

Arduino: Servo Motors

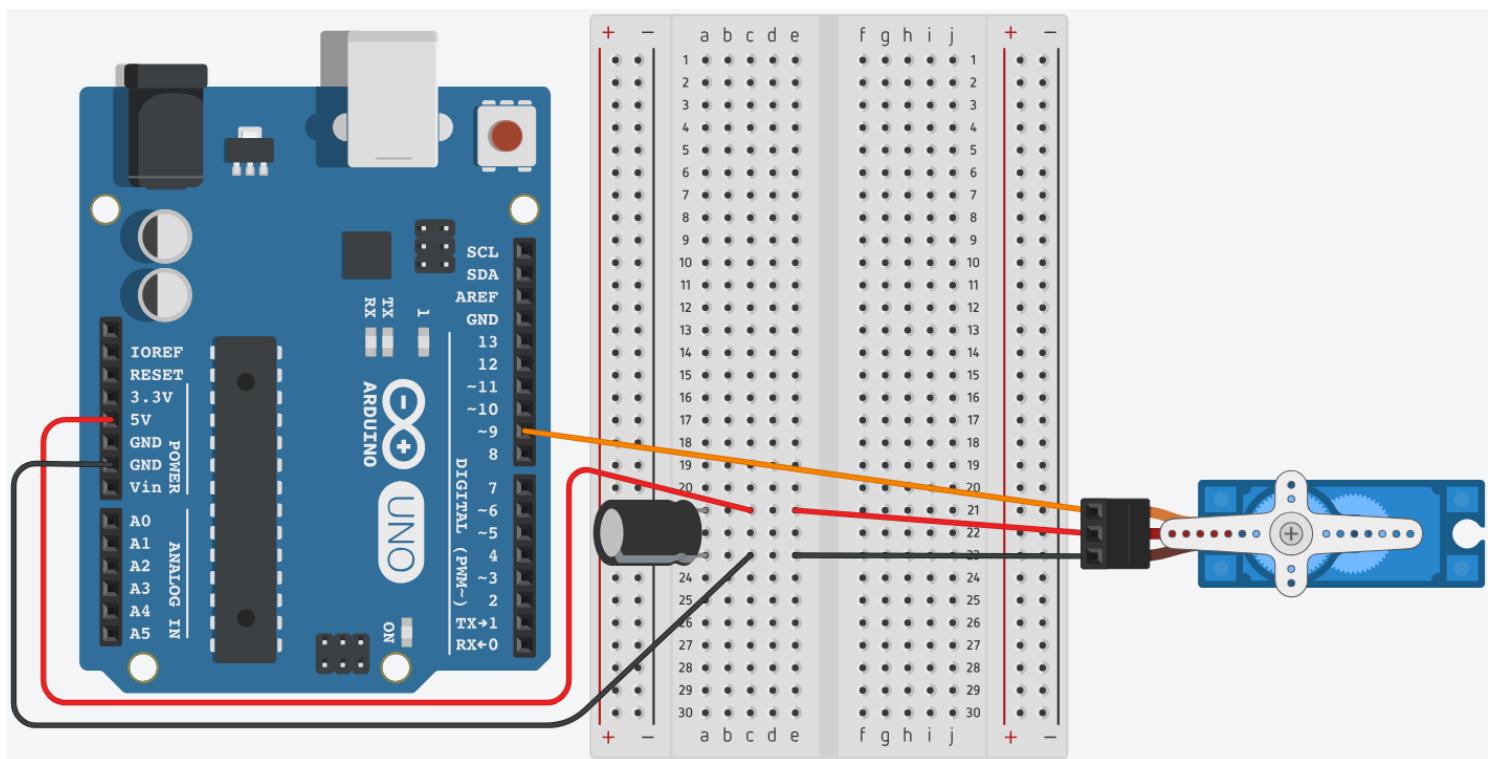
Diagrams & Code

Brown County Library

Project 01: Sweep

Components needed:

- Arduino Uno board
- breadboard
- 5 jumper wires
- Servo motor
- Capacitor - 100 μF



```

/*
Servo 01 : Sweep
Source: Code adapted from Adafruit Arduino - Lesson 14. Servo Motors
(https://learn.adafruit.com/adafruit-arduino-lesson-14-servo-motors)
*/

```

```

#include <Servo.h> // indicate that we want to use the Servo library

Servo servo;          // initialize the Servo library

int servoPin = 9;      // control lead of servo connected to pin 9

int angle = 0; // set the initial servo position in degrees

void setup() {
  servo.attach(servoPin); // indicate that servo motor is attached to the servoPin
}

void loop() {
  for(angle = 0; angle < 180; angle++) // counts up from 0 to 180 (max angle) using the variable "angle"
  {
    servo.write(angle); // set the new angle
    delay(15);         // delay between the steps
  }

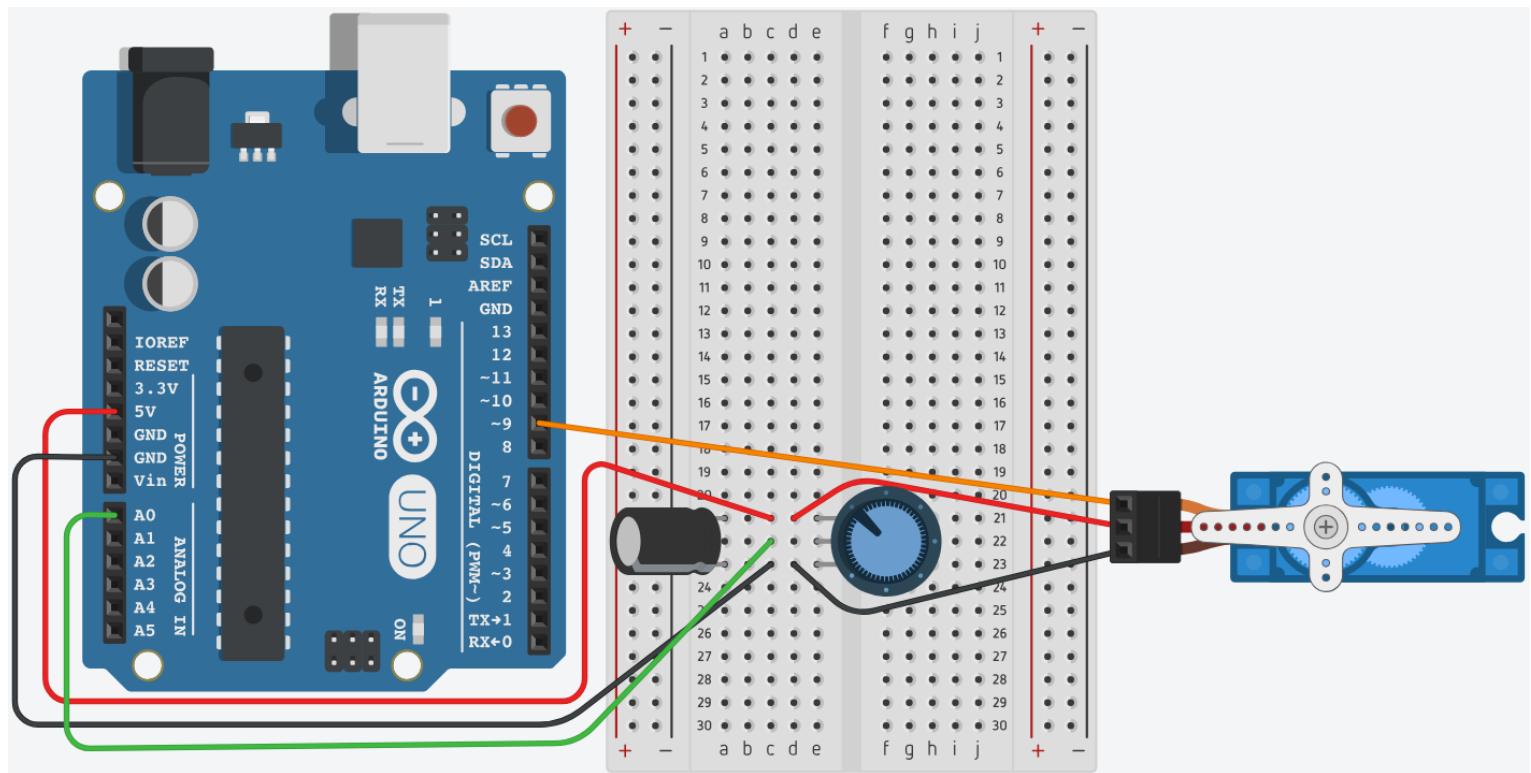
  for(angle = 180; angle > 0; angle--) // counts down from 0 to 180 (max angle) using the variable "angle"
  {
    servo.write(angle); // set the new angle
    delay(15);         // delay between the steps
  }
}

```

Project 02: Potentiometer

Components needed:

- Arduino Uno board
- breadboard
- 6 jumper wires
- Servo motor
- Capacitor - 100 μF
- 10k potentiometer



```
/*
Servo 02 : Potentiometer
Source: Code adapted from Arduino.cc Knob (https://www.arduino.cc/en/Tutorial/Knob) and
Sparkfun's Inventor Kit Experiment Guide for Arduino V4.0 – Circuit 3A: Servo Motors
(https://learn.sparkfun.com/tutorials/sparkfun-inventors-kit-experiment-guide---v40/circuit-3a-servo-motors)
*/
```

```
#include <Servo.h> // indicate that we want to use the Servo library

Servo servo; // initialize the Servo library

int servoPin = 9; // control lead of servo connected to pin 9

int potPin = 0; // analog pin used to connect the potentiometer

int val; // variable to read the value from the analog pin
int angle; // variable for the angle that we will calculate

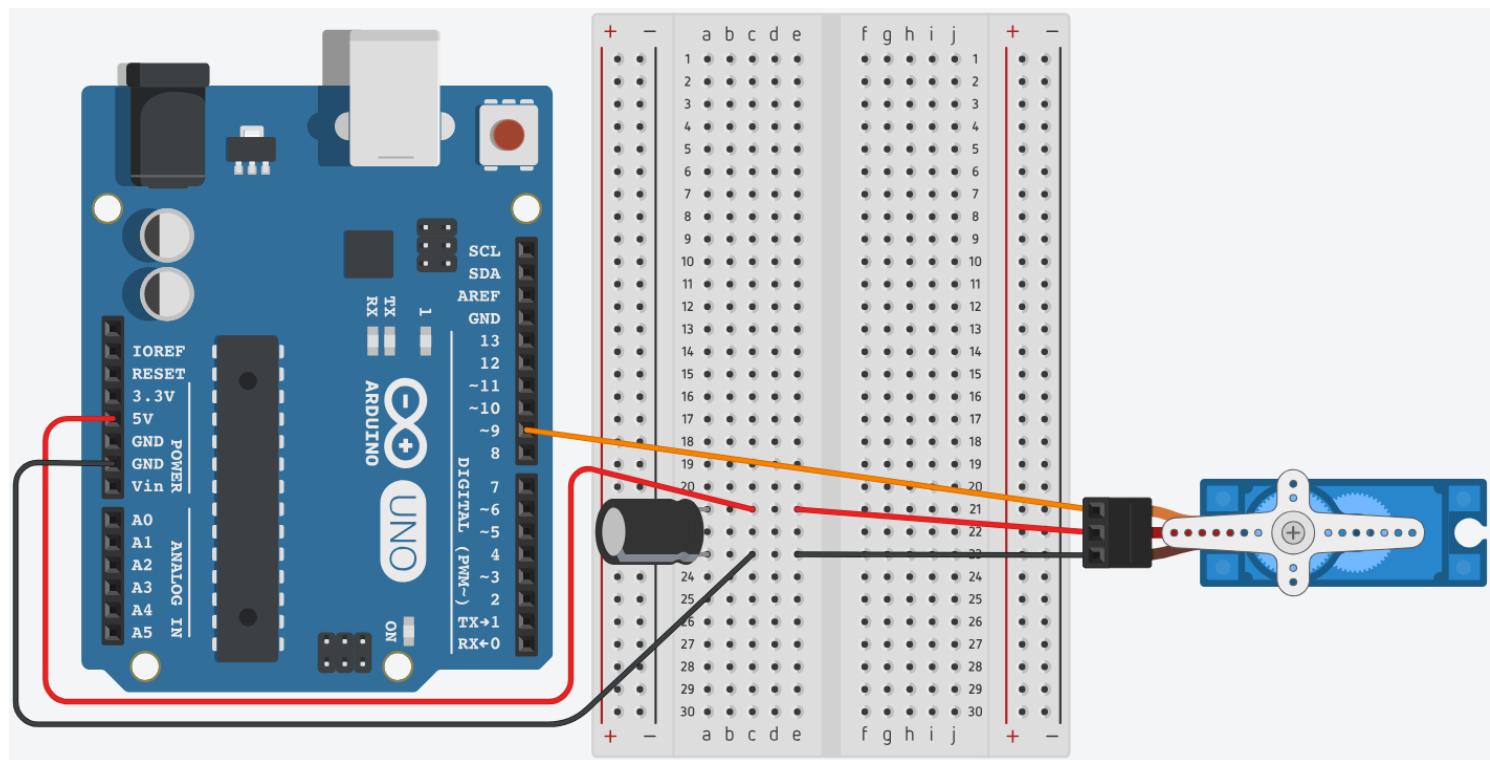
void setup() {
    servo.attach(servoPin); // indicate that servo motor is attached to the servoPin
}

void loop() {
    val = analogRead(potPin); // reads the value of the potentiometer (value between 0 and 1023)
    angle = map(val, 0, 1023, 0, 180); // scale that value to use it with the servo (value between 0 and 180)
    servo.write(angle); // sets the servo position according to the scaled value
    delay(15); // waits for the servo to get there
}
```

Project 03: Serial Monitor

Components needed:

- Arduino Uno board
- breadboard
- 5 jumper wires
- Servo motor
- Capacitor - 100 μF



```
/*
Servo 03 : Serial Monitor
Source: Code adapted from Sparkfun's Inventor Kit Experiment Guide for Arduino V3.3 – Experiment 8:
Driving a Servo Motor (https://learn.sparkfun.com/tutorials/sik-experiment-guide-for-arduino---v33/experiment-8-driving-a-servo-motor)
*/
```

```
#include <Servo.h> // indicate that we want to use the Servo library

Servo servo; // initialize the Servo library

int servoPin = 9; // control lead of servo connected to pin 9

int angle; // establish the angle variable (to be used later)

void setup() {
    servo.attach(servoPin); // indicate that servo motor is attached to the servoPin
    Serial.begin(9600); // initialize the serial communication

    Serial.println("Type an angle (0-180) into the box above,"); // print two lines of instructions
    Serial.println("then click [send] or press [return]");
    Serial.println(); // and then a blank line
}

void loop() {
    while (Serial.available() > 0) // check to see if incoming data is available
    {
        angle = Serial.parseInt(); // if it is, we'll use parseInt() to pull out any numbers
        angle = constrain(angle, 0, 180); // make sure the number is between 0 & 180

        Serial.print("Setting angle to "); // print a message in the serial monitor with the new angle
        Serial.println(angle);

        servo.write(angle); // move the servo to that angle
    }
}
```

Ideas to Build On

Build a simple knock lock that would open the door after knocking on a piezo the correct number of times!
See page 9 of this document. Warning – this one is a bit finicky!

Build a prototype of the automatic sunglasses.

This project uses an Arduino Mini – can you adjust what is found on the website with the materials you have on hand on? Remember, this will just be a prototype!

https://create.arduino.cc/projecthub/ashraf_minhaj/sunglass-bot-an-automated-pair-of-sunglasses-bdd1b6?ref=platform&ref_id=424_trending_&offset=94

Build a simple version of this servo “sunflower” – the motor rotates to follow the light hitting two photoresistors.

https://create.arduino.cc/projecthub/Rick_Findus/arduino-sunflower-c4fd84?ref=tag&ref_id=servo&offset=3

Learn More

Want to learn more about how servo motors, Arduino Libraries and capacitors work? Try these resources:

Adafruit All About Arduino Libraries. Bill Earl.

<https://learn.adafruit.com/adafruit-all-about-arduino-libraries-install-use?view=all>

Adafruit Arduino Lesson 14: Servo Motors. Simon Monk.

<https://learn.adafruit.com/adafruit-arduino-lesson-14-servo-motors?view=all>

Adafruit Tips, Tricks & Techniques: Arduino Libraries. Lady Ada and Tyler Cooper.

<https://learn.adafruit.com/arduino-tips-tricks-and-techniques/arduino-libraries>

Arduino – Libraries. <https://www.arduino.cc/en/Reference/Libraries>

Arduino – Servo Library. <https://www.arduino.cc/en/Reference/Servo>

Arduino Project Handbook. Mark Geddes. 2016. Pg. 78-84.

Arduino Projects Book. Scott Fitzgerald, Michael Shiloh & Tom Igoe. 2012. Pg. 124-134.

With corrections found here: <https://forum.arduino.cc/index.php?topic=175831.0>

Exploring Arduino: Tools and Techniques for Engineering Wizardry. Jeremy Blum. 2013. Pg. 80-86.

How Does a Capacitor Work? Øyvind Nydal Dahl.

<https://www.build-electronic-circuits.com/how-does-a-capacitor-work/>

How Servo Motors Work & How to Control Servos Using Arduino. Dejan Nedelkovski.

<https://howtomechatronics.com/how-it-works/how-servo-motors-work-how-to-control-servos-using-arduino/>

Sparkfun SIK Experiment Guide for Arduino V3.3 – Experiment 8: Driving a Servo Motor.

<https://learn.sparkfun.com/tutorials/sparkfun-inventors-kit-experiment-guide---v40/circuit-4a-lcd-hello-world>

Sparkfun SIK Experiment Guide for Arduino V4.0 – Circuit 3A: Servo Motors.

<https://learn.sparkfun.com/tutorials/sparkfun-inventors-kit-experiment-guide---v40/circuit-3a-servo-motors>

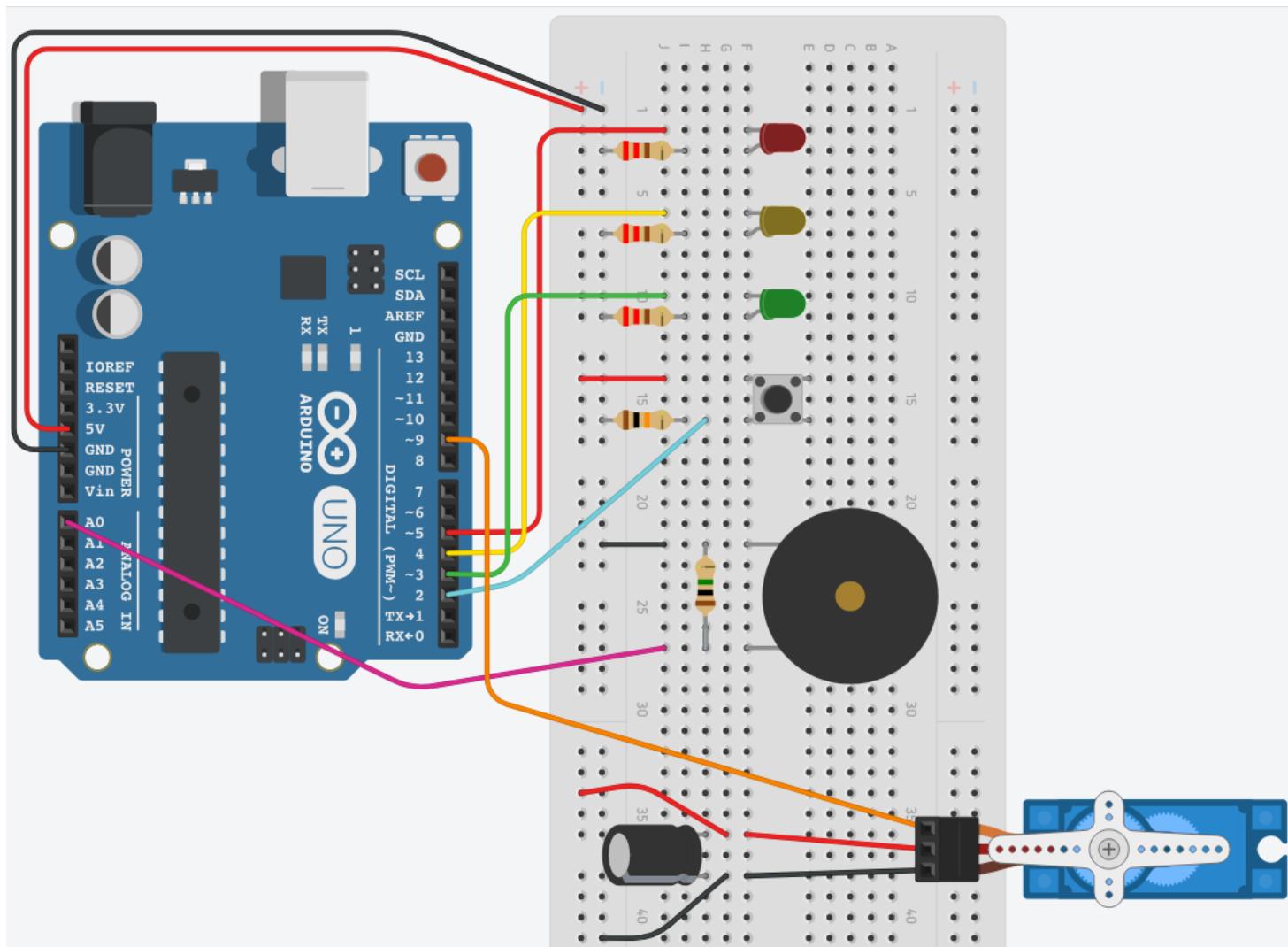
Sparkfun Capacitor Tutorial. <https://learn.sparkfun.com/tutorials/capacitors>

Sparkfun Hobby Servo Tutorial. <https://learn.sparkfun.com/tutorials/hobby-servo-tutorial>

Extra Project: Knock Lock

Components needed:

- Arduino Uno board
- breadboard
- 14 jumper wires
- Servo motor
- Capacitor - 100 μF
- Push button
- Piezo
- 3 x LEDs (green, yellow, red)
- 3 x 220 ohm resistor
- 10k ohm resistor
- 1 million ohm resistor



**/*Created 18 September 2012
by Scott Fitzgerald Thanks to Federico Vanzati for improvements
<http://arduino.cc/starterKit>
This example code is part of the public domain.
*/**

```
#include <Servo.h>
Servo servo9; // Pin connected to servo mpo
const int piezo = A0; // Pin connected to piezo
const int switchPin = 2; // Pin connected to servo
const int yellowLed = 3; // Pin connected to yellow LED
const int greenLed = 4; // Pin connected to green LED
const int redLed = 5; // Pin connected to red LED
int knockVal; // Value for the knock strength
int switchVal;
const int quietKnock = 10; // Set min value that will be accepted
const int loudKnock = 100; // Set max value that will be accepted
boolean locked = false; // A true or false variable
int numberOfKnocks = 0; // Value for number of knocks
void setup() {
    servo9.attach(9);
    pinMode(yellowLed, OUTPUT); // Set LED pins as outputs
    pinMode(greenLed, OUTPUT);
    pinMode(redLed, OUTPUT);
    pinMode(switchPin, INPUT); // Set servo pin as input
    Serial.begin(9600);
    digitalWrite(greenLed, HIGH); // Green LED is lit when the
    // sequence is correct
    servo9.write(0);
    Serial.println("The box is unlocked!");
}
void loop() {
    if (locked == false) {
        switchVal = digitalRead(switchPin);
        if (switchVal == HIGH) {
            locked = true;
            digitalWrite(greenLed, LOW);
            digitalWrite(redLed, HIGH);
            servo9.write(90);
            Serial.println("The box is locked!");
            delay(1000);
        }
    }
    if (locked == true) {
        knockVal = analogRead(piezo); // Knock value is read by analog pin
        if (numberOfKnocks < 3 && knockVal > 0) {
            if (checkForKnock(knockVal) == true) { // Check for correct
```

```

// number of knocks
numberOfKnocks++;
}
Serial.print(3 - numberOfKnocks);
Serial.println(" more knocks to go");
}
if (numberOfKnocks >= 3) { // If 3 valid knocks are detected,
// the servo moves
locked = false;
servo9.write(0);
delay(20);
digitalWrite(greenLed, HIGH);
digitalWrite(redLed, LOW);
Serial.println("The box is unlocked!");
numberOfKnocks = 0; // resets number of knocks to 0
}
}
}

boolean checkForKnock(int value) { // Checks knock value
if (value > quietKnock && value < loudKnock) { // Value needs to be
// between these
digitalWrite(yellowLed, HIGH);
delay(50);
digitalWrite(yellowLed, LOW);
Serial.print("Valid knock of value ");
Serial.println(value);
return true;
}
else { // If value is false then send this to the IDE serial
Serial.print("Bad knock value ");
Serial.println(value);
return false;
}
}

```