

# Build Your First Robot with Arduino

UW Tacoma IEEE Student Branch

May 12, 2015

An Arduino is an open-source physical computing platform. That means its a way to program *things*. You can use an Arduino to control lights, motors, buttons, or almost anything. Arduinos provide a beginner friendly way to program microcontrollers. We'll be using the Arduino Uno, the most standard Arduino model.

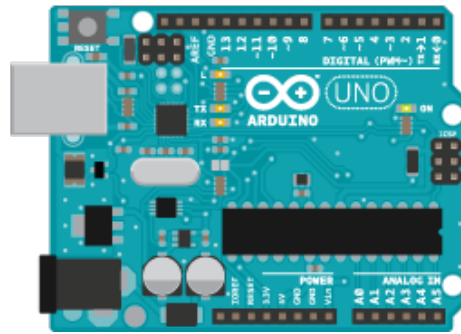


Figure 1: Arduino Uno

## 1 Getting Started

To program the Arduino, we'll use the Arduino IDE (integrated development environment). Open the start menu and type "arduino" to find it. It should look something like Figure 2. This is the application we'll use to write code and upload it to the Arduino board.

We'll start by blinking a light with our Arduino. Select **File** **Examples** **01.Basics** **Blink**. This will load the code for a program that blinks a light.

Next we put the code onto the board. Select **Tools** **Board** **Arduino Uno**. This tells the program which type of board we're using. Plug the board into your computer with a USB cable. Go to **Tools** **Port** and make sure something is checked. This tells the computer where to find your Arduino. Finally, select **File** **Upload**. A progress bar will show you as the code is sent to the board.

Congratulations! You have just programmed your Arduino. You should see the LED (light emitting diode) on your board blinking.

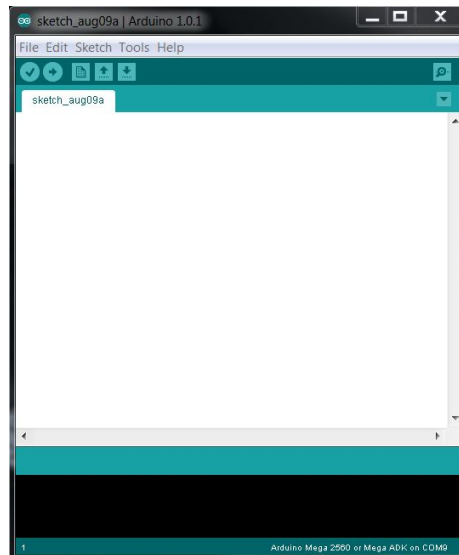


Figure 2: Arduino IDE

## 2 Building Your Robot

```

1 // This is a comment.
2
3 // The program will ignore anything we type after "//".
4
5 // We use comments to explain the code we write.
6
7 // You don't need to include the comments in your code for it to run;
8 // they're to help other people understand it.
9
10 #include <Servo.h> // This lets the program know we want to use servos.
11
12 // First we create variables to hold all the values
13 // that the program is going to need to run.
14
15 // For the motors:
16 Servo servo_right; // this makes a servo variable called servo_right.
17
18 Servo servo_left; // this makes another one called servo_left.
19
20 // For the speeds:
21 int spd_left; // an integer (whole number) to hold the speed of the left servo
22
23 int spd_right; // the right servo speed
24
25
26 // Next we tell the program what to do.
27 // Every Arduino program has a setup function and a loop function.
28 // The setup function runs once when the program starts.
29 // The loop function runs over and over forever.
30
31 // This will run only once at the program's beginning.
32 void setup()
33 {
34     servo_right.attach(10); // This tells the program which pins
35     servo_left.attach(11); // the servos are plugged into.
36 }
37
38 // This function will repeat forever.
39 void loop()
40 {
41     // The write function sets the speed of the servo.
42     // 0 is full speed in one direction
43     // and 180 is full speed in the opposite direction.
44     // 90 tells the servo to stand still.
45
46     servo_right.write(180); // Set the right motor to full speed.
47
48     servo_left.write(0); // The left motor is pointed the opposite way,
49                          // so we'll spin it the opposite direction.
50
51     delay(500); // This tells the program to wait 500 milliseconds (or 1/2 s).
52
53     servo_right.write(0); // Let's switch direction.
54     servo_left.write(180);
55
56     delay(500); // Wait another 1/2 second.
57
58     // The program will now jump back to the beginning of the loop function
59     // and repeat it until we turn the Arduino off.
60 }

```