

Advanced Manufacturing Academy

ROBOTICS – 4
More Outputs

College of Engineering and Technology

East Carolina University

Improved Output Controls

- Tried Simple digital controls
- Multiple devices
- Controlled speed of lights
- Did anybody get the light to DIM?
 - How would you control the brightness?
 - Pulse Width Modulation (PWM)
- Introduce PWM
- The Servo

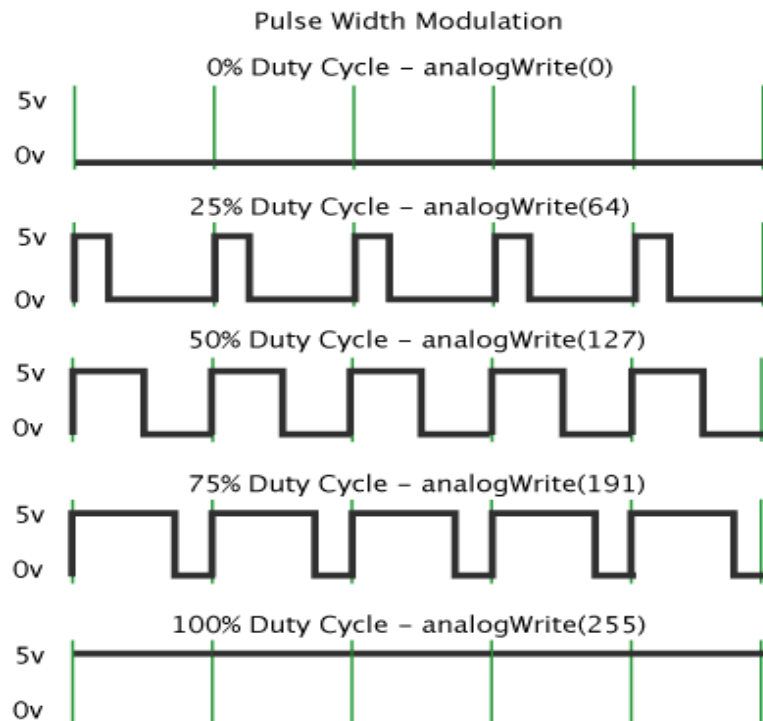
What is PWM?

- Pulse Width Modulation
- Digital way to vary power to device

Digital control is used to create a square wave, a signal switched between on and off. This on-off pattern can simulate voltages in between full on (5 Volts) and off (0 Volts) by changing the portion of the time the signal spends on versus the time that the signal spends off. The duration of "on time" is called the pulse width. To get varying analog values, you change, or modulate, that pulse width. If you repeat this on-off pattern fast enough with an LED for example, the result is as if the signal is a steady voltage between 0 and 5v controlling the brightness of the LED.

What is PWM?

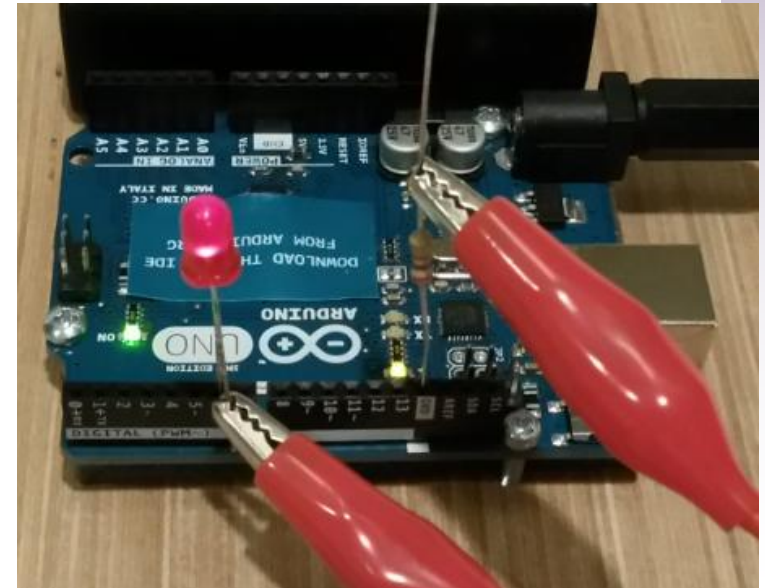
In the graphic below, the green lines represent a regular time period. This duration or period is the inverse of the PWM frequency. In other words, with Arduino's PWM frequency at about 500Hz, the green lines would measure 2 milliseconds each. A call to `analogWrite()` is on a scale of 0 - 255, such that `analogWrite(255)` requests a 100% duty cycle (always on), and `analogWrite(127)` is a 50% duty cycle (on half the time) for example.



<http://www.arduino.cc/en/Tutorial/PWM>

Dimmable LED

- You will need:
 - Robot, Laptop, External LED/Resistor
- Is there any way to vary the power to the LED?
 - Change/remove resistor
 - Change the blinks?
- Install LED in pin 6
- Install resistor in GND



Dimmable LED - Code

- Load code from desktop –
 - Test_2_camp_061515

```
Test_2_camp_061515$  
// This program increases the intensity of light from a LED light over several steps  
  
int ledPin = 6;      // LED connected to digital pin 9  
  
void setup()  
{  
  pinMode(ledPin, OUTPUT); // sets the pin as output  
}  
  
void loop()  
{  
  analogWrite(ledPin, 0); // analogWrite values from 0 to 255  
  delay(1500); // Wait 1.5 seconds  
  analogWrite(ledPin, 50); // analogWrite values from 0 to 255  
  delay(1500); // Wait 1.5 seconds  
  analogWrite(ledPin, 100); // analogWrite values from 0 to 255  
  delay(1500); // Wait 1.5 seconds  
  analogWrite(ledPin, 150); // analogWrite values from 0 to 255  
  delay(1500); // Wait 1.5 seconds  
  analogWrite(ledPin, 200); // analogWrite values from 0 to 255  
  delay(1500); // Wait 1.5 seconds  
  analogWrite(ledPin, 255); // analogWrite values from 0 to 255  
  delay(1500); // Wait 1.5 seconds  
}
```

analogWrite

- Used to Send PWM signal to output
 - Only on designated pins 3,5,6,9,10,11
 - 0 – 255: 0 = off, 255 = max
- SYNTAX
 - `analogWrite(pin #,value)`
 - Don't have to declare pin an output to use this comand...

Dimmable LED - Code

- Load code from desktop –
 - Test_3_camp_061515

```
Test_3_camp_061515
// This program increases the intensity of light from a LED smoothly
// using a for loop
int ledPin = 6;      // LED connected to digital pin 6

void setup()
{
  pinMode(ledPin, OUTPUT);  // sets the pin as output
}

void loop()
{
  // use for loop to smoothly increase value of i in the analogWrite statement
  for (int i=0; i <= 255; i++){
    analogWrite(ledPin, i);
    delay(15);
  }
}
```


For Loop

- The For loop
- Repeat a command set over and over till a specific parameter is met.

- SYNTAX

➤ for (initialization; condition; increment) {
 //statement(s);
}

The diagram illustrates the syntax of a for loop with the example code: `for(int x = 0; x < 100; x++){ println(x); // prints 0 to 99 }`. Annotations include: 'parenthesis' pointing to the entire for loop structure; 'declare variable (optional)' pointing to `int x = 0`; 'initialize' pointing to `x = 0`; 'test' pointing to `x < 100`; 'increment or decrement' pointing to `x++`; and a large curved arrow pointing from the increment part back to the test part, indicating the loop's repetition.

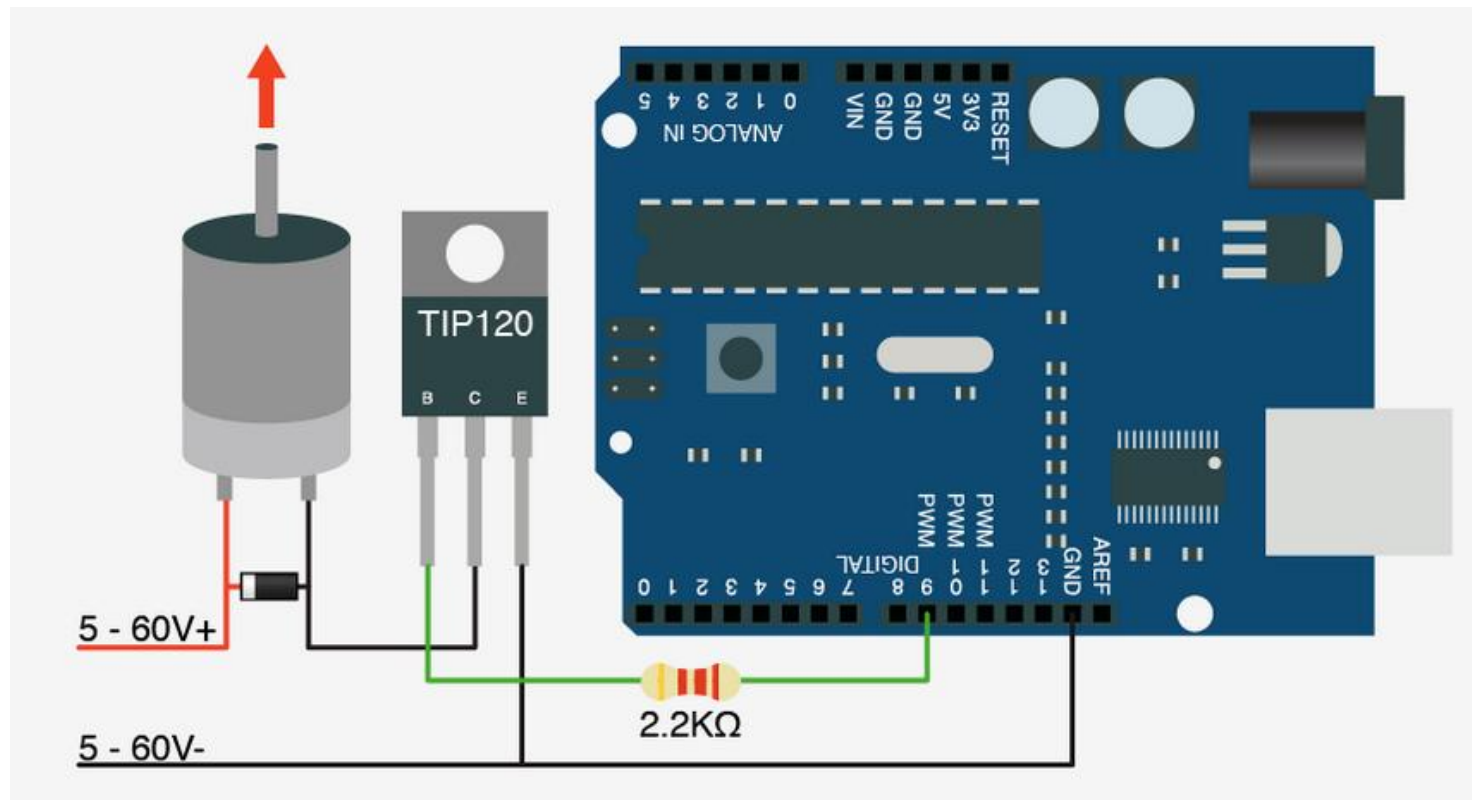
```
for(int x = 0; x < 100; x++){  
    println(x); // prints 0 to 99  
}
```

Motor Control

- We can vary output on a PWM Pin
- Now we can control motor speed, right?
 - Nope
 - Power problem - $.040A \times 5v = .2$ watts max
 - OK for small device like LED but not good for big device like motor
- Use the small signal from the Arduino to adjust the power from a larger source
 - Motor Controller Circuit between the Arduino and the motor

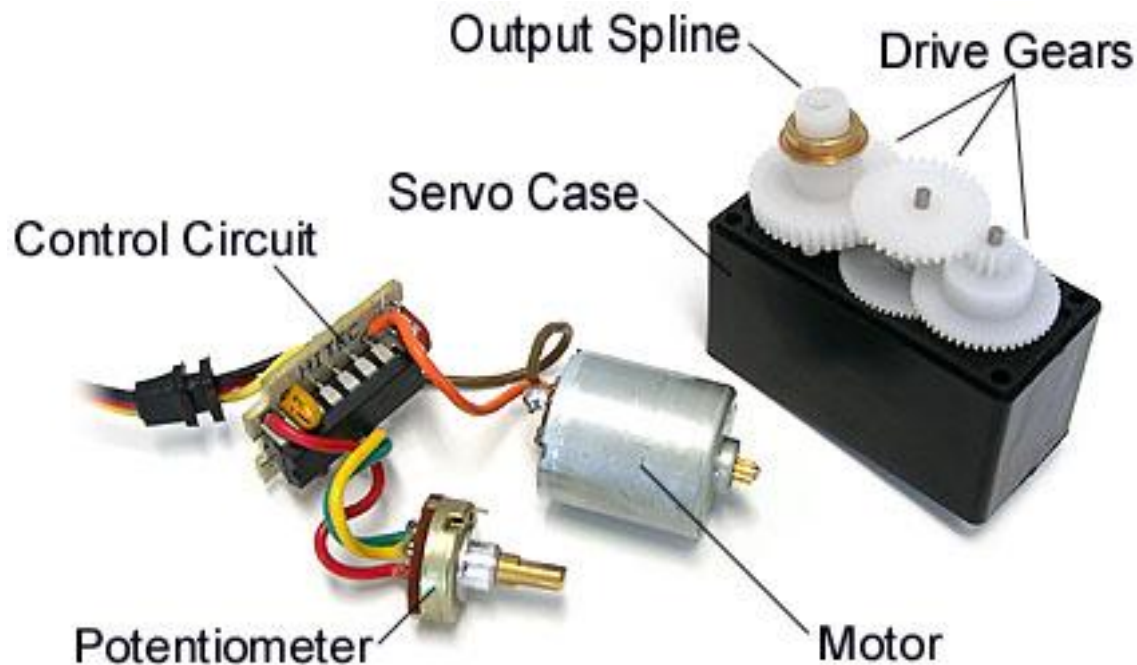
Motor Control - Example

- Use a Transistor
 - Small current from Arduino controls large current to motor



Combine Parts - SERVO

- Servo combines all parts in one.
 - Motor, gearbox, controls and sensor



https://www.servocity.com/html/how_do_servos_work_.html#.VYEwVvIViko

Servo – Feedback Control

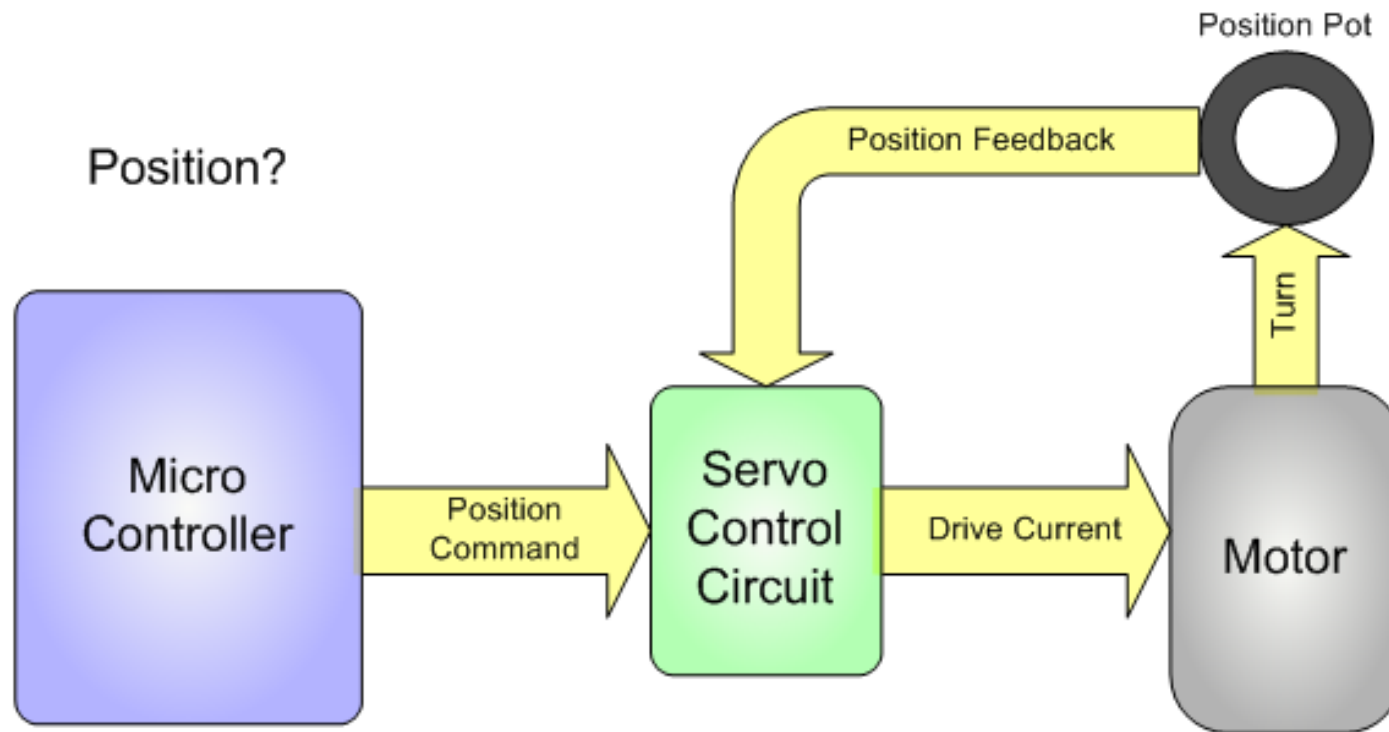
- Typical servo used for positioning
- Servo moves to desired position and holds that position
- Internal feedback loop
 - Potentiometer in servo measures position
 - Control circuit compares input request to signal from the potentiometer
 - If servo is not in correct position, it will move till the input and output signal match



<http://gentles.ltd.uk/control/index.htm>

Servo Control Loop

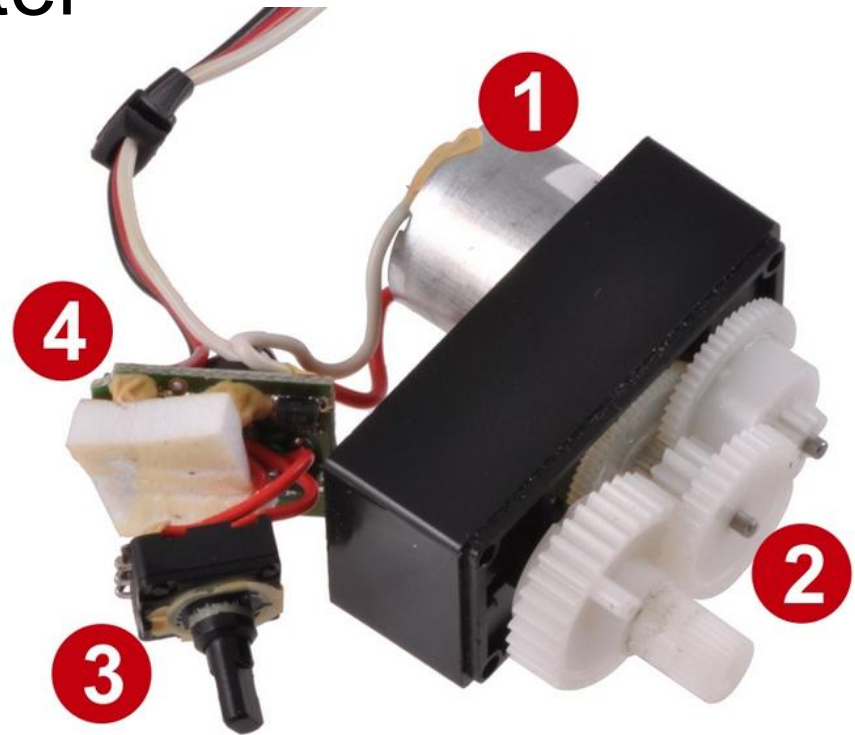
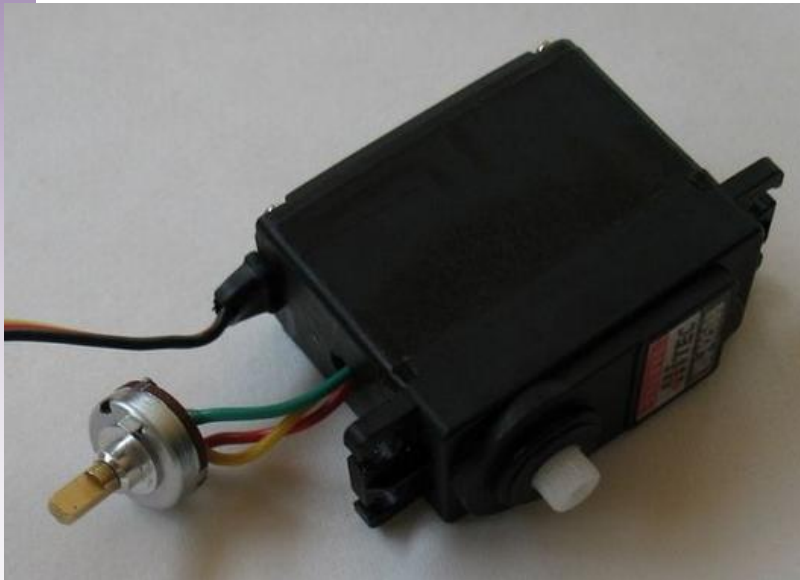
- Normal Servo looks like this.



<https://learn.adafruit.com/analog-feedback-servos/about-servos-and-feedback>

Continuous rotation - Difference

- Break connection at position sensor.
- Modify Potentiometer



<http://www.instructables.com/id/Low-Cost-Hobby-Servo-XY-Table/step6/Modify-the-Servos-for-Continuous-Rotation-and-Remo/>

<http://www.pololu.com>

Servo Control

- Let's try out a “normal” servo
- You will need:
 - Robot
 - Main harness back in place
 - Normal servo from parts
- Disconnect the right wheel servo
- Connect the test servo on the “9” plug
- Install a servo adapter



www.pololu.com

Sub-micro servo 3.7 g on a US quarter for size reference.

Normal Servo

- Normal servo behavior
- Example:
- You will need:
 - Robot
 - Main harness back in place
 - Normal servo from parts
- Disconnect the right wheel servo
- Connect the test servo on the “9” plug
- Install a servo adapter on the servo



www.pololu.com

Sub-micro servo 3.7 g on a US quarter for size reference.

Test the Servo

- Open test_4_camp_061515
 - In desktop folder

```
Test_4_camp_061515
```

```
// Controlling a servo position using set values
// modified code from Michal Rinott <http://people.interaction-ivrea.it/m.rinott>
// modified by William McClung

#include <Servo.h>

Servo myservo; // create servo object to control a servo

int val; // variable 0 to read the value for the servo
int val1; // variable 1 to read the value for the servo
int val2; // variable 2 to read the value for the servo
int val3; // variable 3 to read the value for the servo

void setup()
```

Test the Servo

- Open: test_4_camp_061515
- Load and test

```
void setup()
{
  myservo.attach(9); // attaches the servo on pin 9 to the servo object
  val = 20;
  val1 = 60;
  val2 = 120;
  val3 = 170;

  myservo.write(val); // sets the servo position according to the value
  delay(1500); // waits for the servo to get there
  myservo.write(val1); // sets the servo position according to the value
  delay(1500); // waits for the servo to get there
  myservo.write(val2); // sets the servo position according to the value
  delay(1500); // waits for the servo to get there
  myservo.write(val3); // sets the servo position according to the value
  delay(1500); // waits for the servo to get there
}
```

Servo Library – The Rules

- Important notes:
 - Using “Servo” library
 - Must load library at beginning
 - Special command sets
 - Takes the place of analogWrite
- RULES for Servo library
 - Must call library at start
 - Create a servo “object”
 - Must attach it to a pin
 - ✓ Must be a PWM Pin

Now Continuous Rotation

- What happens if you:
 - Connect right wheel servo and run Test4?
 - Run away servos... Why?
 - How to stop the servo?
- If 0 is full left and 180 is full right
 - What is the middle?
 - Try setting the servo for 90 on last line
 - What happens?
- How do we use this?

Load New Code for Servos

- Let's Load a new code to control servos
- Open file: Test_5_camp_061515

```
Test_5_camp_061515
// Code for camp robots 061515 - William McClung

#include <Servo.h> // get servo library for our robot

Servo rmyservo; // create a right servo
Servo lmyservo; // create a left servo

void setup()
{
  rmyservo.attach(9); // attach right servo to pin 9
  lmyservo.attach(10); // attach left servo to pin 10
  rmyservo.write(90); // set right servo to mid-point (stop)
  lmyservo.write(90); // set left servo to mid-point (stop)
  delay(1000); // wait one second
}
```

New Code for Servos

Continue Review

```
rmyservo.write(45); // set right servo to drive forward 1/2 speed
lmyservo.write(135); // set left servo to drive forward 1/2 speed
delay(2000); // wait 2 seconds

rmyservo.write(90); // set right servo to mid-point
lmyservo.write(90); // set left servo to mid-point
delay(1000); // wait 1 second

rmyservo.write(135); // set right servo to drive backward 1/2 speed
lmyservo.write(45); // set left servo to drive backward 1/2 speed
delay(2000); // wait 2 seconds

rmyservo.write(90); // set right servo to mid-point
lmyservo.write(90); // set left servo to mid-point
delay(1000); // wait 1 second
|
rmyservo.write(135); // set right servo to drive backwards 1/2 speed
lmyservo.write(135); // set left servo to drive forwards 1/2 speed
delay(2000); // wait 2 seconds

rmyservo.write(90); // set right servo to mid-point
lmyservo.write(90); // set left servo to mid point
delay(1000); // wait 1 second

rmyservo.write(45); // set right servo to drive forward 1/2 speed
lmyservo.write(45); // set left servo to drive backwards 1/2 speed
delay(2000); // wait 2 seconds

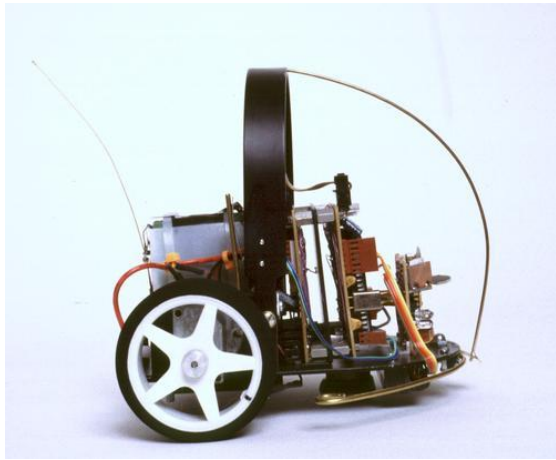
rmyservo.write(90); // set right servo to mid-point
lmyservo.write(90); // set left servo to mid-point
```

Test_5 Code Questions?

- Help me figure out these questions
 - What is full speed forward?
 - What is full speed backward?
 - Why are the speeds different for the left and right servo when going forward or backward?
 - How do you make a right turn?
 - How do you make a left turn?

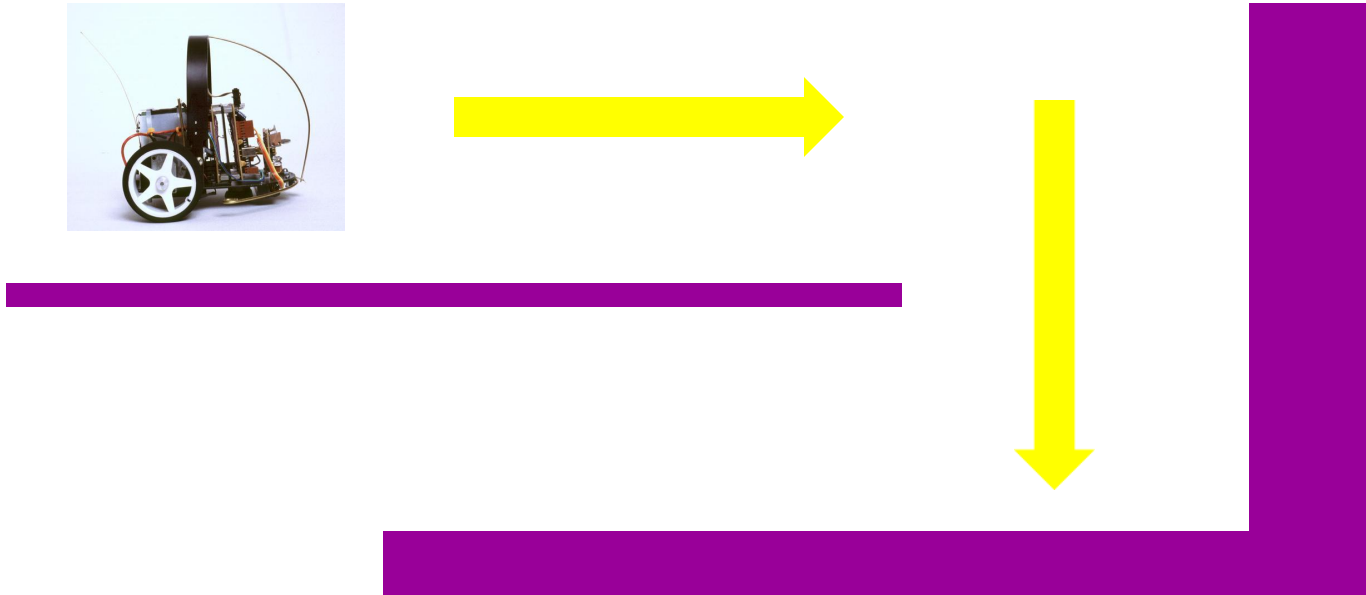
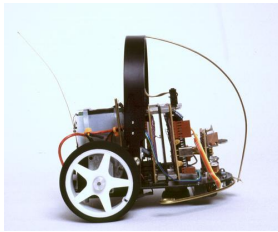
Challenge 1

- Challenge 1
 - Start 3 feet from the wall
 - Stop as close as possible without touching



Challenge 2

- Follow along divider
- Stop without hitting wall
- Turn
- Move close to wall without touching



Challenge 3

- Follow along divider
- Stop without hitting wall
- Turn
- Move close to wall without touching
- **Return to where you started!**

