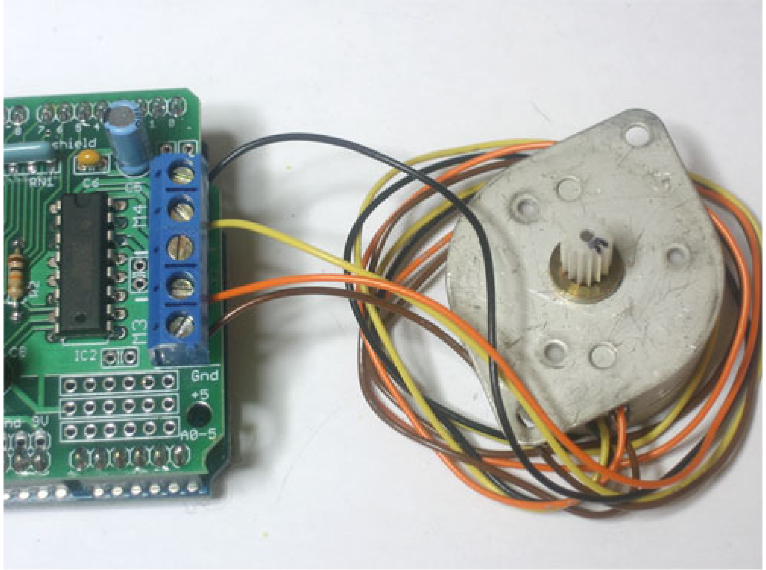
The Arduino Games Team

Stepper Motor



The Arduino Microcontroller

The Arduino board is a prototyping tool that makes interaction with machines much easier. The hardware consists of a small printed circuit board assembled with a microprocessor and various ports to allow for digital and analog inputs and outputs. Using these ports is as easy as plugging wire leads from other components into the Arduino board's various pins. More complex constructions require a separate prototyping board, such as a solderless breadboard. The main input is a USB port, where code is uploaded from a PC to be stored on the Arduino board. From there, the board can be used independently of a computer (with a separate power source) because it can store and process its own code to run sensors, motors, and other interactive components – in this case, a small stepper motor.

Motor Shield

The Arduino can only handle 5V but sometimes it is necessary to use more than 5V to power the various components of a design. In this case, a motor shield can be used. The motor shield can handle a higher voltage than the Arduino, in our case 11V. The motor shield is attached to the Arduino and an external power source that can be as high as 11V. The motor shield uses a printed circuit board to direct the proper amount of voltage to the Arduino and the other components, such as the motor in this example. Because our stepper motors for the Drawbot Competition require 5.4-6.4V in order to run, it is necessary for us to use a motor shield.

Stepper Motors

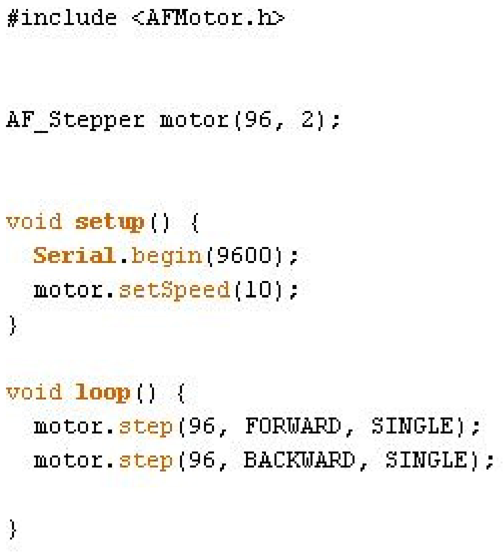
Stepper Motors are an electric DC motors that can rotate in discrete and equal steps. A stepper motor can only rotate by an integer multiple of its steps. Unlike servos, stepper motors do not need feedback sensors to rotate by only a certain degree. Stepper motors usually have 4,5,6,8 wires and they can be found in scanners, printers, and other electronics that requires a precise movement.

In our drawbot competition, our drawbot must be able to wind and unwind the wires connected to the pen by exact amaount and exact ratio to control the movement of the pen. While any ordinary motor can wind and unwind wires, it is diffficult to do the work by exact amounts. Therefore, we will be using stepper motors since stepper motors can rotate by exact amount as programmed.

Programming Code

Programming codes are necessary in order to effectively control motors, sensors, LED, and etc. Without code, the essential purpose of the Arduino is gone. With programming codes, there are infinite applications to any one device or combination of devices. All codes are basically input/ouput codes: collect data or send a signal to something, get feedback, output something else. There are variations, of course, like when to output, or what to output, how to output, but even the most complicated codes are broken down to just input and output.

The stepper motor code may seem intricate, but it is in fact quite simple. The single line that makes the coding much simpler is the #include<AFMotor.h> . Because of



```
#include <AFMotor.h>


AF_Stepper motor(96, 2);


void setup() {
  Serial.begin(9600);
  motor.setSpeed(10);
}

void loop() {
  motor.step(96, FORWARD, SINGLE);
  motor.step(96, BACKWARD, SINGLE);

}
```

Arduino’s open source library and fellow programmers who share their wisdom on the internet, amateurs like us have less work to do. The “.h” file contains all the motor definitions already defined, so turning the motor right is just in fact telling it to turn right, instead of incorporating a lot of state change functions and complicated variable assemblies. The second line of the code is to create the “motor” object with 96 representing the steps our stepper motor makes for a full revolution and the 2 is just the port it is connected to on the Arduino. The setup of the code is to just set consistent values throughout the entire run of the program. The serial begin command is to start the serial library and the setSpeed command is to give the stepper motor a constant rpm of 10. The “loop” of the code is to run this part of the code to repeatedly until a new code is uploaded or there is no power to the Arduino. The step commands (96, Forward, Single) is equivalent to saying (# of steps, direction (CCW or CW), the type of step). Step type is either single, double, interleave, or microstep. It represents how many steps you want to consider as one step. With the proper syntax and connections of the stepper motor to the Arduino and Arduino to computer, the code can be successfully uploaded and the stepper motor will turn 360 degrees one direction and then 360 degrees in the other direction.