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.

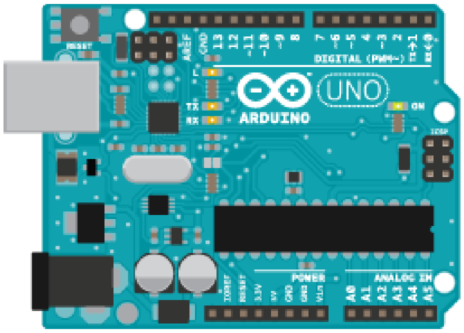


Figure 1: Arduino Uno

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.

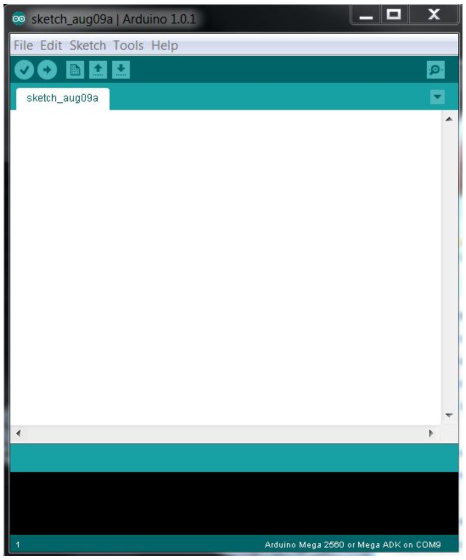

sketch_aug09a | Arduino 1.0.1
File Edit Sketch Tools Help
sketch_aug09a
1
Arduino Mega 2560 or Mega ADK on COM9

Figure 2: Arduino IDE

2 Building Your Robot

Before we program our robot, we have to build it. We'll use two servo motors to move. A servo is a precisely controlled motor (see Figure 3a). A servo has three wires to connect as shown in Figure 3b. The black and red wires are for power. Black is connected to *ground* which means zero volts. Red is connected to 5 volts. The white wire is for *signal*. This is what we use to tell the servo how to turn.

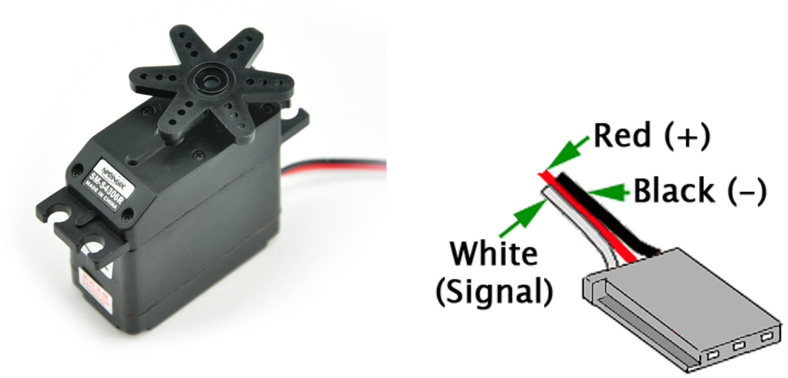
Servos use *pulse width modulation* (PWM). PWM is a special kind of signal from the Arduino. The Arduino can only make this signals on pins labeled with a '~'. We'll use pin 10 for the right servo and pin 11 for the left servo.

Figure 4 shows all the connections we need to make. Use jumper wires to make these connections:

1. Right servo ground (black) to Arduino ground (GND)
2. Left servo ground (black) to Arduino ground (GND)
3. Right servo signal (white) to Arduino pin 10
4. Left servo signal (white) to Arduino pin 11
5. Right servo high (red) to Arduino 4 volt (5V) to left servo high (red)

Since there is only one 5V pin on the Arduino, we'll use a Y-jumper cable with and extra connection point.

Once you have made all your connections, you're ready to start programming.



(a) servo motor

(b) servo connector

Figure 3

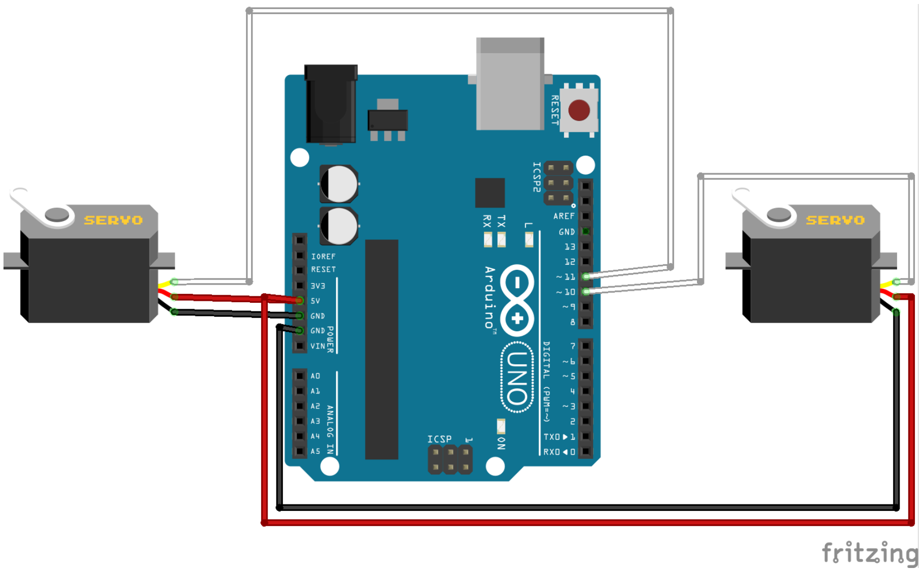


Figure 4: robot ciruit connections

```
// This is a comment.

// The program will ignore anything we type after "//".

// We use comments to explain the code we write.

// You don't need to include the comments in your code for it to run;
// they're to help other people understand it.

#include <Servo.h> // This lets the program know we want to use servos.

// First we create variables to hold all the values
// that the program is going to need to run.

// For the motors:
Servo servo_right; // this makes a servo variable called servo_right.

Servo servo_left;  // this makes another one called servo_left.

// For the speeds:
int spd_left;  // an integer (whole number) to hold the speed of the left servo

int spd_right; // the right servo speed


// Next we tell the program what to do.
// Every Arduino program has a setup function and a loop function.
// The setup function runs once when the program starts.
// The loop function runs over and over forever.

// This will run only once at the program's beginning.
void setup()
{
    servo_right.attach(10); // This tells the program which pins
    servo_left.attach(11);  // the servos are plugged into.
}

// This function will repeat forever.
void loop()
{
    // The write function sets the speed of the servo.
    // 0 is full speed in one direction
    // and 180 is full speed in the opposite direction.
    // 90 tells the servo to stand still.

    servo_right.write(180); // Set the right motor to full speed.

    servo_left.write(0);    // The left motor is pointed the opposite way,
                            // so we'll spin it the opposite direction.

    delay(500); // This tells the program to wait 500 milliseconds (or 1/2 s).

    servo_right.write(0);   // Let's switch direction.
    servo_left.write(180);

    delay(500); // Wait another 1/2 second.

    // The program will now jump back to the beginning of the loop function
    // and repeat it until we turn the Arduino off.
}
```