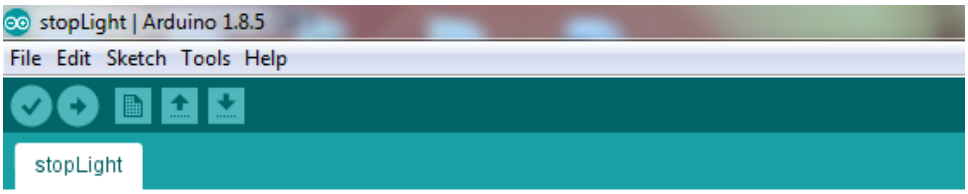
    b. Port – to make sure you are connected to the correct COM port.





Notice that my board type is Arduino/Genuino Uno and I am connected to COM9. If for some reason you have more than one COM port listed, choose the last on in the list.

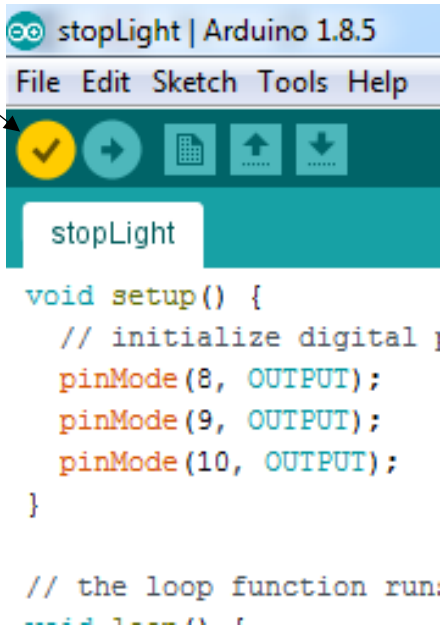
3. Write you program code and save it to a convenient location. I use a folder on my desktop called MyArduino. Be sure to give it a meaningful filename such as stoplight.




```
void setup() {
  // initialize digital pin 8, 9, and 10 as an output.
  pinMode(8, OUTPUT);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  //Green
  digitalWrite(10, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(6000);
  digitalWrite(10, LOW);    // turn the LED off (LOW is the voltage level)


  //Yellow
  digitalWrite(9, HIGH);    // turn the LED on (HIGH is the voltage level)
  delay(2000);
  digitalWrite(9, LOW);     // turn the LED off (LOW is the voltage level)

  //Red
  digitalWrite(8, HIGH);    // turn the LED on (HIGH is the voltage level)
  delay(6000);
  digitalWrite(8, LOW);     // turn the LED off (LOW is the voltage level)

}
```

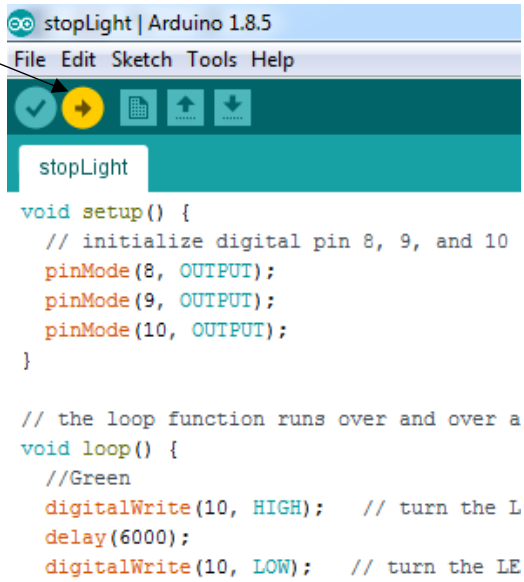
4. Compile your program. This will convert your source code (which you just wrote into 0's and 1's you processor can understand.

Compile Code



5. Upload you code to Arduino.

Upload Code



6. If your upload was successful, you should get the following message at the bottom of the IDE window.