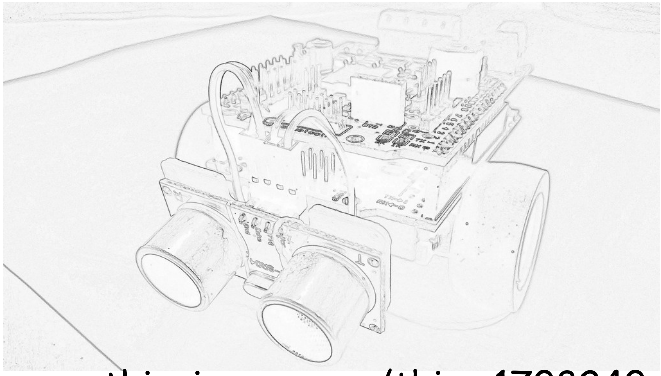

www.thingiverse.com/thing:1708240

Move Your Motors!

BabySCRU-FE Arduino Programming Lessons > rtheiss.com

Lesson One: Continuous Rotation Servos

Name:

Date:

School/Subject:

In this activity you will be introduced to the Arduino IDE and how to use it to move your Servo Motors.

1. Start Up Arduino: when you start up Arduino you will see:

```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

To put it pretty simply, if you put something in the setup code brackets {it will only run that program once}, and if you put something in between the loop brackets {it will run that program on a loop forever}.

2. Setup: Type each of these parts at the start of the code and in between the setup brackets

```
#include <Servo.h>

Servo Lservo;  // servo on baby SCRU-FE's left side (if you were looking in the same direction as the robot)
Servo Rservo;  // servo on baby SCRU-FE's right side (if you were looking in the same direction as the robot)

void setup()
{   // put your setup code here, to run once:
  Lservo.attach(14);
  Lservo.write(90);  // set servo to Stop

  Rservo.attach(16);
  Rservo.write(90);  // set servo to Stop
}
```

In this setup you will notice a few things:

- Adding two forward slashes before anything makes it a note and not part of the program.
- All lines of programs end with a semi-colon.
- You can include other libraries such as servo libraries so you don't have to reinvent the wheel.
- You can define what Servos there are and give them a cool nickname like “Left” or “Right.”
- You can attach the servo sensor wires to a specific pin on the Arduino Board and Identify where it was connected in the code
- What you “write” to the servo, is the electrical signal that tells the servo what to do

3. Compile, Upload and Test Servos! Well this code if it works well will do nothing, so don't get too excited, this is just an exercise to see if your servos need adjusting and that you can send a signal to your board.

- Connect your Arduino Board via USB to your Computer.
- Wait for about 15 seconds then look under Tools > Ports in the Arduino IDE Program and specify which port the Board is connected to.
- If no ports show up, and the Board is getting USB power, you may have to search for how you can fix arduino driver issues at www.arduino.cc.
- If you can connect, then click the check mark to Verify or Compile your code
- If the code cannot be compiled, check the spelling and order of the words and symbols you have typed into the code. This is called Syntax, your microprocessor has to break this text down into ones and zeroes (binary code), so if a punctuation point is in the wrong place the whole program might act completely different or make no sense to the microchip/board
- UPLOAD! If all is well hit the upload button, the Arduino Board Should Blink, and you should get a message that the program was successfully loaded to the board.

Don't worry if the motors don't spin at this point, if all is set up right they will not. If they do at this point, then take them off of the chassis and adjust the little screw inside the bottom of the servo until they do not spin. This means the servos are calibrated correctly to the signals that are given to them by the board.

4. Get your motors Moving! Add this loop code below your setup code, then experiment by changing the values from 0 to 180 and the delays (which are in microseconds) to see what your robot can do.

```
void loop()
{   // put your main code here, to run repeatedly:

 //Forward
 Lservo.write(180);  // set servo speed 180 = Fast Forward // 0 = Fast backwards // 90 = Stopped
 Rservo.write(0);  // set servo speed 0 = Fast Forward // 180 = Fast backwards // 90 = Stopped
 delay(500);

 //Backward
 Lservo.write(0);  // set servo speed 180 = Fast Forward // 0 = Fast backwards // 90 = Stopped
 Rservo.write(180);  // set servo speed 0 = Fast Forward // 180 = Fast backwards // 90 = Stopped
 delay(500);

 //Turn on a dime
 Lservo.write(180);  // set servo speed 180 = Fast Forward // 0 = Fast backwards // 90 = Stopped
 Rservo.write(180);  // set servo speed 0 = Fast Forward // 180 = Fast backwards // 90 = Stopped
 delay(500);
}
```