Julia Frothingham

Lab 19 - Servo motors, ultrasonic sensors

What does “duty cycle” mean when you are using the PWM pins?

The duty cycle refers to the percent of one cycle supplied by the PWM pin in which the output supplied is HIGH rather than LOW. So a 50% duty cycle would be HIGH for 50% of the time and LOW for 50% of the time.

When you use a device like the servo motor, you have to have at least 3 (but possibly more) different lines of code, usually in 3 different places:

`#include <Servo.h>`

`Servo myServo; `

`myServo.write(XX)  // or myServo.attach(X)`

Roughly, what are each of these three lines doing?

The first line is importing the Servo library header file. This contains definitions for objects and methods

used in the library to control servo motors. The second line creates a new servo motor object called `myServo.` The first statement on the third line sends a signal to the pin connected to the servo motor associated with `myServo.` The second statement on the third line specifies which pin the servo motor associated with `myServo` is connected to.

The if statement has the form:

`if (logic statement) {  …some line(s) of code…  } `

In your own words, what is a logic statement?

A logic statement is a condition that must be true for the lines of code inside the curly brackets to run. It is a check whose output is either true or false.

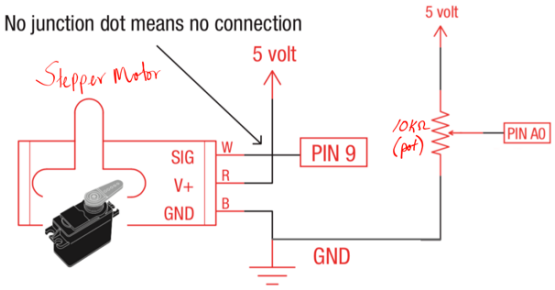
How is “Else if” different from “Else”?

`else if` is different from `else` because it is not a guarantee that the code inside the scope of the `else if` statement will run. An `else` statement will always run if the preceding `if` statement is false. An `else if` statement will only run if the preceding `if` statement is false AND the logic statement inside the `else if` statement is true.

What’s the difference between ‘float’ and ’int’ type variables? What advantages are there to using ‘int’ whenever possible? When do you have to use ‘float’?

A float is a floating point number. It is a decimal number. An int is purely an integer. The advantage to using an int whenever possible is that it takes up less computer memory. Integers are easier to represent in binary, and so you can store much higher-value integers than floats, and waste less memory. You must use float if you want to represent more exact numbers with decimal places.

Figure 1. Project 3 Circuit 3A diagram and code.



Circuit_3A_Servo

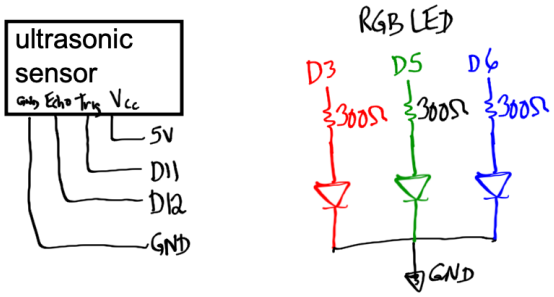
```
/*
 * Circuit 3A-Servo
 * Move a servo attached to pin 9 so that its angle matches a potentiometer attached to A0.
 */

#include <Servo.h>   //import servo control library
Servo myservo;       //create a servo object called myservo

// variables to store potentiometer and servo postions
int potPosition;
int servoPosition;

// assign pin connections to variables - minimal code changes if wiring changed
int motorPin = 9;
int potPin = A0;

void setup() {
  // specify that servo myservo is connected to motorPin
  myservo.attach(motorPin);
}

void loop() {
  // read the position of the potentiometer
  potPosition = analogRead(potPin);

  // map the read potentiometer position from a range of 0-1023 to a range of 20-160
  // store the converted value in servoPosition
  servoPosition = map(potPosition, 0, 1023, 20, 160);

  // move the servo to the specified position in degrees
  myservo.write(servoPosition);
}
```

The circuit contains a servo motor, which moves to different positions to mirror the position of a potentiometer. A video of the circuit in action can be viewed here: https://youtu.be/weWXhpJsry8

Figure 2. Project 3 Circuit 3B diagram.

Circuit_3B_DistanceSensor

```
/*
 * Circuit 3B-Distance Sensor
 * Control the color of an RGB LED using an ultrasonic distance sensor
 */

// constant-value integers to store pin connections for sensor and LED legs
const int trigPin = 11;
const int echoPin = 12;
const int redPin = 3;
const int greenPin = 5;
const int bluePin = 6;
// float variable to store distance value
float distance = 0;

void setup() {
  // begin serial communication at 9600 baud
  Serial.begin(9600);
  // set sensor trigger pin as output and sensor echo pin as input
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  // set LED pins as output
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
}

void loop() {
  // measure distance between sensor and any detected object
  distance = getDistance();
  // print distance in inches
  Serial.print(distance);
  Serial.println(" in");

  // if object detected less than 10 inches away, set LED to red
  if(distance <= 10){
    analogWrite(redPin, 255);
    analogWrite(greenPin, 0);
    analogWrite(bluePin, 0);


  // if object detected 10-20 inches away, set LED to yellow-orange
  } else if(10 < distance && distance < 20){
    analogWrite(redPin, 255);
    analogWrite(greenPin, 50);
    analogWrite(bluePin, 0);

  // if no object detected within 20 inches, set LED to green
  } else{
    analogWrite(redPin, 0);
    analogWrite(greenPin, 255);
    analogWrite(bluePin, 0);
  }
  delay(50);  // wait 50ms before checking again - gives time to actually run code
}

//------------FUNCTIONS-------------

float getDistance(){
 // variables to store measured time between signal echos and calculated distance values
  float echoTime;
  float calculatedDistance;

  // output a pulse signal on trigger pin
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // measure time between pulses on echo pin
  echoTime = pulseIn(echoPin, HIGH);

  // convert from time to distance in inches
  calculatedDistance = echoTime/148.0;

  return calculatedDistance;
}
```



The circuit changes the color of an LED based on whether it can detect objects within a certain range. A video of the circuit in action can be viewed here: https://youtu.be/LcbmDld85Is

The circuit can be modified by adding: 1) a buzzer between digital pin 3 and ground and 2) a stepper motor connected to digital pin 9, 5V power, and ground

When nearby objects are detected, the circuit can respond with light, sound, and motion.

Figure 3. Project 3 Circuit 3C code.

Circuit_3C_MotionAlarm

```
/*
 * Circuit 3C-Motion Alarm
 * Control the color of an RGB LED using an ultrasonic distance sensor.
 * When an object is close to the sensor, buzz the buzzer and wiggle the servo motor.
 */

#include <Servo.h>  //import servo control library

// assign pin connections to variables - minimal code changes if wiring changed
const int trigPin = 11;
const int echoPin = 12;
const int redPin = 3;
const int greenPin = 5;
const int bluePin = 6;
const int buzzerPin = 10;
float distance = 0;

Servo myservo;  //create a servo object called myservo

void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);
  pinMode(buzzerPin, OUTPUT);

  myservo.attach(9);   // specify that servo myservo is connected to motorPin
}

void loop() {
  // measure distance between sensor and any detected object
  distance = getDistance();
  // print distance in inches
  Serial.print(distance);
  Serial.println(" in");

  // if object detected less than 10 inches away
  // set LED to red, move servo back and forth, and play a repeated tone
   if(distance <= 10){
    analogWrite(redPin, 255);
    analogWrite(greenPin, 0);
    analogWrite(bluePin, 0);

    //play a tone and move the servo to 10 degrees
    tone(buzzerPin, 272);
    myservo.write(10);
    delay(100);   // pause for 100ms
    //stop the tone and move the servo to 150 degrees
    noTone(buzzerPin);
    myservo.write(150);
    delay(100);   // pause for 100ms

  // if object detected 10-20 inches away, set LED to yellow-orange
  } else if(10 < distance && distance < 20){
    analogWrite(redPin, 255);
    analogWrite(greenPin, 50);
    analogWrite(bluePin, 0);
  // if no object detected within 20 inches, set LED to green
  } else{
    analogWrite(redPin, 0);
    analogWrite(greenPin, 255);
    analogWrite(bluePin, 0);
  }
  delay(50); // wait 50ms before checking again - gives time to actually run code
}
```

```
//-----------FUNCTIONS-------------

float getDistance(){
  // variables to store measured time between signal echos and calculated distance values
  float echoTime;
  float calculatedDistance;

  // output a pulse signal on trigger pin
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // measure time between pulses on echo pin
  echoTime = pulseIn(echoPin, HIGH);
  // convert from time to distance in inches
  calculatedDistance = echoTime/148.0;

  return calculatedDistance;
}
```

The circuit changes the color of an LED based on whether it can detect objects within a certain range. When it detects nearby objects, it also plays sound and moves a “guardian” dinosaur figurine. A video of the circuit in action can be viewed here: https://youtu.be/7j8s_jNFtPw