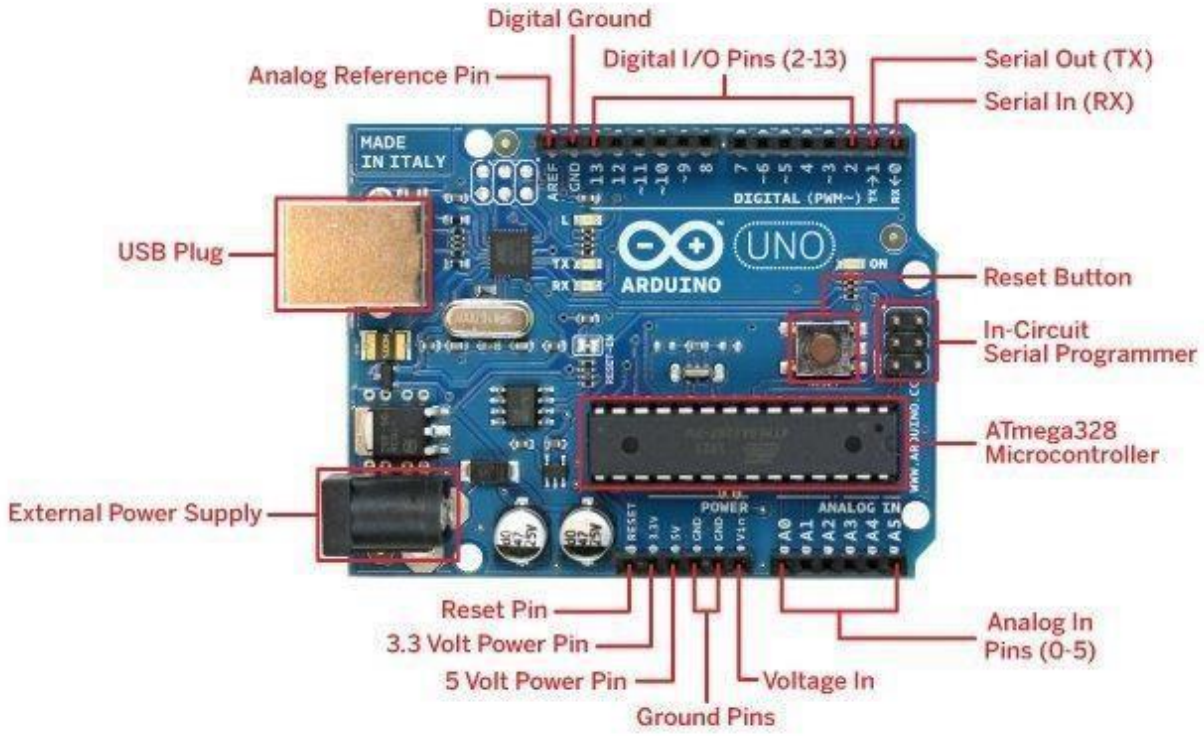
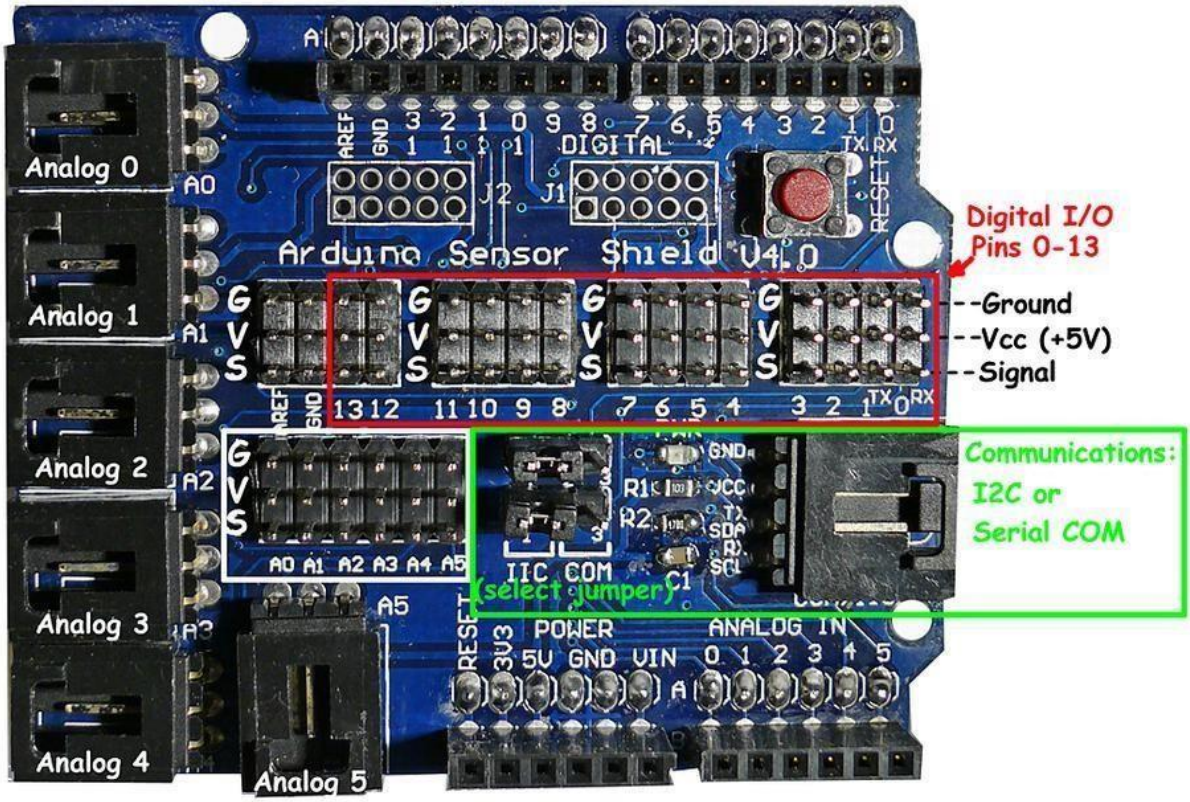
Arduino and Servo Motor

By: Matthew Jourden
Brighton High School

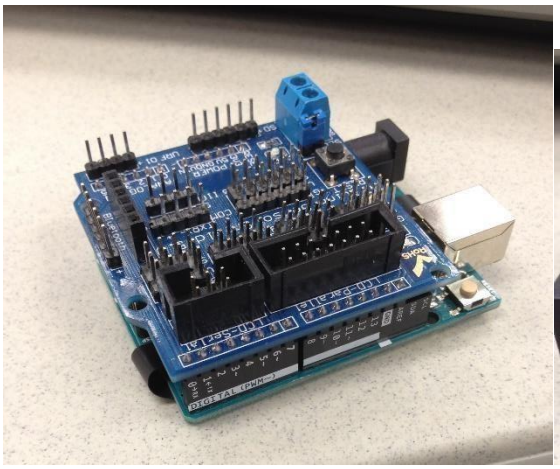
1. Basics of the Arduino Board and Arduino
   a. Arduino is a mini computer that can input and output data using the digital and analog pins

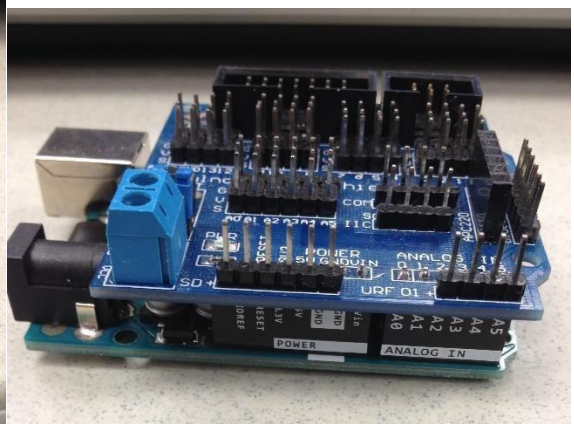


b. Arduino Shield: mounts on top of Arduino Uno board lining up the pins. The use of the shield is to expand the flexibility of the Arduino Uno board.



2. Link the Arduino Shield on top of the Arduino Board linking the proper pins to each port





Arduino and Servo Motor Tutorial
By: Matthew Jourden
Brighton High School

3. Motor Control Syntax
    a. include <Servo.h> : Library for a the servo motors to send a digital signal to the motor to turn on or off
    b. Variable Type Servo : variable that allows the programmer to set a variable name for the motor(s) that are being used
    c. "servo name".attach ("servo pin location") : Allows user to attach the variable that is declared as a Servo Type to a specific pin location.

For Example

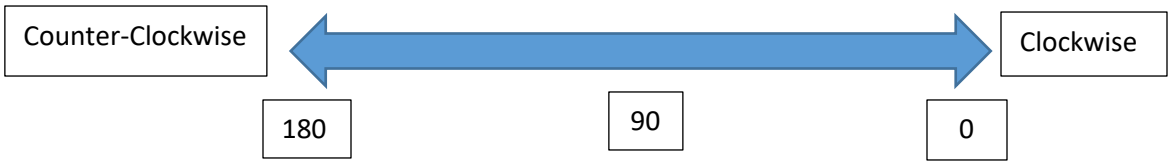
```
#include <Servo.h);          //Sets Servo Library

int Servopin = 13;           //Sets Servo Digital Pin 13
Servo servo;                 //Declares Variable Name servo
void Setup()
{
  servo.attach (Servopin);   //Attaches Variable servo to Pin 13
}
```

    d. "servo name". write ("value 0-180) : writes the rotational direction for the motor

Motor Movement: Servo motor movements are rated 0 to 180 degrees

a 0 = Full Speed Clockwise Rotation
b 90 = Theoretical Neutral position
c 180 = Full Speed Counter-Clockwise Rotation



e. "servo name".detach (); : Removes power from the Variable Servo

4. Servo Wires

Wires: Use Servo Motor Wire to Female End to connect to the board
Wire the Motor to the Arduino Shield in Pin Row 4 Colors
Brown or Black= Ground (G)
Red or Red= Voltage (V)
Orange or White= Signal (S)

5. Ultrasonic Sensor
   Wires: Use Four (4) Female to Female wire connectors
   Wire the Following
   Follow the proper color set

Ultrasonic Sensor	Arduino Board	Pin Row
Ground (GND)	G	9
Volt (VCC)	V	9
Echo	S	9
Trig	S	10

NOTE: Voltage and Ground Wires must be in the same digital port row in order to complete the circuit with the Arduino board and the Ultrasonic Sensor circuit board. The Trig and Echo Pins can be located anywhere in the Signal Column (I.E Echo in port 9 and Trig in port 10).

Arduino and Servo Motor Tutorial
By: Matthew Jourden
Brighton High School

Program: Servo Motor Sweeping Angles Clockwise versus Counter-wise Spin

```
int servoPin = 4;                //Links Digital Port 4 to Servo Motor
Servo myservo;                   //Creates a variable time of Servo that represents the phyiscal motor

void loop() {

  myservo.attach (servoPin);     //Turns on the flow of electricity from the physical motor (myservo)
                                 //via Digital Port 4 (servoPin) on the Arduino Board

  //Scans From 0 to 90 Degrees (Clockwise Motion)varying the speed each time thru the loop
  for (int angle = 0; angle <= 90; angle++)
  {
    myservo.write (angle);       //Sets the direction and speed for the motor clockwise
    //Note: each time thru the loop the angle will increase by 1 and get slower as it approaches 90
    delay (100);
  }
  myservo.detach ();             //Disconnects power to the physical motor
  //Note: program already knows what port to disconnect from via the myservo.attach
  delay (1500);
  myservo.attach (servoPin);     //Turns power on to the motor

  for (int angle = 180; angle > 90; angle --)
  {
    myservo.write (angle);       //Sets the direction and speed for the motor counterclockwise
    //Note: each time thru the loop the angle will increase by 1 and get slower as it approaches 90
    delay (100);
  }
  myservo.detach ();            //Disconnects power to the physical motor
  //Note: program already knows what port to disconnect from via the myservo.attach
  delay (1500);
}
```

Modify the code to test speed, direction and duration

Arduino and Servo Motor Tutorial
By: Matthew Jourden
Brighton High School

Program : Motor with Ultrasonic: Upload and test the program using the single motor

Ultrasonic Sensor is designed to send out a sound wave signal called the Trigger; and receive the bounced back sound wave into the Echo port. The sound wave will pulsate the Trigger on and off so the sound wave returning from the contacted object will be able to pass between the pulses. If the Trigger was constantly on the returning sound wave would be distorted.

Trigger Sound Wave will be a conical shape and can be distorted from ambient noise and materials that absorb sound (i.e cardboard, tennis ball, etc.)

Sound Calculation Formula: Distance L = 1/2 × T × C

where L is the distance, T is the time between the emission and reception, and C is the sonic speed. (The value is multiplied by 1/2 because T is the time for go-and-return distance.)

Arduino and Servo Motor Tutorial
By: Matthew Jourden
Brighton High School

```
#include <Servo.h>
Servo motor; // variable type Servo assigns a physical motor
int trigPin = 8; // front sensor attach const
int echoPin = 9;
int myservo = 4;

void setup()
{
motor.attach(myservo);
Serial.begin(9600);
}
```

```
void loop()
{
  long duration, inches; //Sends out signal for ultrasonic sensor
  pinMode(trigPin, OUTPUT);
  //Next four lines of code pulsate a single out then in for defined time
  //calulates the duration the sound wave takes to come back to sensor
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

 // converts duration (Sound) to inches
 duration = pulseIn(echoPin, HIGH);

 // sends data to function microsecondsToInches
 inches = microsecondsToInches(duration);
 Serial.print ("Inches= ");
 Serial.println (inches);


// Turning the motor on/off
// Writing to motor 0 = Full Speed Forward;
// 180 = Full Speed Reverse; 90 = Mid Point Of motor
if (inches > 2)
{
  motor.attach(myservo);
  motor.write(0);
  delay(1000);
}

 else
 {
   // Detach disconnects the motor from the Arduino Board;
   //Allowing the motor to turn off

   motor.detach();
   delay(1000);
 }
}
```

```
long microsecondsToInches(long microseconds)
// Function is a sub program that can be utilized
// in many parts of the program without rewriting the code.
// The Return Function returns a value of microseconds to
// the main program body

{
  return microseconds / 74 / 2;
}
```

Assignment: Add LED RGB Light to turn

Green: When distance is greater than 4 inches

Yellow: When Distance is between 2 and 4 inches

Red: When distance is less than 2 inches

See Class Website > Tutorial Library > RGB LED Tutorial