

<https://www.youtube.com/watch?v=wNSz-KCOxB4>

<https://www.youtube.com/watch?v=cpdjQ0gheDQ>

<https://www.youtube.com/watch?v=uUtAQnKdkY0>



Useless Box Arduino



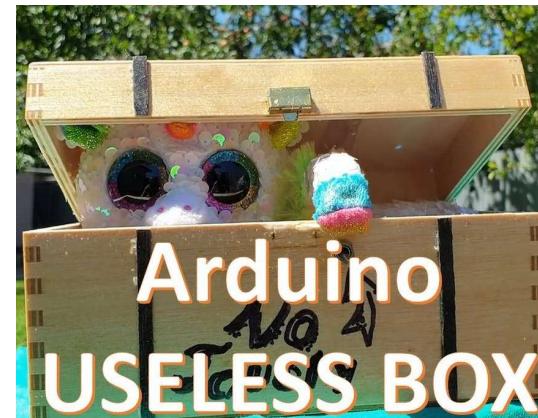
<https://www.hackster.io/viorelracoviteanu/useless-box-with-arduino-d67b47>



Useless Box controlled by Arduino DIY



How to Make a Useless Box | Amazing DIY Project



<https://www.youtube.com/watch?v=cpdjQ0gheDQ>



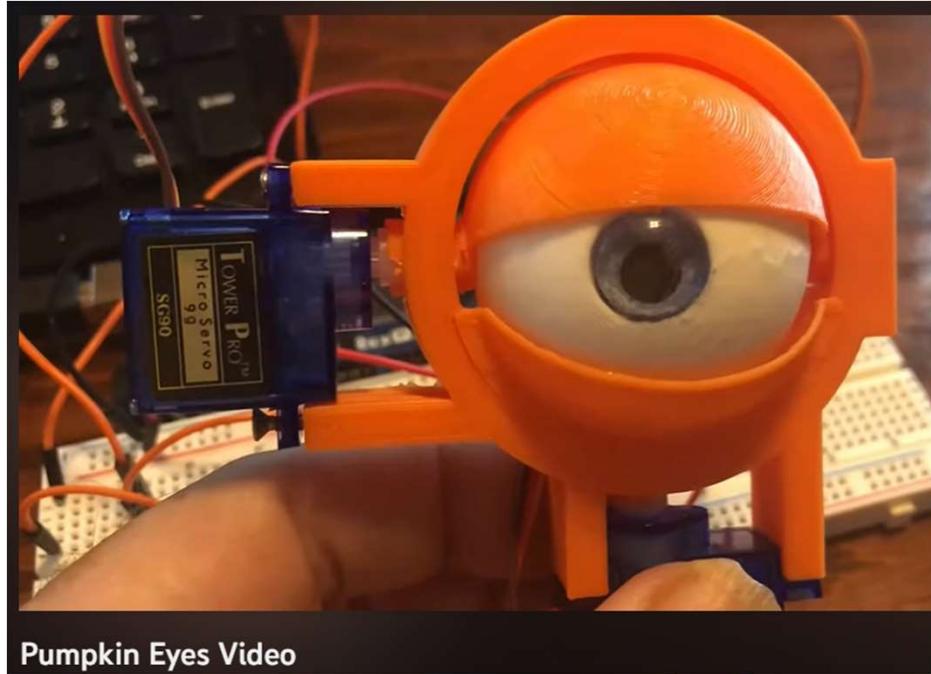
DIY Useless Box with Arduino

<https://www.youtube.com/watch?v=4gGtBTHyiNc>

https://www.youtube.com/watch?v=kTkNVE17_lc



<https://www.youtube.com/watch?v=jQcxa1lEXb0>



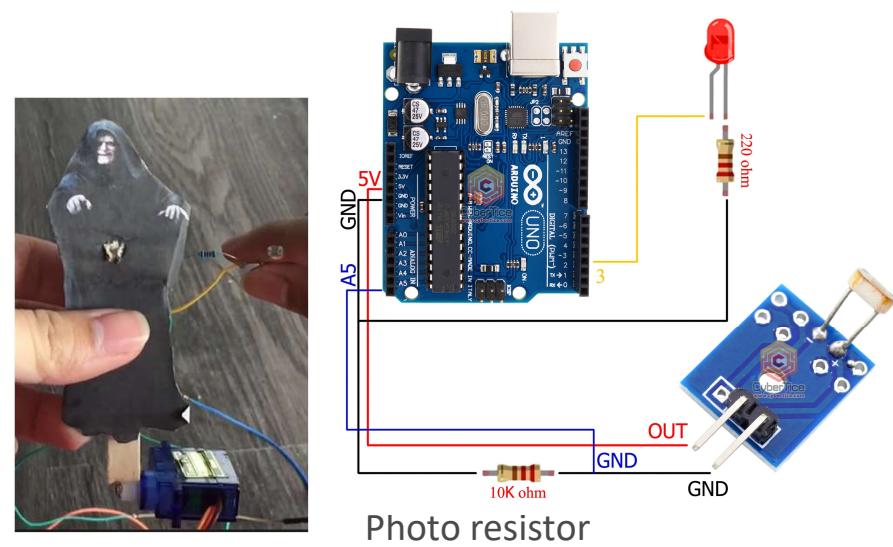
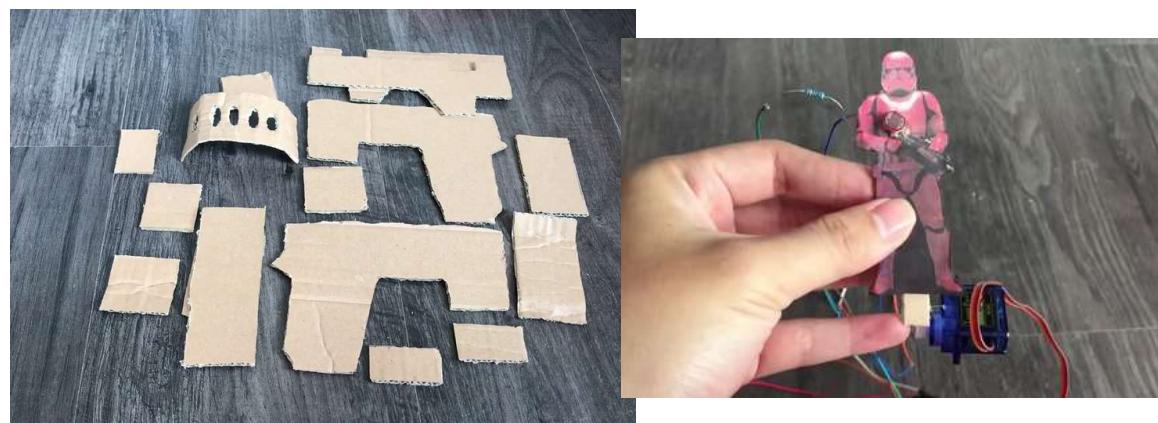
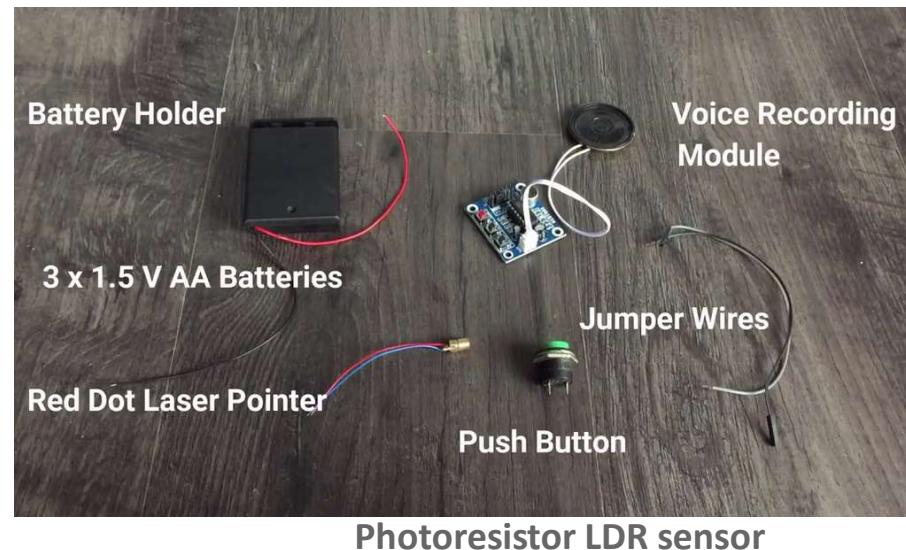
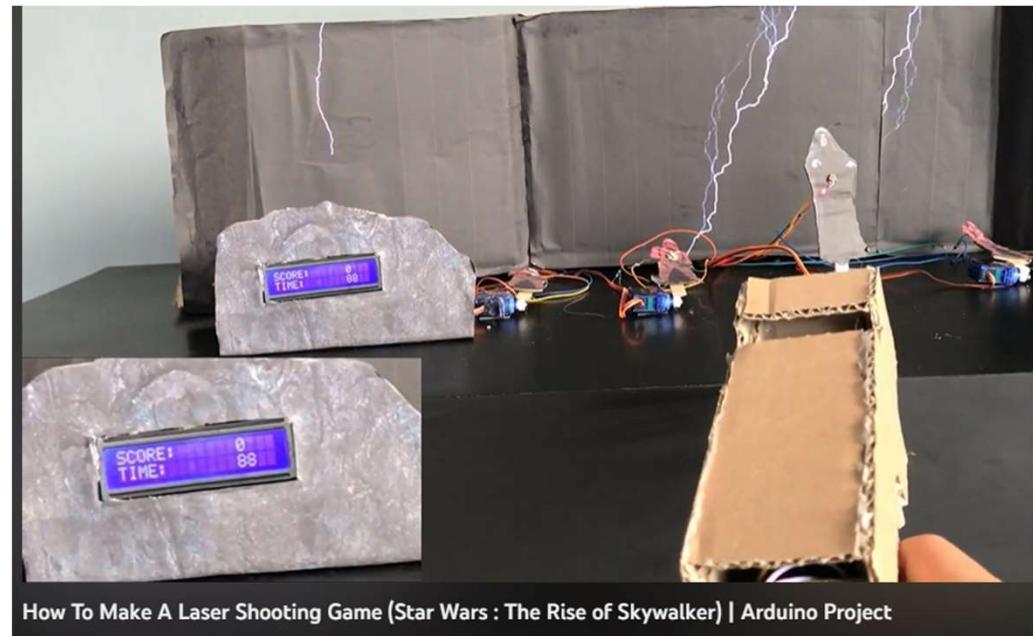
Pumpkin Eyes Video

Required Hardware:

Arduino Board
Servo Motor SG90
3D Printer for models
Paperclip
Mini breadboard
Jumper Wires



<https://www.instructables.com/Laser-Shooting-Game-Star-Wars/>



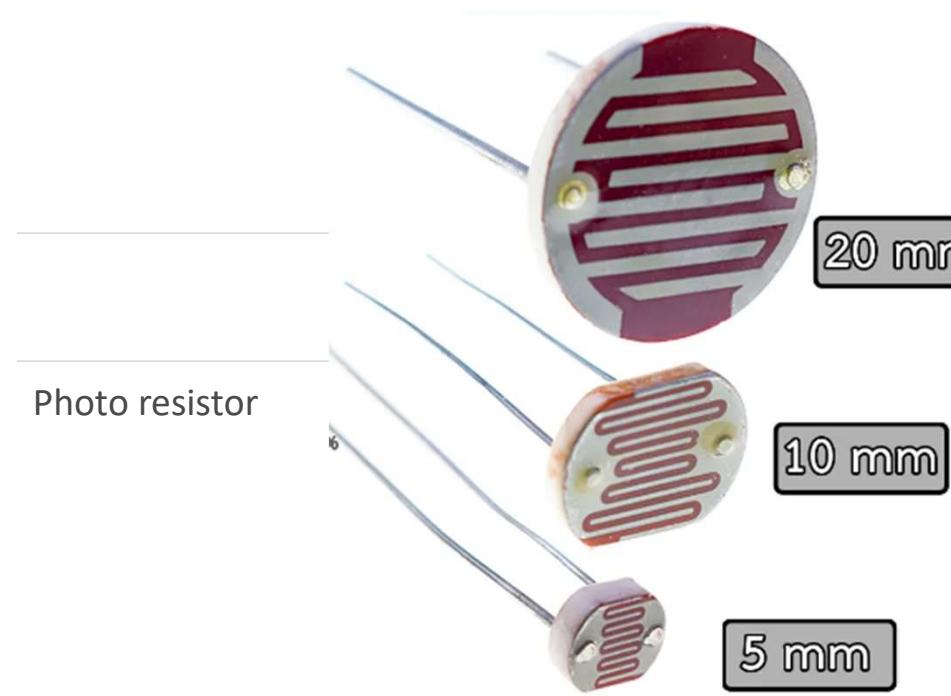
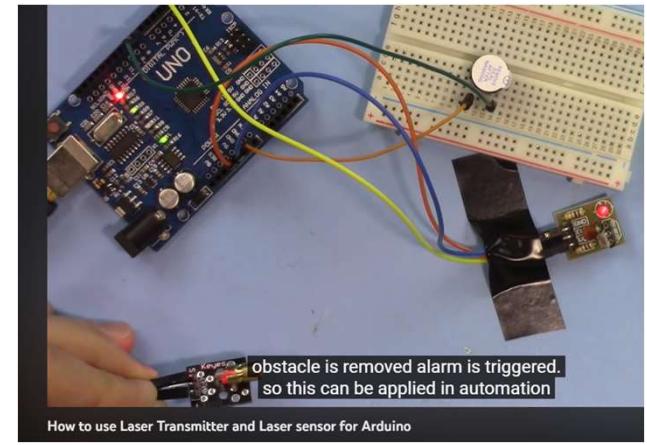
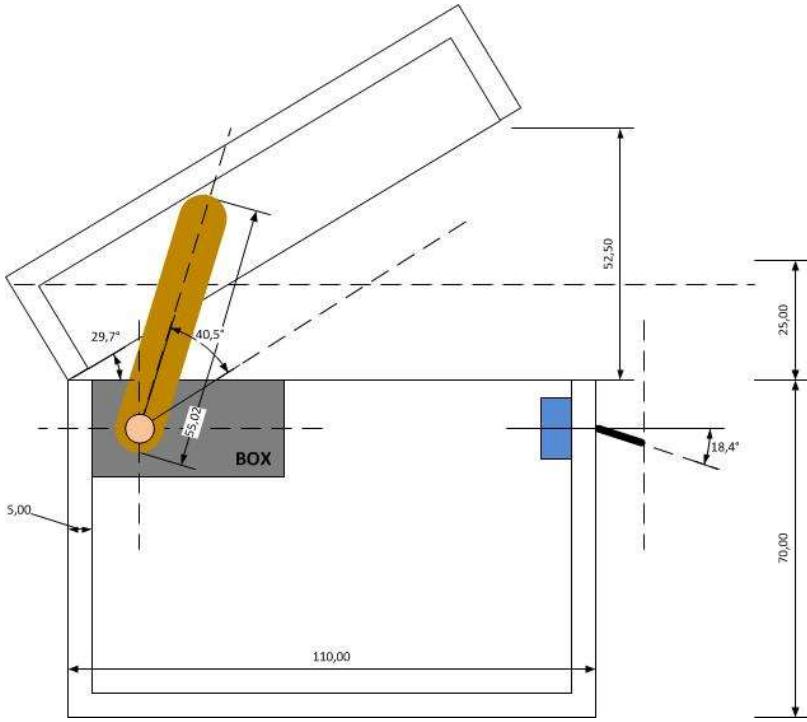
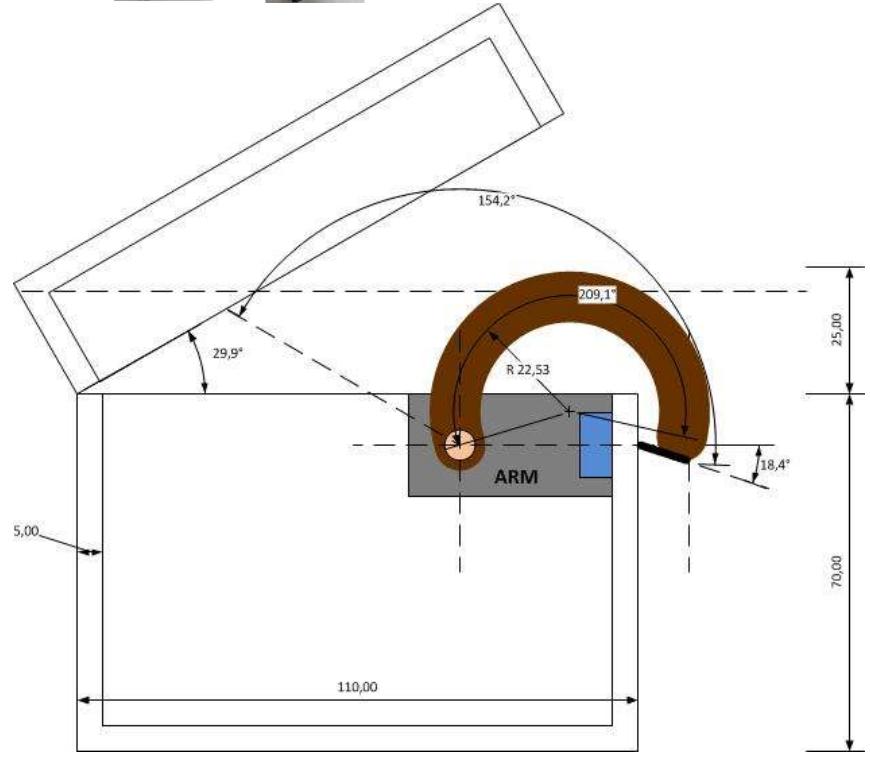
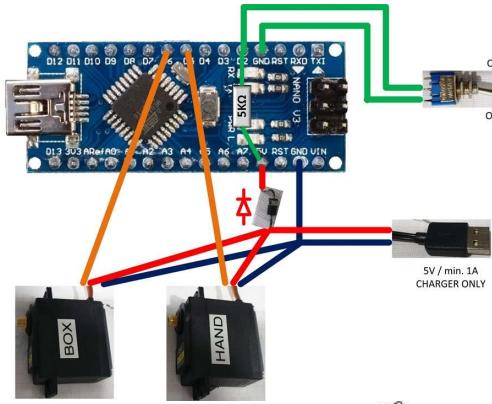
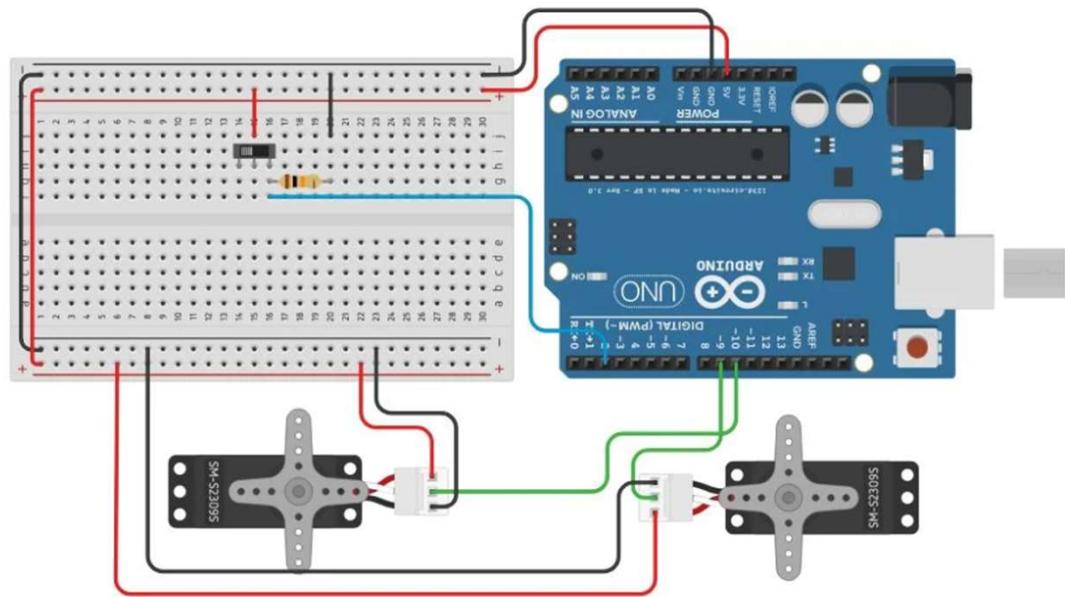


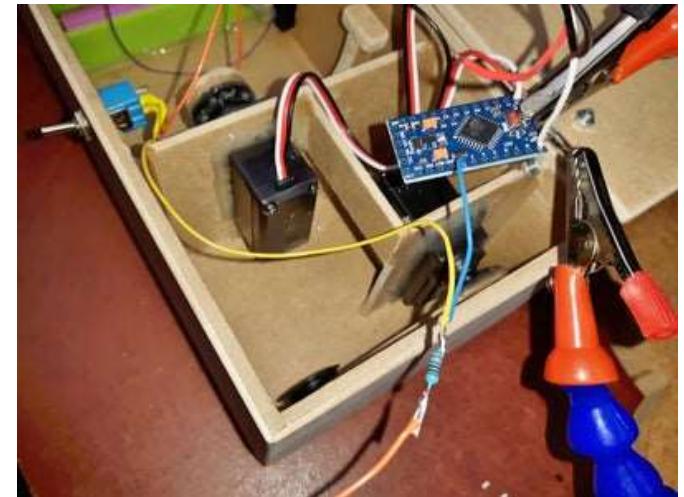
Photo resistor



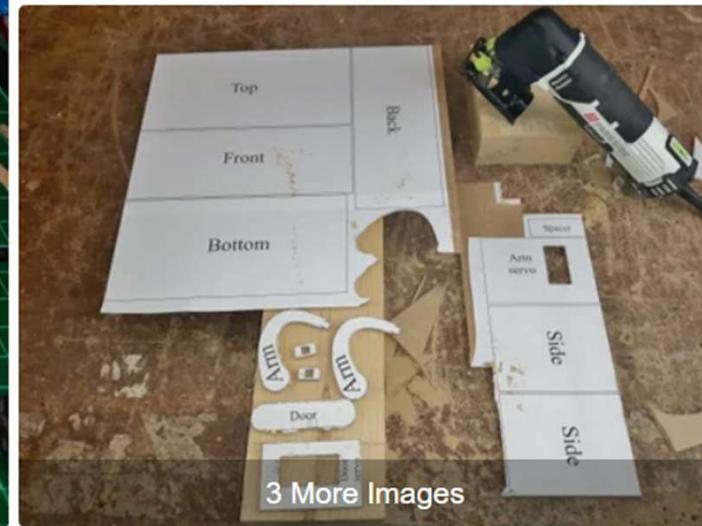
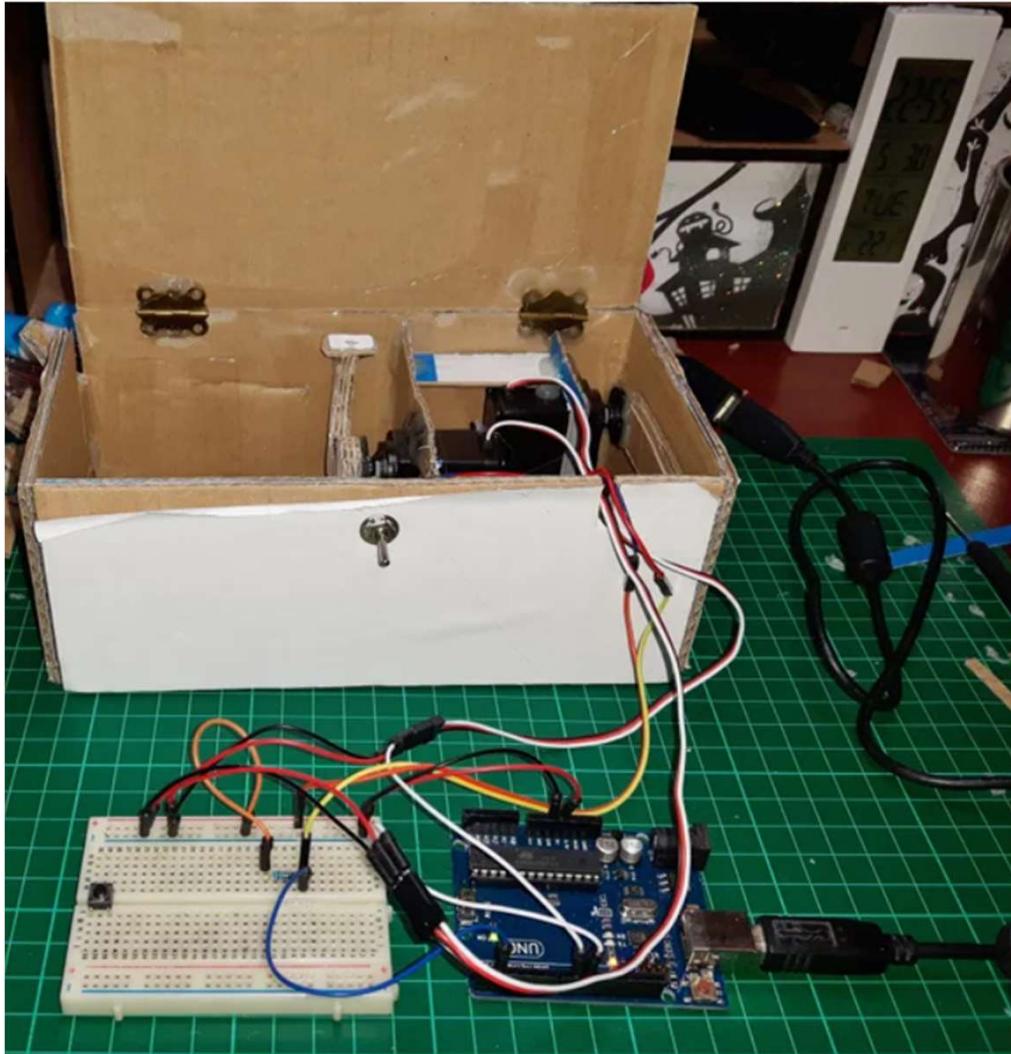


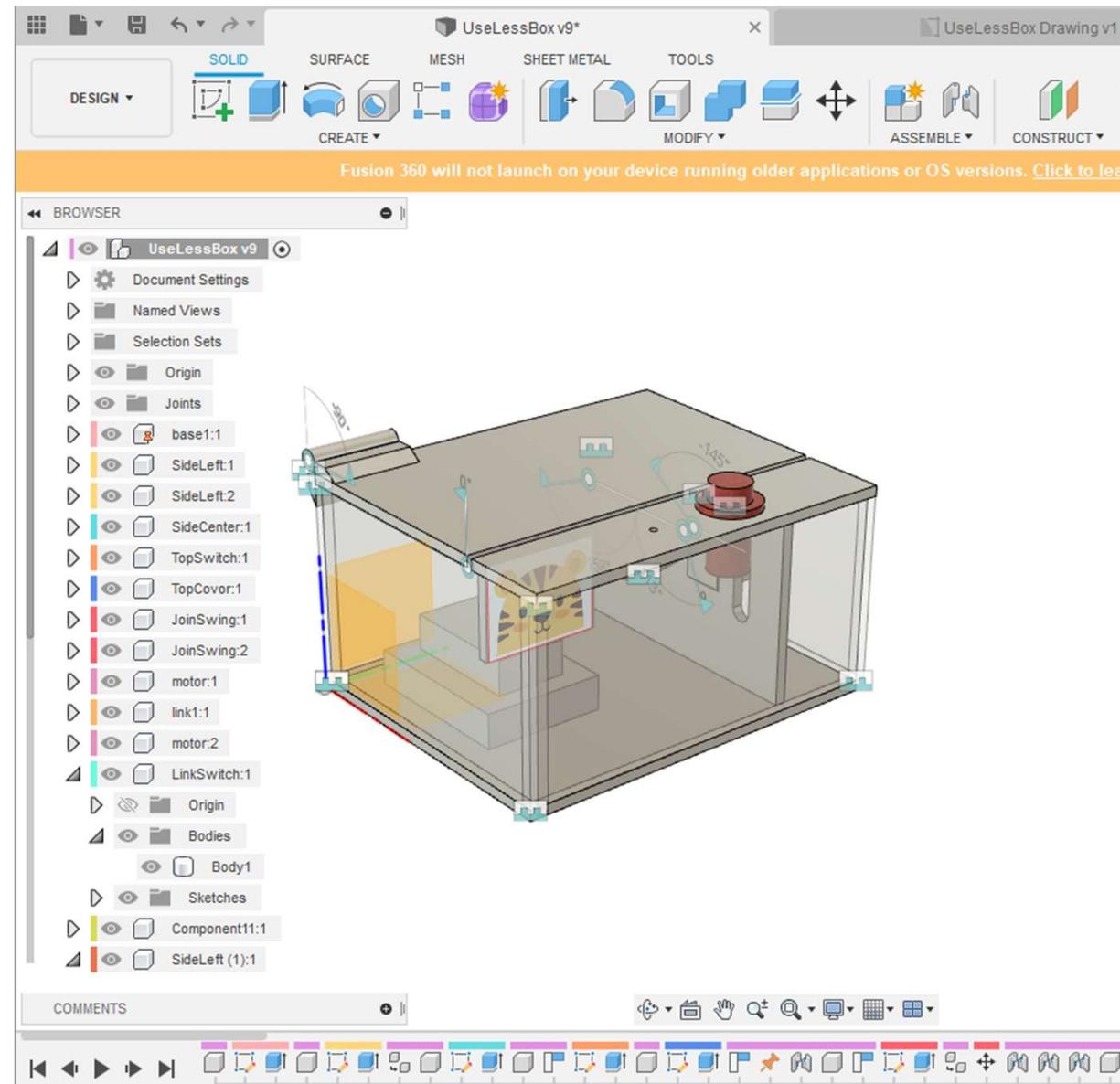
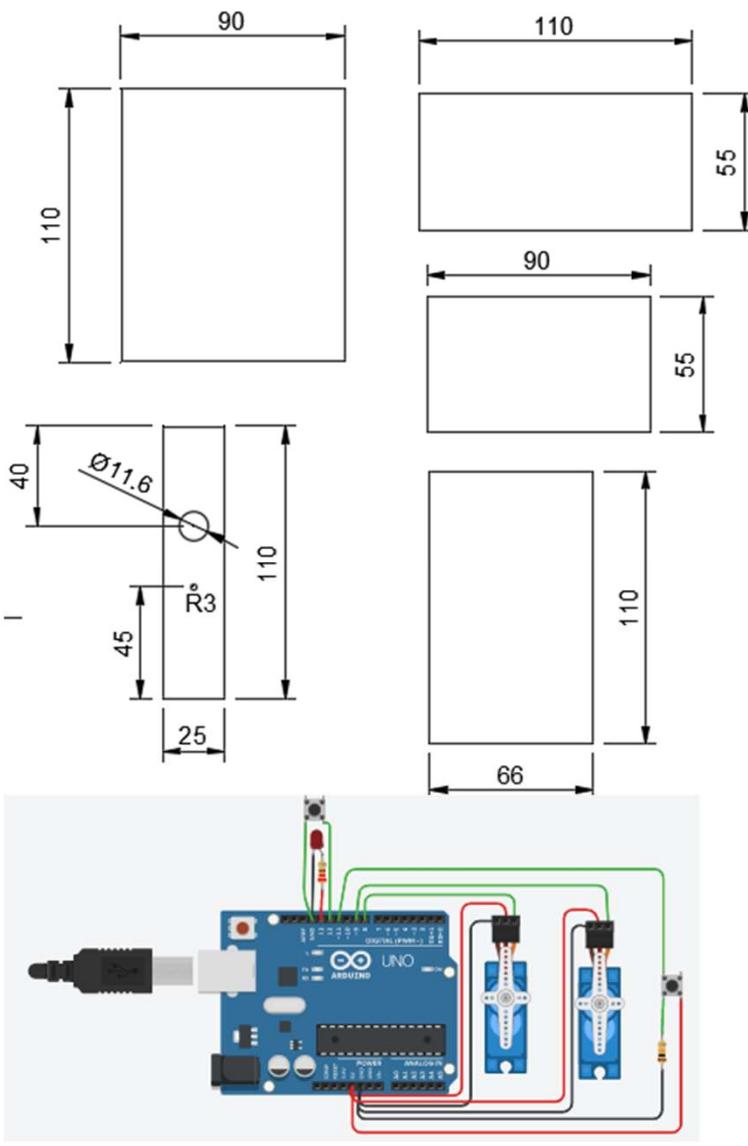


3 More Images



<https://www.instructables.com/Useless-Box-4/>

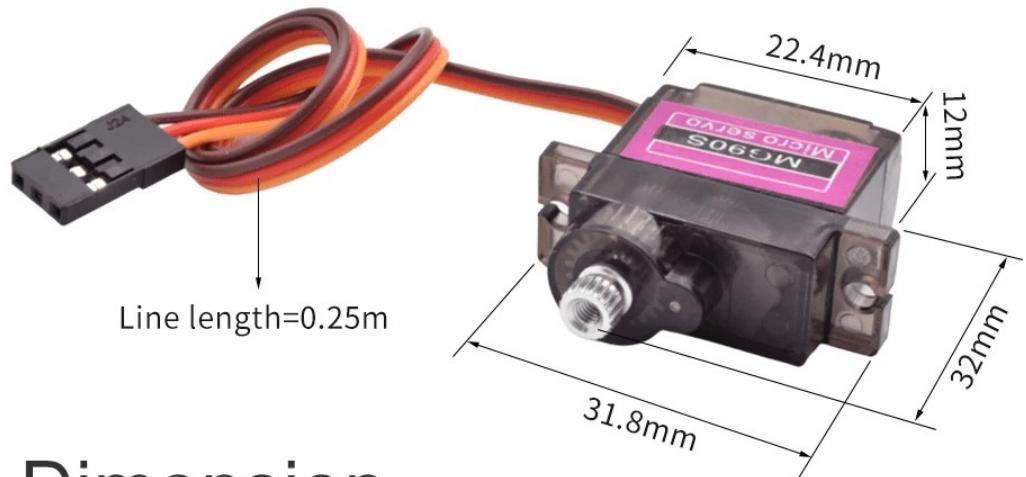




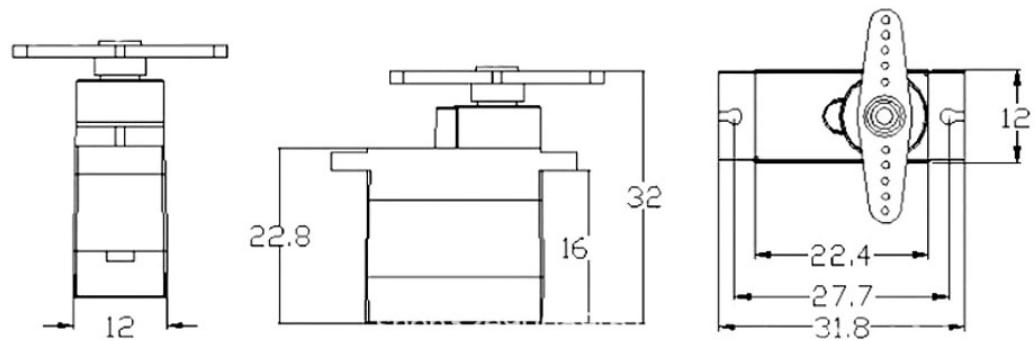
Useless Box 1 ชุด

- สวิตซ์กด ติดปล่อยดับ แบบมีเกลียวล็อก 1 ตัว
- MG90S Servo Motor หมุน 0-180 องศา 2 ตัว
- หลอด LED 1 ดวง
- Arudino 1 ตัว
- บานพับ 1 นิ้ว 2 ตัว

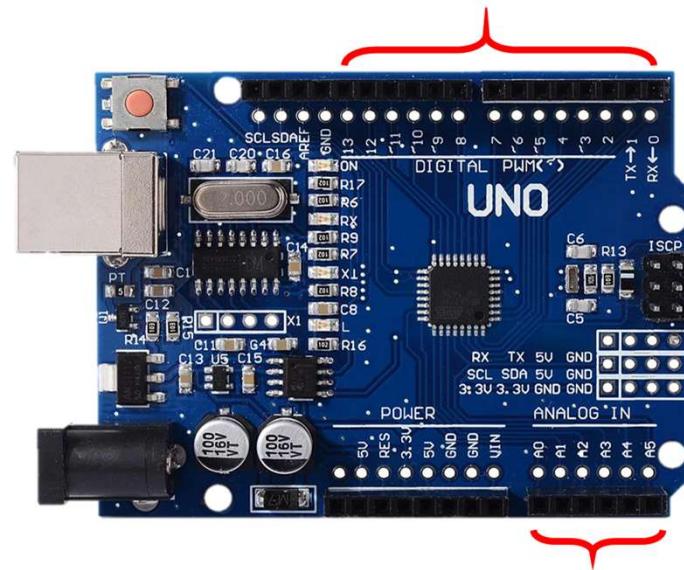




Dimension



GPIO ขาดิจิตอล (Input/output)



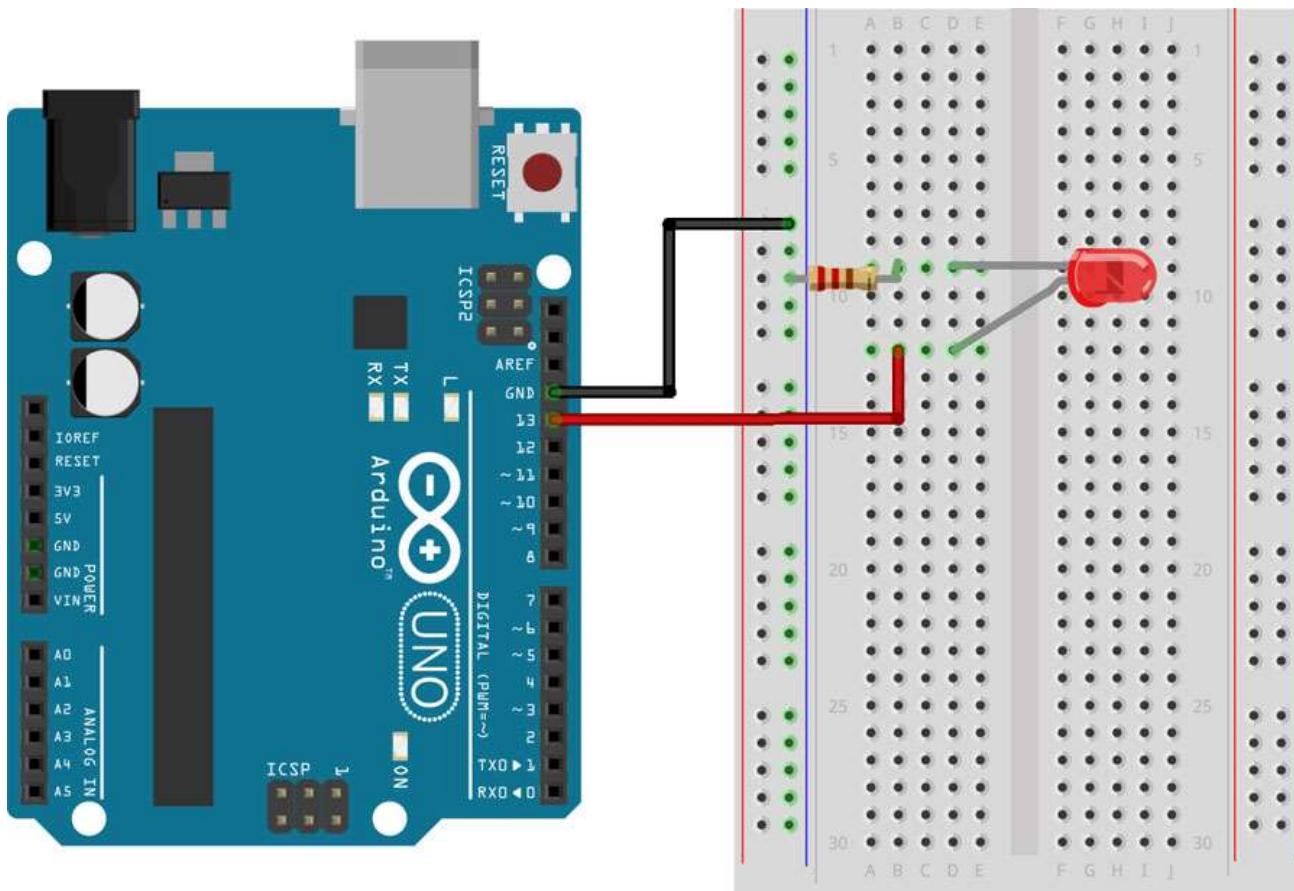
ดิจิตอล คือเปิดหรือปิด (1-13) Input/Output
5v (ON) 0v (OFF)

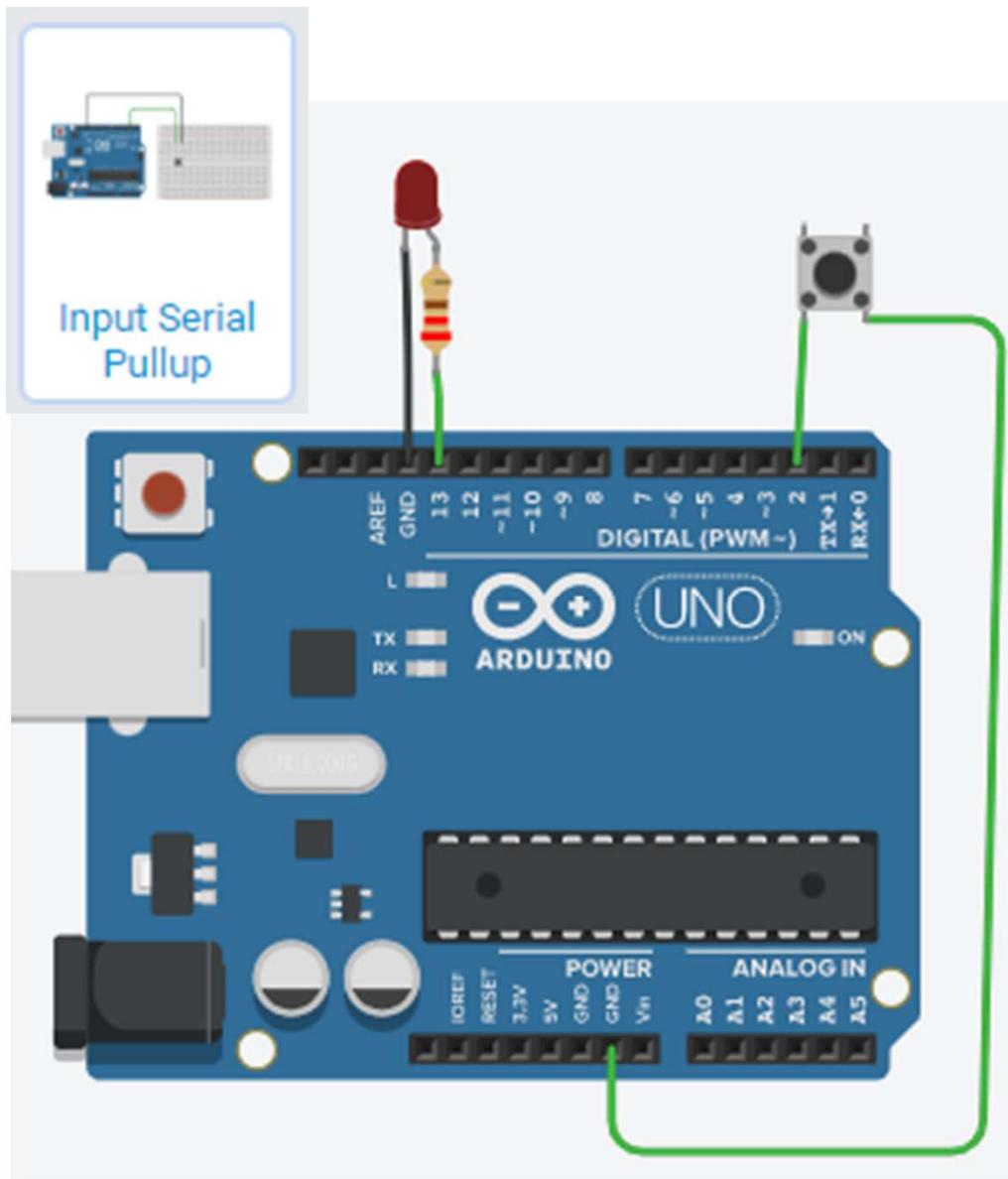
อนาล็อก คือระดับสัญญาณคล้ายๆระดับน้ำ ระดับตั้งแต่ 0-5v

GPIO ขาดิจิตอล (input/output)

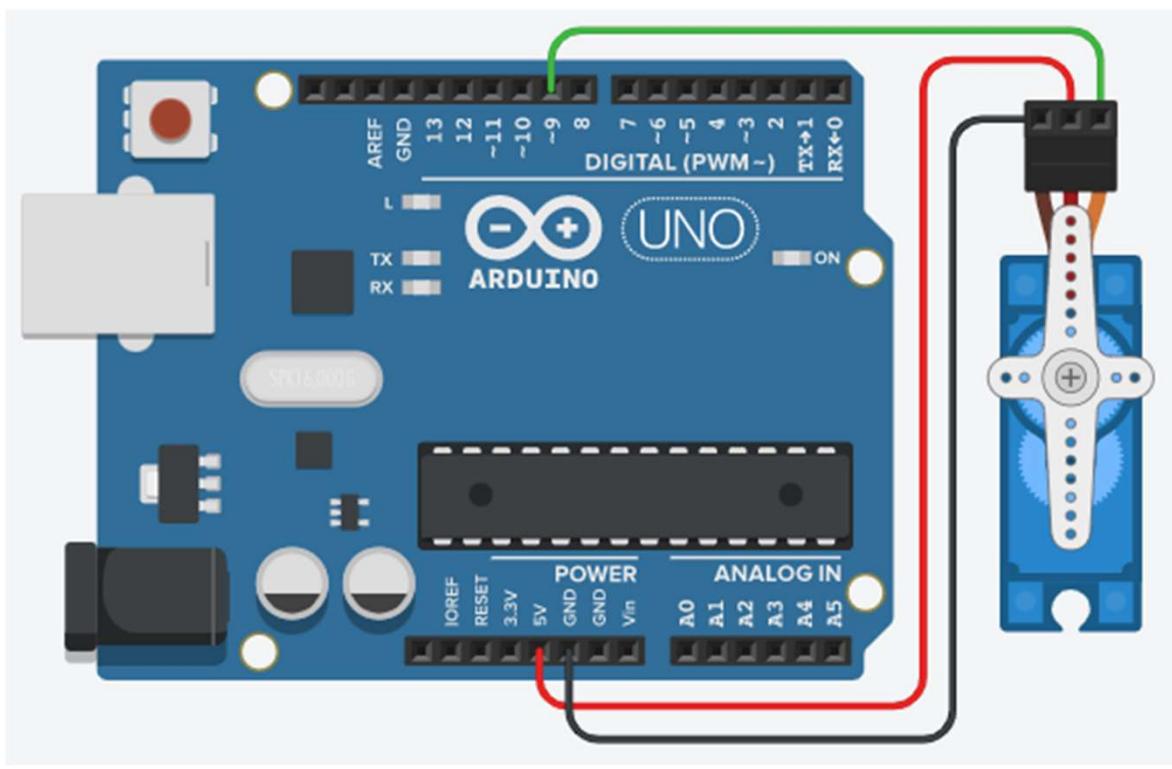
GPIO ขาอนาล็อก (input)

<https://www.tinkercad.com/>





```
void setup() {  
    Serial.begin(9600);  
    pinMode(2, INPUT_PULLUP);  
    pinMode(13, OUTPUT);  
}  
  
void loop() {  
  
    int sensorVal = digitalRead(2);  
    Serial.println(sensorVal);  
  
    if (sensorVal == HIGH) {  
        digitalWrite(13, LOW);  
    } else {  
        digitalWrite(13, HIGH);  
    }  
}
```



```
#include <Servo.h>
int pos = 0;
Servo servo_9;

void setup()
{
    servo_9.attach(9);
}

void loop()
{
    servo_9.write(0);
    delay(1000);
    servo_9.write(180);
    delay(1000);
}
```

```
for (int pos1 = 0; pos1 <= 3; pos1 += 1) {  
    digitalWrite(led13, HIGH);  
    delay(250); // Wait for 15 millisecond(s)  
    digitalWrite(led13, LOW);  
    delay(250);  
}
```

```
int a = digitalRead(12);  
  
if(a==LOW){  
    Serial.println("PullBotton");  
}
```

```
int rnd = random(1, 5);  
switch (rnd) {  
    case 1: MyStep1(); break;  
    case 2: MyStep2(); break;  
    case 3: MyStep3(); break;  
    case 4: MyStep4(); break;  
    case 5: MyStep5(); break;  
}
```

```

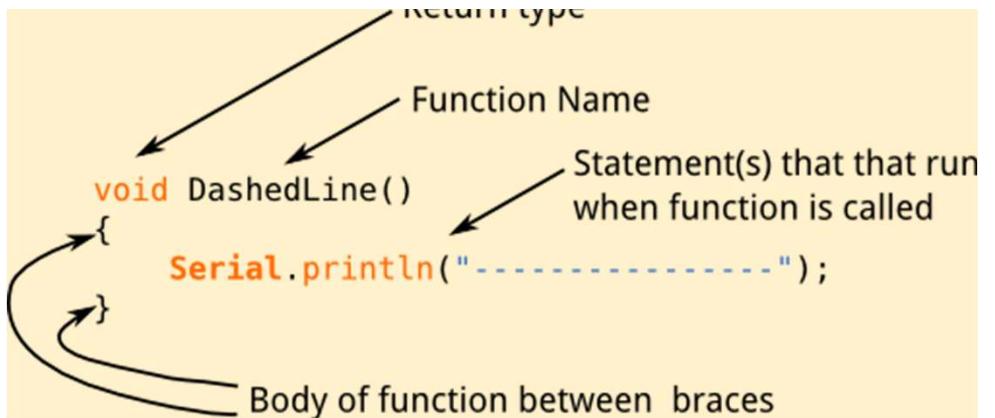
void setup() {
  Serial.begin(9600);

  DashedLine();
  Serial.println("| Program Menu |");
  DashedLine();
}

void loop() {
}

void DashedLine()
{
  Serial.println("-----");
}

```



```

void setup() {
  Serial.begin(9600);

  DashedLine();           ← Function is called here
  Serial.println("| Program Menu |");
  DashedLine();           ← Function is called again
}

void loop() {
}

void DashedLine()
{
  Serial.println("-----");
}

```

Annotations for the `DashedLine()` function:

- Function is created here**: Points to the opening brace of the `DashedLine()` function definition.
- Function is called here**: Points to the first call to `DashedLine()` in the `setup()` function.
- Function is called again**: Points to the second call to `DashedLine()` in the `setup()` function.

Anatomy of a C function

Datatype of data returned,
any C datatype.

"void" if nothing is returned.

Parameters passed to
function, any C datatype.

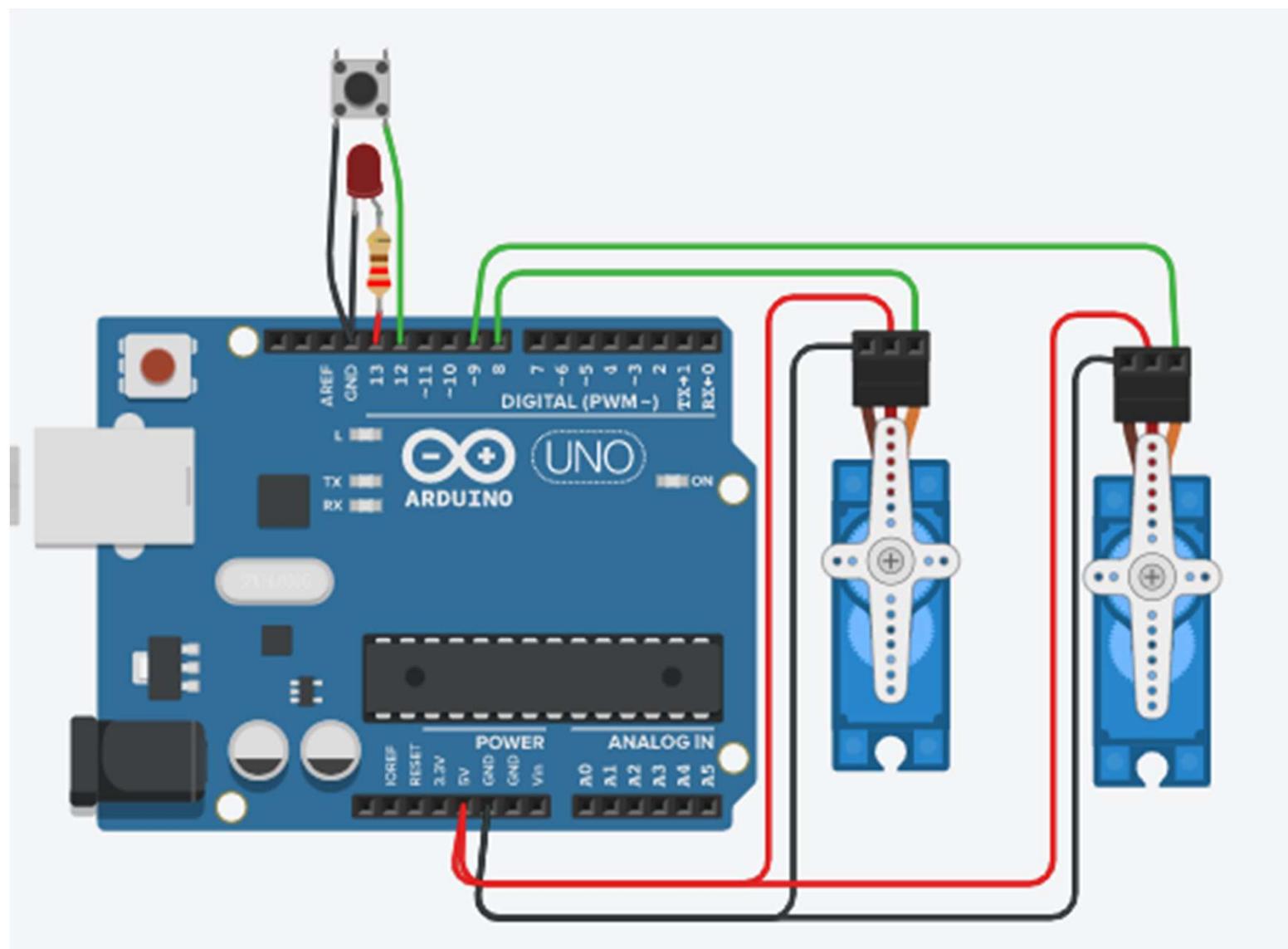
```
int myMultiplyFunction(int x, int y){  
    int result;  
    result = x * y;  
    return result;  
}
```

Function name

Return statement,
datatype matches
declaration.

Curly braces required.

```
void setup(){  
    Serial.begin(9600);  
}  
  
void loop() {  
    int i = 2;  
    int j = 3;  
    int k;  
  
    k = myMultiplyFunction(i, j); // k now contains 6  
    Serial.println(k);  
    delay(500);  
}  
  
int myMultiplyFunction(int x, int y){  
    int result;  
    result = x * y;  
    return result;  
}
```



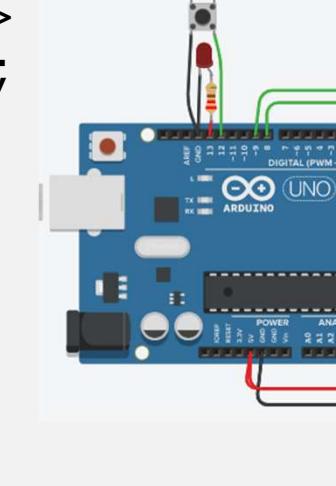
```
#include <Servo.h>
int buttonState = 0;
int sw12 = 12;
int led13 = 13;
int pos8 = 0;
int pos9 = 0;

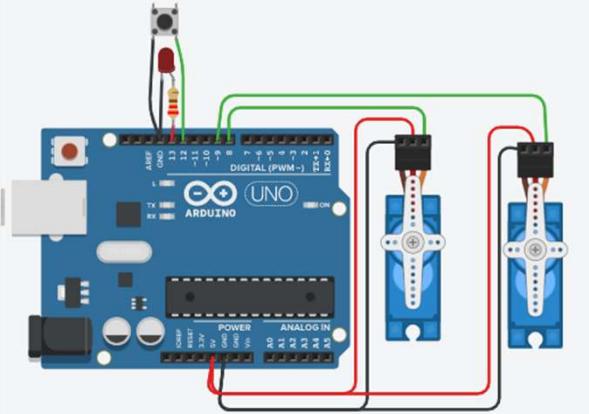
Servo servo_9;
Servo servo_8;

void setup()
{
    Serial.begin(9600);
    pinMode(sw12, INPUT_PULLUP); // INPUT_PULLUP
    pinMode(led13, OUTPUT);

    servo_8.attach(8);
    servo_9.attach(9);
    StartUp();
}

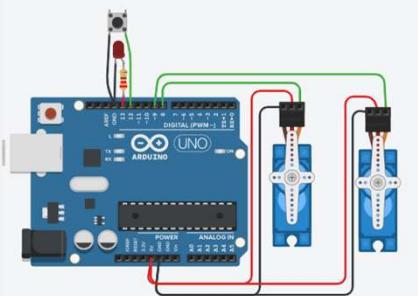

```





```
void StartUp() {  
  
    Serial.println("StartUp");  
    servo_8.write(0);  
    servo_9.write(0);  
    delay(1000);  
  
    servo_8.write(180);  
    servo_9.write(180);  
    delay(1000);  
    servo_8.write(0);  
    servo_9.write(0);  
  
    for (int pos1 = 0; pos1 <= 3; pos1 += 1) {  
        digitalWrite(led13, HIGH);  
        delay(250);  
        digitalWrite(led13, LOW);  
        delay(250);  
    }  
}
```

```
void loop()
{
int a = digitalRead(12);
if(a==LOW){
  Serial.println("PullButton");
  digitalWrite(led13,HIGH);
  int rnd = random(1, 5);
  switch (rnd) {
    case 1: MyStep1(); break;
    case 2: MyStep2(); break;
    case 3: MyStep3(); break;
    case 4: MyStep4(); break;
    case 5: MyStep5(); break;
  }
  MyTestMotorSlow();
  digitalWrite(13,LOW);
}
}
```



```
void MyStep1() {
  Serial.println("MyStep1");
}

void MyStep2() {
  Serial.println("MyStep2");
}

void MyStep3() {
  Serial.println("MyStep3");
}

void MyStep4() {
  Serial.println("MyStep4");
}

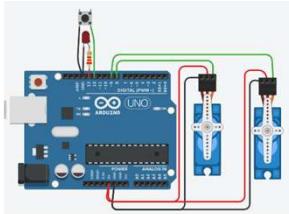
void MyStep5() {
  Serial.println("MyStep5");
}
```

```
void MyTestMotorSlow() {  
    Serial.println("MyTestMotorSlow");  
    MoveServo_8(0,180,1,15);  
    MoveServo_9(0,180,1,15);  
    delay(500);  
    MoveServo_8neg(180,0,1,15);  
    MoveServo_9neg(180,0,1,15);  
}
```

```
void MoveServo_8(int Ang1, int Ang2 , int Degree ,int DelayTime) {  
  
    for (pos8 = Ang1; pos8 <= Ang2; pos8 +=Degree) {  
        servo_8.write(pos8);  
        delay(DelayTime); // Wait for 15 millisecond(s)  
    };  
}
```

```
void MoveServo_8neg(int Ang1, int Ang2 , int Degree ,int DelayTime) {  
  
    for (int pos8  = Ang1; pos8 >= Ang2; pos8 -=Degree) {  
        servo_8.write(pos8);  
        delay(DelayTime); // Wait for 15 millisecond(s)  
    };  
}
```

```
void MoveServo_9(int Ang1, int Ang2 , int Degree ,int DelayTime) {  
    for (pos9 = Ang1; pos9 <= Ang2; pos9 +=Degree) {  
        servo_9.write(pos9);  
        delay(DelayTime); // Wait for 15 millisecond(s)  
    };  
}  
  
void MoveServo_9neg(int Ang1, int Ang2 , int Degree ,int DelayTime) {  
    for (pos9 = Ang1; pos9 >= Ang2; pos9 -=Degree) {  
        servo_9.write(pos9);  
        delay(DelayTime); // Wait for 15 millisecond(s)  
    };  
}
```

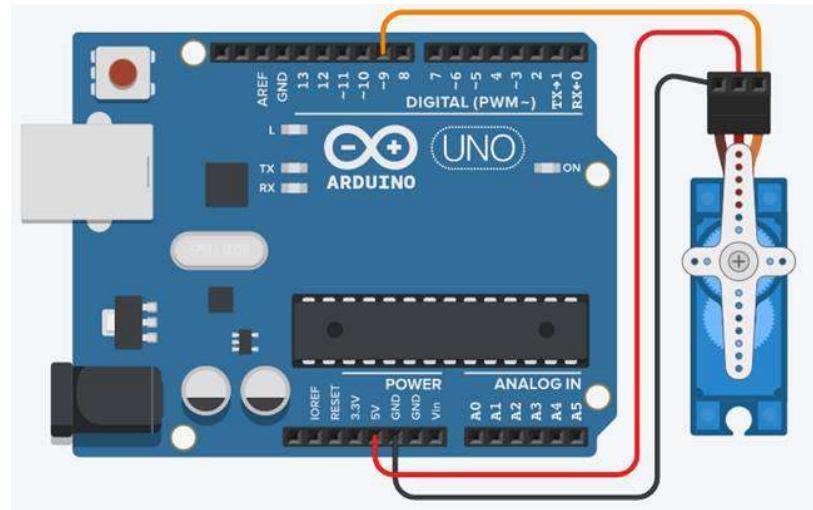
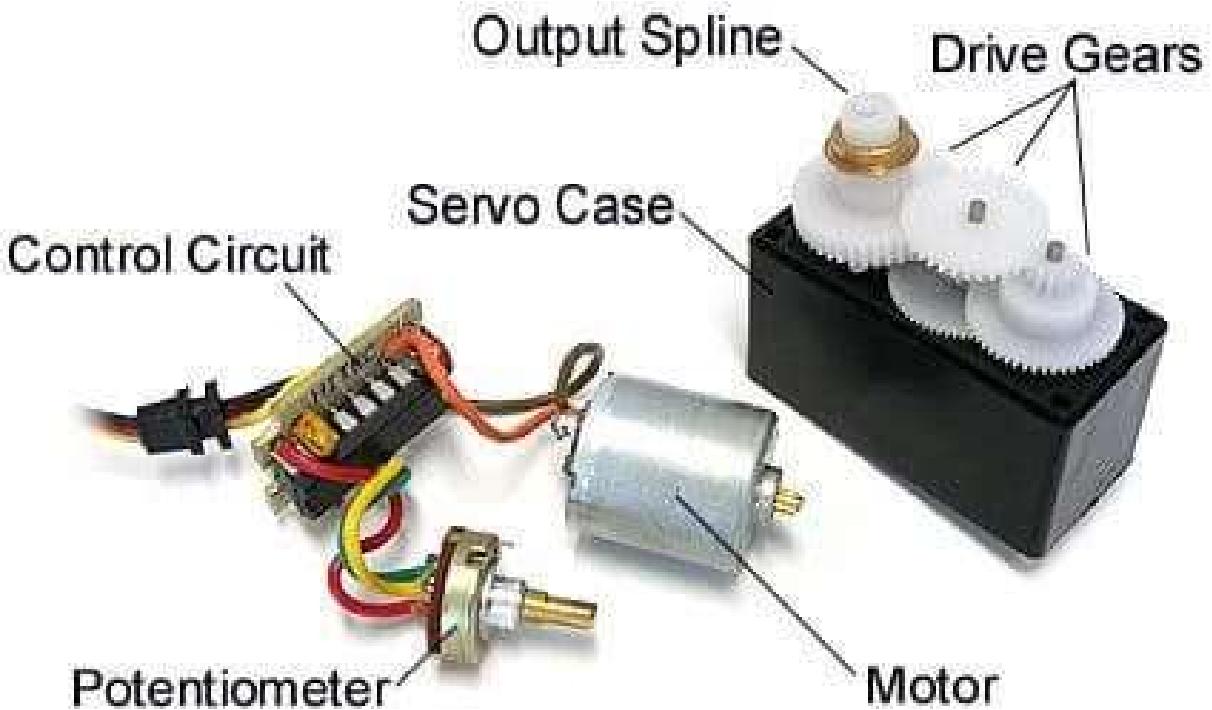


```
void MoveServo_8(int Ang1, int Ang2 , int Degree  
,int DelayTime){  
    for (pos8 = Ang1; pos8 <= Ang2; pos8  
+=Degree) {  
        servo_8.write(pos8);  
        delay(DelayTime); // Wait for 15 millisecond(s)  
    };  
}
```

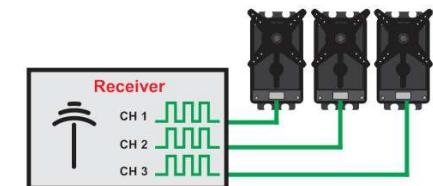
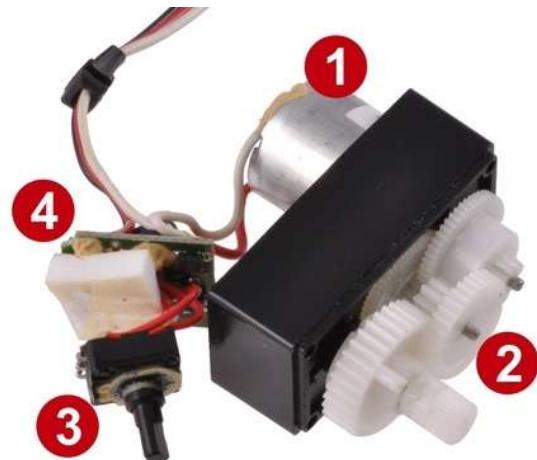
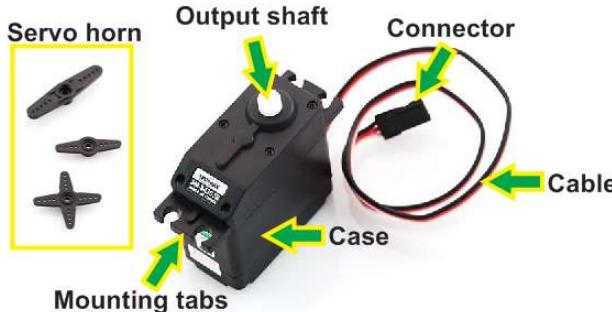
```
void MoveServo_8neg(int Ang1, int Ang2 , int  
Degree ,int DelayTime){  
    for (int pos8  = Ang1; pos8 >= Ang2; pos8 -  
=Degree) {  
        servo_8.write(pos8);  
        delay(DelayTime); // Wait for 15 millisecond(s)  
    };  
}
```

```
void MoveServo_9(int Ang1, int Ang2 , int Degree  
,int DelayTime){  
    for (pos9 = Ang1; pos9 <= Ang2; pos9  
+=Degree) {  
        servo_9.write(pos9);  
        delay(DelayTime); // Wait for 15 millisecond(s)  
    };  
}
```

```
void MoveServo_9neg(int Ang1, int Ang2 , int  
Degree ,int DelayTime){  
    for (pos9 = Ang1; pos9 >= Ang2; pos9 -=Degree)  
    {  
        servo_9.write(pos9);  
        delay(DelayTime); // Wait for 15 millisecond(s)  
    };  
}
```

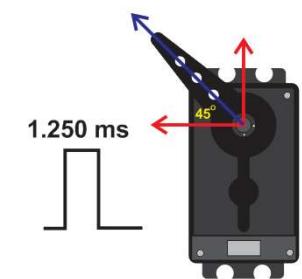
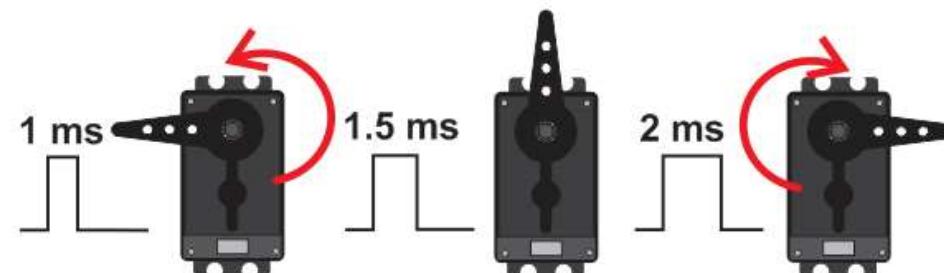


[https://blog.thaieeasyelec.com/example-project-how-to-control-rc-servo-motor-with-arduino/](https://blog.thaieasyelec.com/example-project-how-to-control-rc-servo-motor-with-arduino/)



ถ้าความกว้างของสัญญาณพัลซ์ **1 ms** ตัว Servo Motor จะหมุนไปทางด้วยขี้ยวนสุด

ถ้าความกว้างของสัญญาณพัลซ์ **2 ms** ตัว Servo Motor จะหมุนไปทางด้วยขวาจนสุด

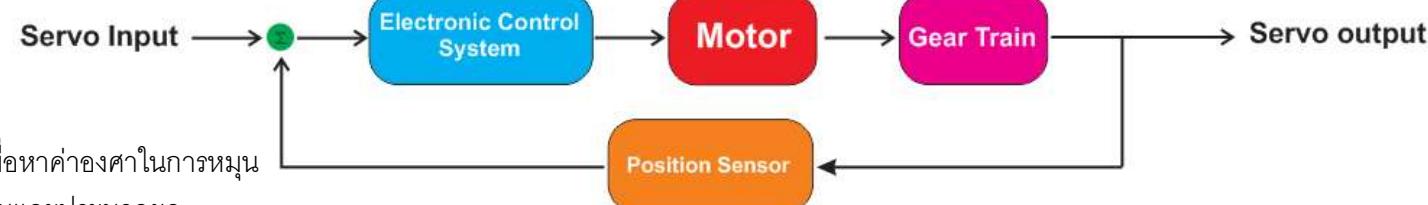


1. Motor เป็นส่วนของตัวมอเตอร์

2. Gear Train หรือ Gearbox เป็นชุดเกียร์ทดแรง

3. Position Sensor เป็นเซ็นเซอร์ตรวจจับตำแหน่งเพื่อหาค่าองศาในการหมุน

4. Electronic Control System เป็นส่วนที่ควบคุมและประมวลผล



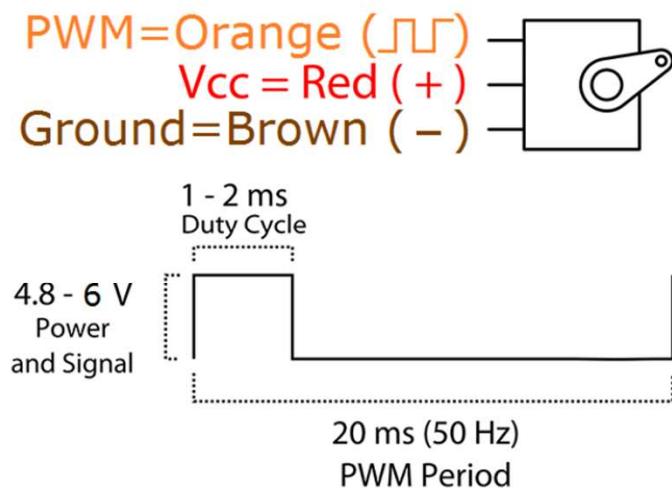
MG90S servo, Metal gear with one bearing

Tiny and lightweight with high output power, this tiny servo is perfect for RC Airplane, Helicopter, Quadcopter or Robot. This servo has *metal gears* for added strength and durability.

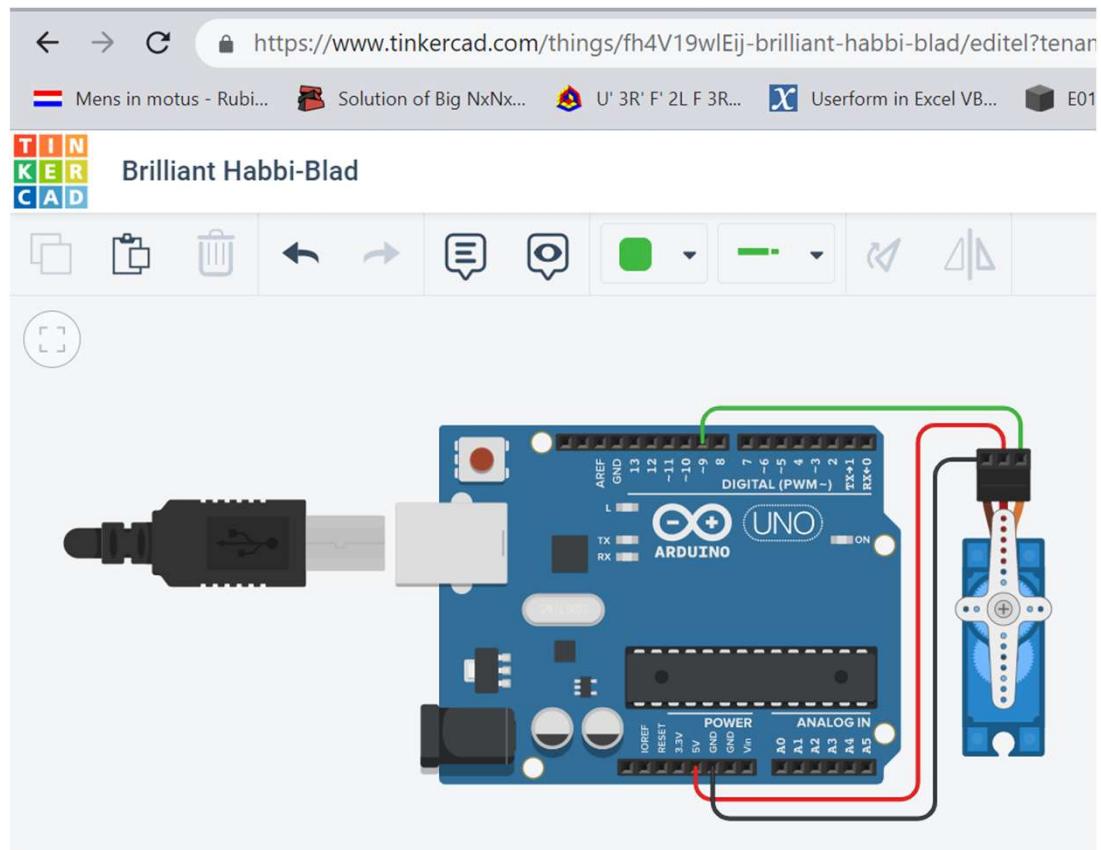
Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but *smaller*. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.

Specifications

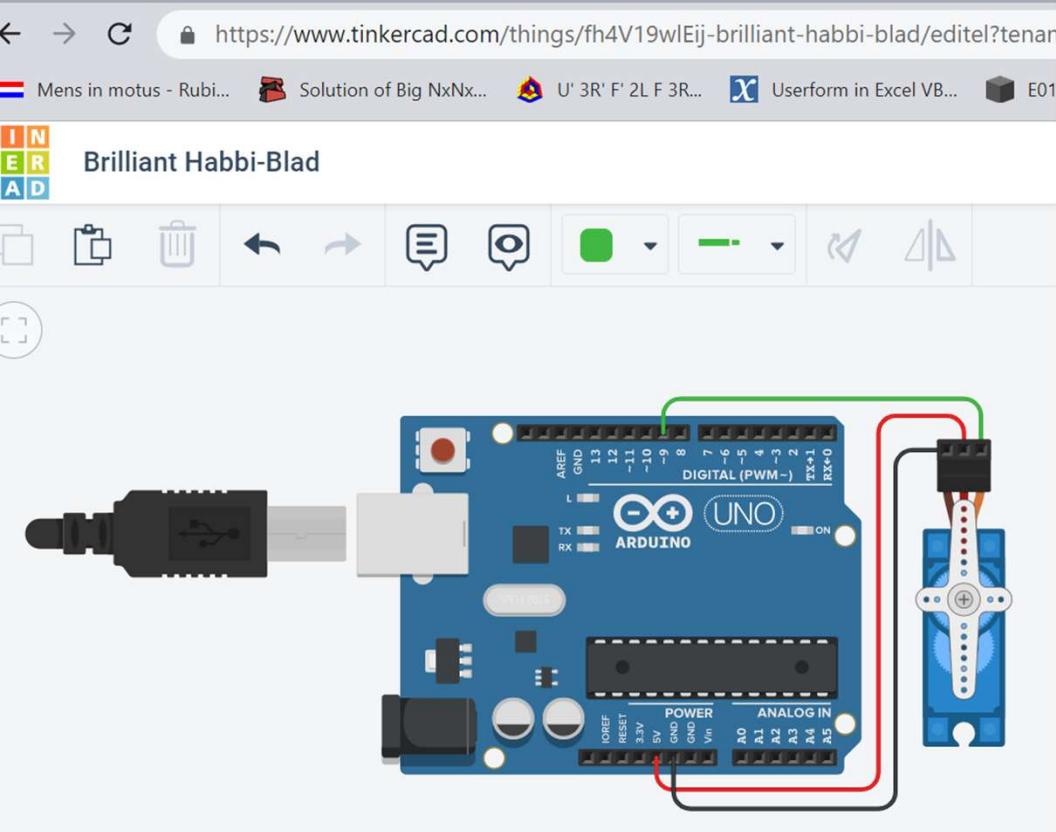
- Weight: 13.4 g
- Dimension: 22.5 x 12 x 35.5 mm approx.
- Stall torque: 1.8 kgf·cm (4.8V), 2.2 kgf·cm (6 V)
- Operating speed: 0.1 s/60 degree (4.8 V), 0.08 s/60 degree (6 V)
- Operating voltage: 4.8 V - 6.0 V
- Dead band width: 5 μ s



Position "0" (1.5 ms pulse) is middle, "90" (~2 ms pulse) is all the way to the right, "-90" (~1 ms pulse) is all the way to the left.



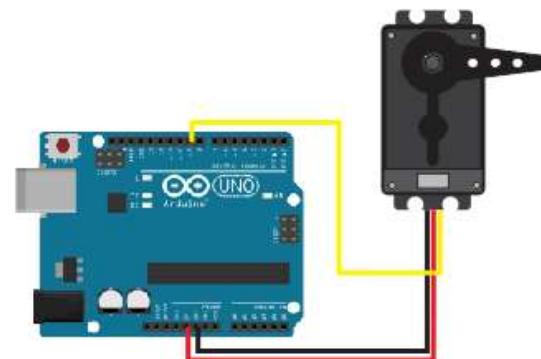
```
forever
  comment sweep the servo from 0 to 180 degrees in steps...
  count up by 1 for pos from 0 to 180 do
    comment tell servo to go to position in variable 'pos'
    rotate servo on pin 9 to pos degrees
    comment wait 15 ms for servo to reach the position
    wait 15 milliseconds
  count down by 1 for pos from 180 to 0 do
    comment tell servo to go to position in variable 'pos'
    rotate servo on pin 9 to pos degrees
    comment wait 15 ms for servo to reach the position
    wait 15 milliseconds
```

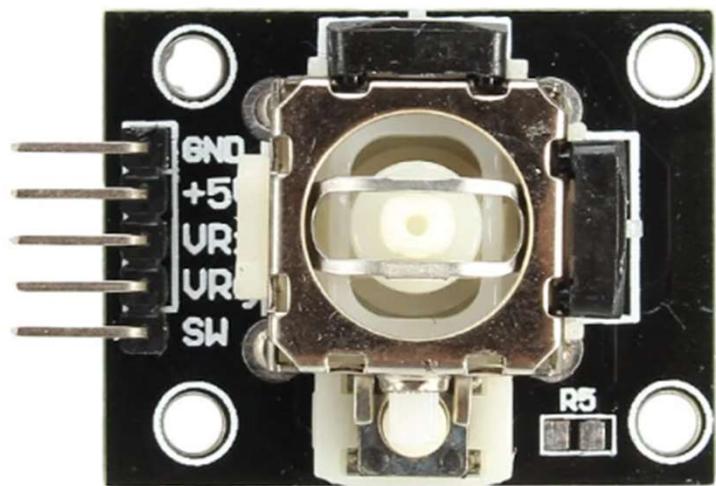
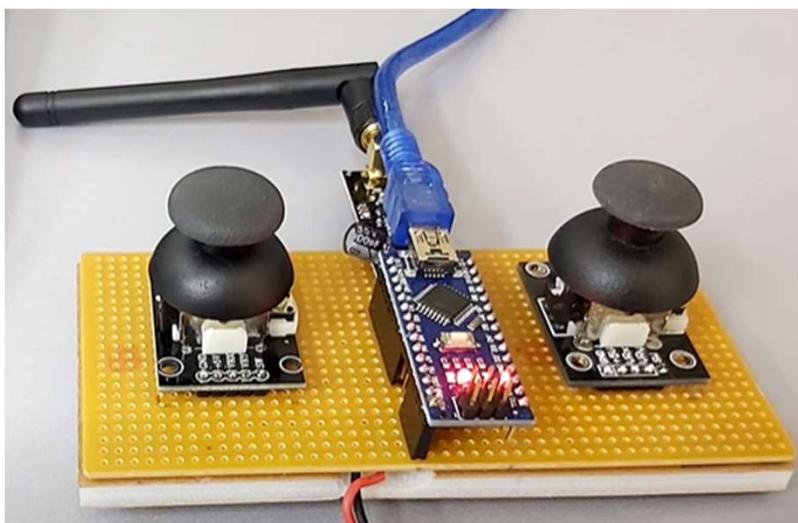
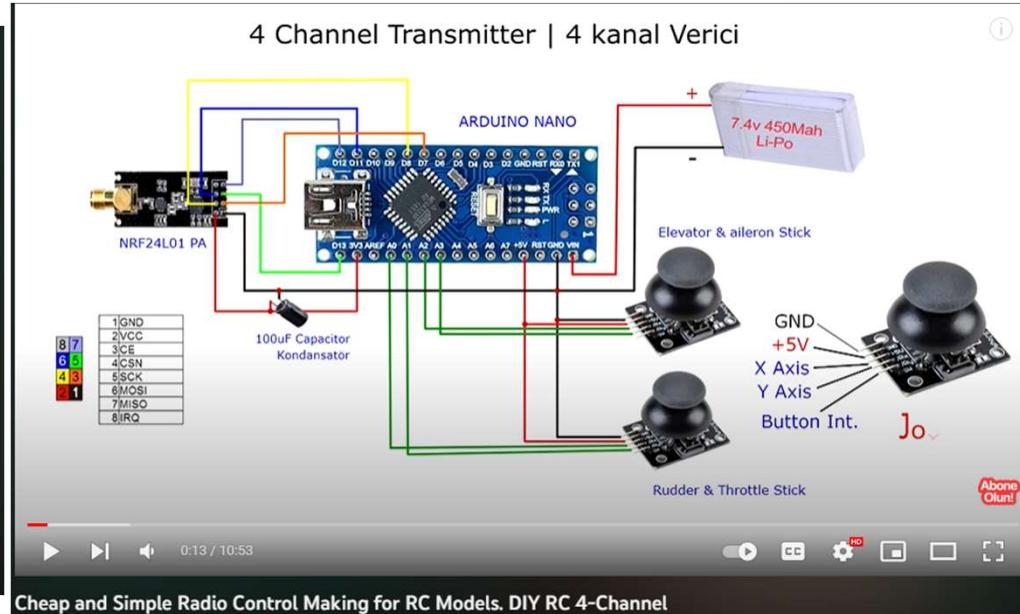
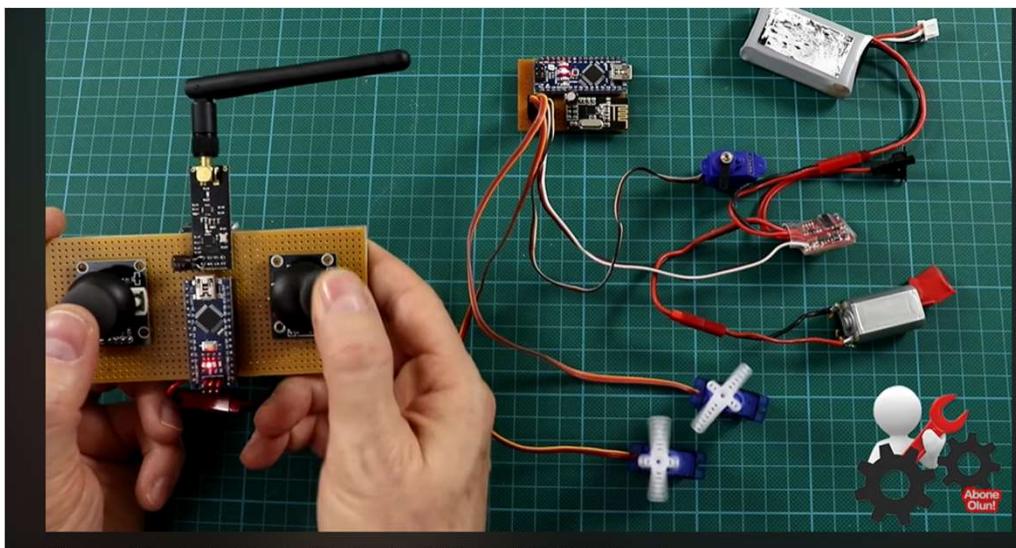


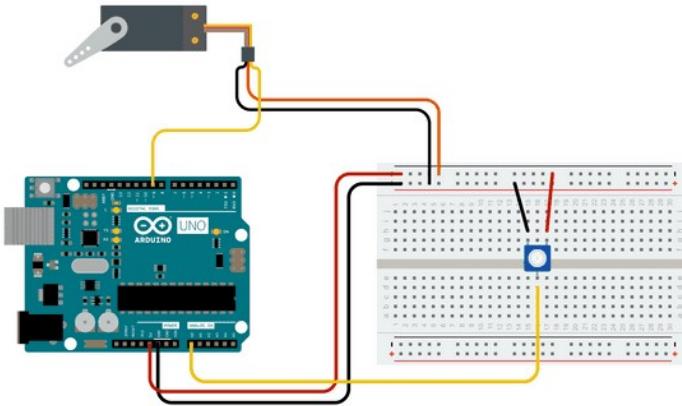
```
#include <Servo.h>
Servo myservo;
void setup()
{
    myservo.attach(9);
}
void loop()
{
    myservo.write(0);
    delay(1000);
    myservo.write(90);
    delay(1000);
    myservo.write(180);
    delay(1000);
}
```

myservo.write(0);
delay(1000);

Servo Motor จะหมุนไปที่ตำแหน่ง 0 องศา และหยุดเป็นเวลา 1 วินาที







Potentiometer ตัวด้านบนแบบปรับค่าได้



```
#include <Servo.h>
```

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

```
int potpin = 0; // analog pin used to connect the potentiometer
int val; // variable to read the value from the analog pin
```

```
void setup() {
```

```
    // attaches the servo on pin 9 to the servo object
    myservo.attach(9);
```

```
}
```

```
void loop() {
```

```
    val = analogRead(potpin); // reads the value of the
    potentiometer (value between 0 and 1023)
```

```
    // scale it to use it with the servo (value between 0 and 180)
```

```
    val = map(val, 0, 1023, 0, 180);
```

```
    // sets the servo position according to the scaled value
```

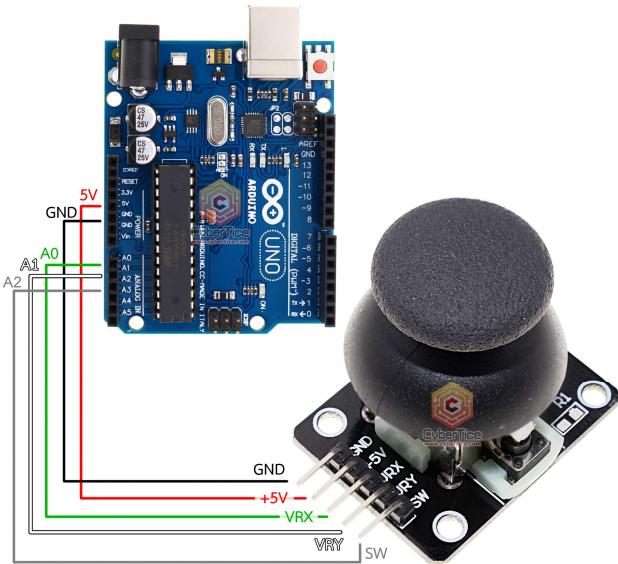
```
    myservo.write(val);
```

```
    delay(15);
```

```
    // waits for the servo to get there
```

```
}
```

analog joystick KY-023



ปุ่มกด KY-023



VRY → "analog output of 4-axis"

Arduino's A/D converter reads:
0 → moved top
1023 → moved down



```
int X = A0;
int XValue = 0;
int Y = A1;
int YValue = 0;
int SW = A2;
int SWValue = 0;

void setup() {
    Serial.begin(9600);
}

void loop() {
    XValue = analogRead(X);
    YValue = analogRead(Y);
    SWValue = analogRead(SW);
    Serial.print("XValue: ");
    Serial.print(XValue);
    Serial.print("\t");
    Serial.print("YValue: ");
    Serial.print(YValue);
    Serial.print("\t");
    Serial.print("SWValue: ");
    Serial.println(SWValue);
    delay(200);
}
```

ในดูดูนี้ประกอบไปด้วย potentiometer ขนาด 10kOhm จำนวน 2 ตัววางตั้งจากกันเพื่อระบุตำแหน่งแกน X และ Y และปุ่มกด

XValue	YValue	SWValue
493	496	315
494	497	284
494	497	231
494	497	319
495	497	210
494	497	290
494	498	330
494	497	337
494	497	380
494	497	402
1019	647	523
1021	824	683
1022	831	691
1022	682	620
1022	659	589
1022	657	567
1022	659	554

Autoscroll Show timestamp

<https://www.youtube.com/watch?v=qDq7vm-7Xs4>



COM7

```
XValue: 495 YValue: 497 SWValue: 777  
XValue: 494 YValue: 497 SWValue: 539  
XValue: 491 YValue: 494 SWValue: 501  
XValue: 493 YValue: 495 SWValue: 685  
XValue: 494 YValue: 497 SWValue: 887  
XValue: 495 YValue: 497 SWValue: 890  
XValue: 495 YValue: 498 SWValue: 889  
XValue: 495 YValue: 498 SWValue: 880  
XValue: 495 YValue: 496 SWValue: 892  
XValue: 495 YValue: 498 SWValue: 878  
XValue: 495 YValue: 498 SWValue: 880  
XValue: 495 YValue: 498 SWValue: 736  
XValue: 495 YValue: 498 SWValue: 602  
XValue: 0 YValue: 497 SWValue: 378  
XValue: 0 YValue: 497 SWValue: 117  
XValue: 0 YValue: 497 SWValue: 112  
XValue: 0 YValue: 496 SWValue: 104
```

Autoscroll Show timestamp

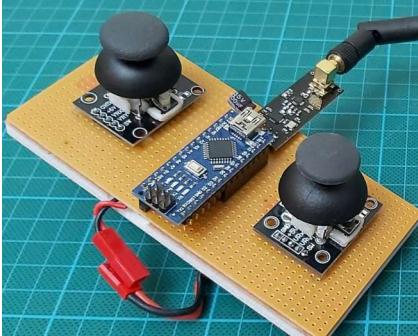


COM7

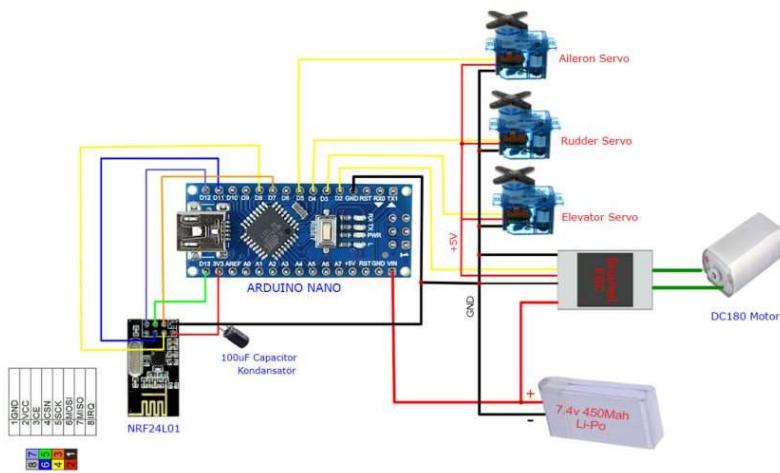
```
XValue: 494 YValue: 497 SWValue: 264  
XValue: 495 YValue: 497 SWValue: 374  
XValue: 495 YValue: 497 SWValue: 552  
XValue: 495 YValue: 498 SWValue: 687  
XValue: 495 YValue: 497 SWValue: 634  
XValue: 495 YValue: 498 SWValue: 752  
XValue: 494 YValue: 497 SWValue: 690  
XValue: 494 YValue: 497 SWValue: 808  
XValue: 495 YValue: 1023 SWValue: 1001  
XValue: 495 YValue: 1023 SWValue: 816  
XValue: 495 YValue: 1023 SWValue: 562  
XValue: 494 YValue: 1023 SWValue: 313  
XValue: 494 YValue: 1023 SWValue: 310  
XValue: 495 YValue: 1023 SWValue: 314  
XValue: 494 YValue: 1023 SWValue: 304  
XValue: 495 YValue: 1023 SWValue: 317  
XValue: 495 YValue: 1023 SWValue: 401
```

Autoscroll Show timestamp

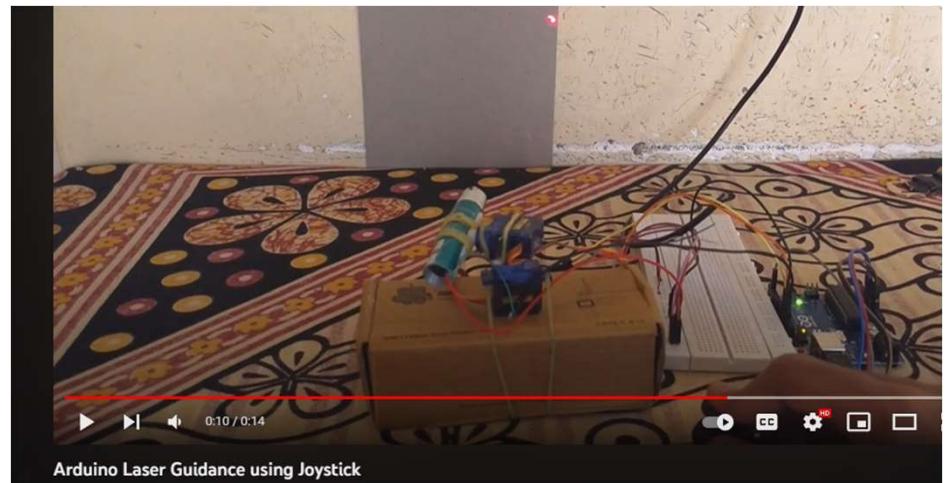
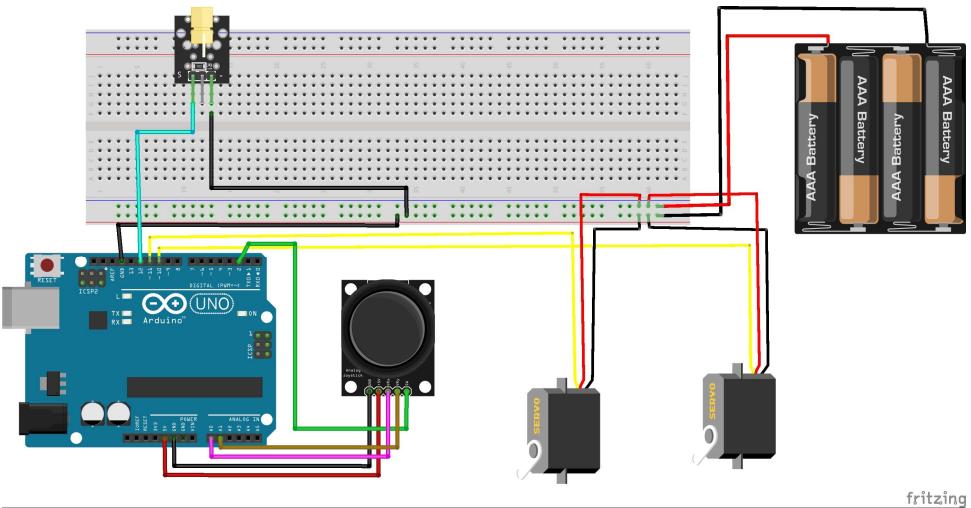
RADIO CONTROL

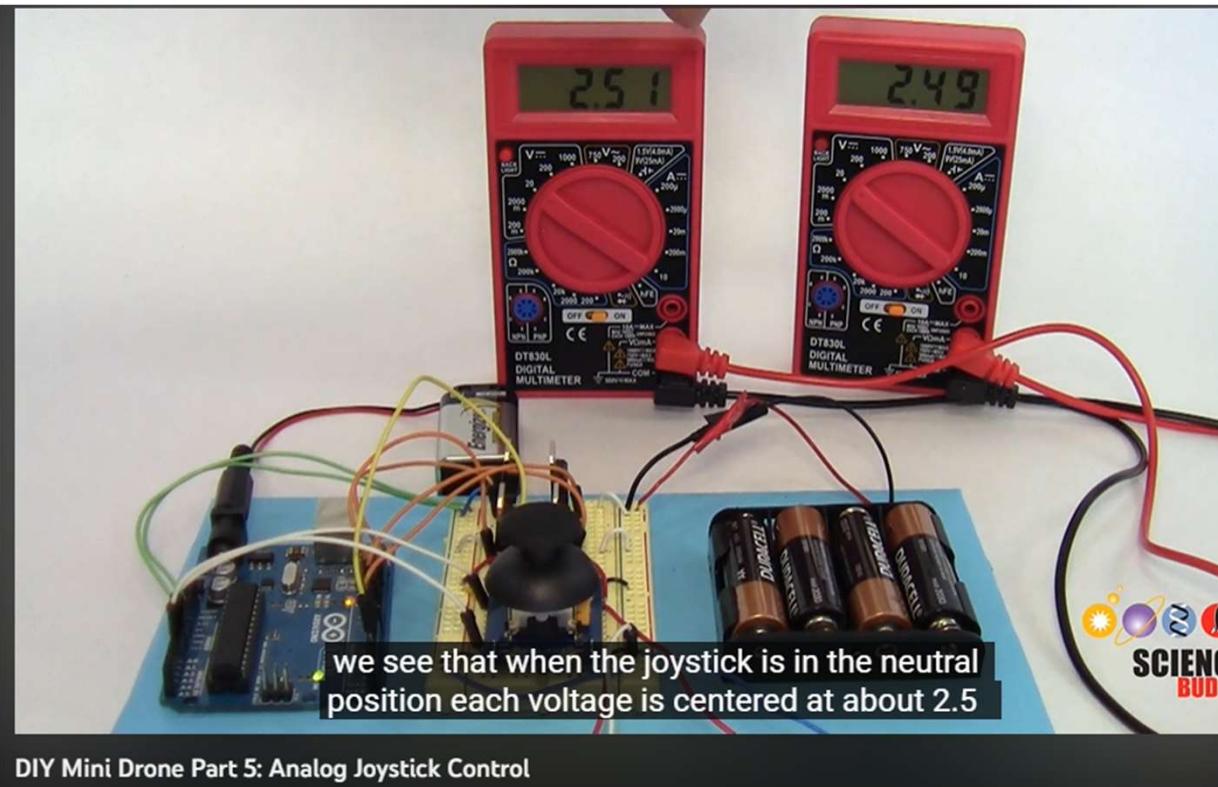


DIY



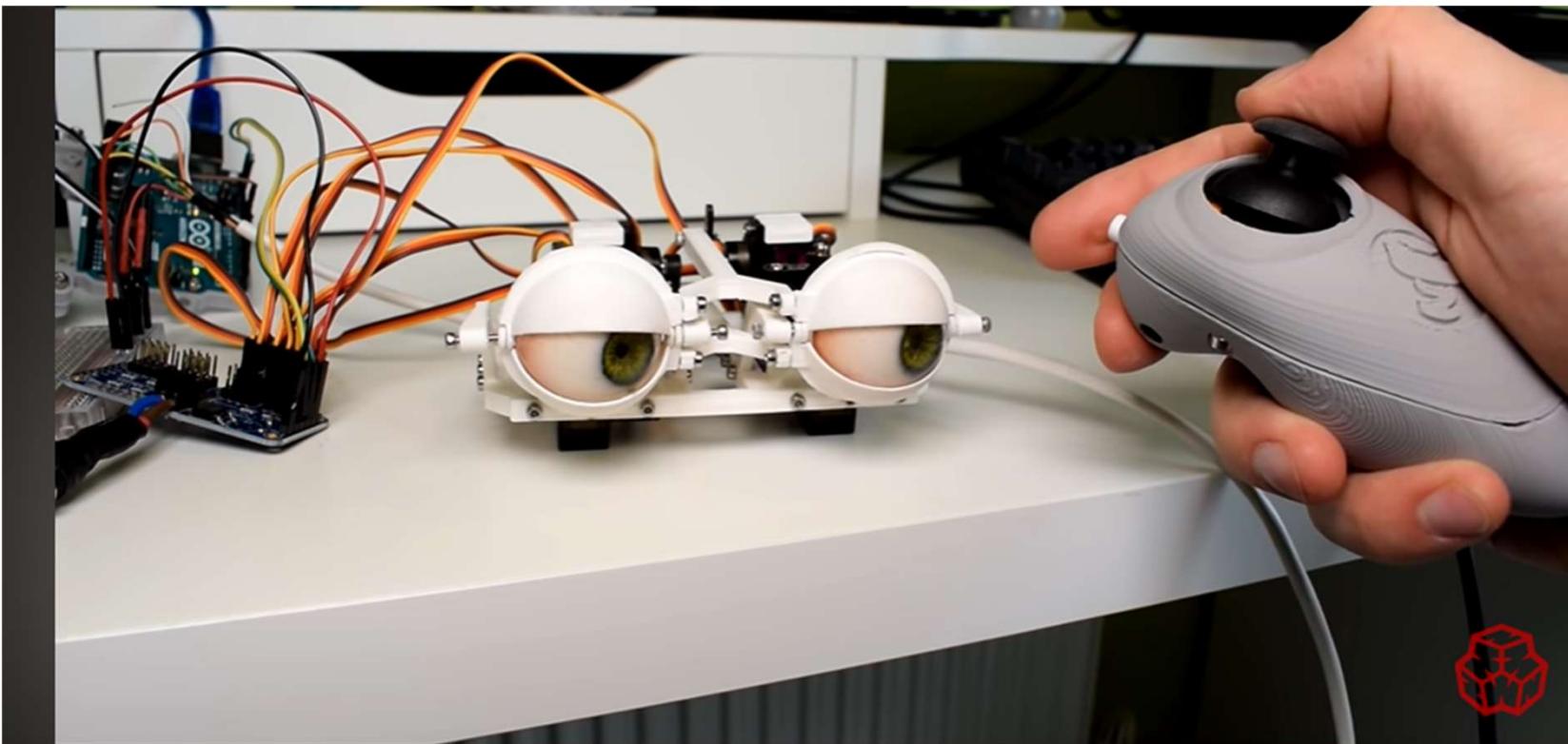
<https://projecthub.arduino.cc/shubhamsantosh99/8e8aacff-ece4-451a-b855-65c24d703a5d>





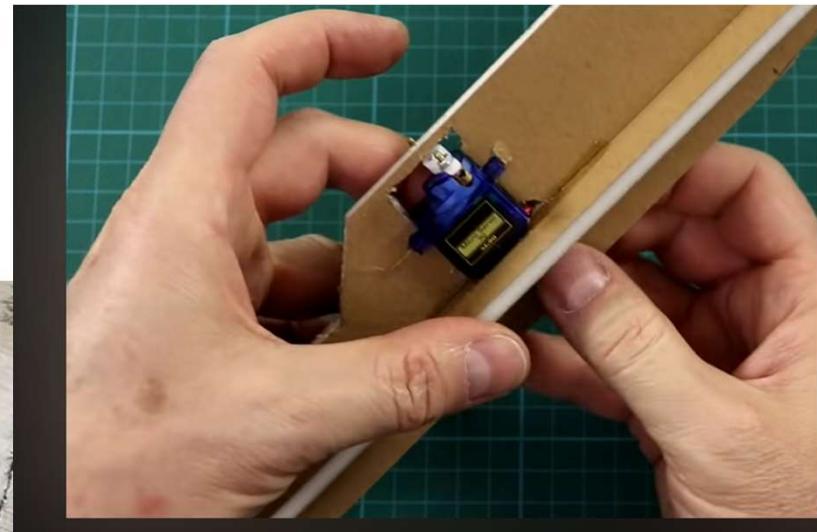
DIY Mini Drone Part 5: Analog Joystick Control

<https://youtu.be/E3MUIPjK-HE?t=95>



How to Make a Compact Animatronic Eye Mechanism with 3D Printing and

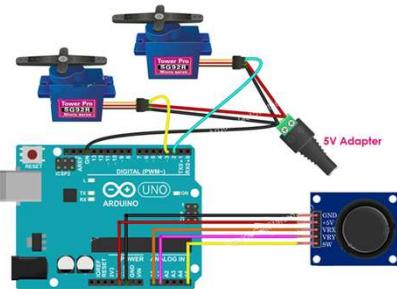
<https://www.youtube.com/watch?v=9SMyBN-B3Vo>

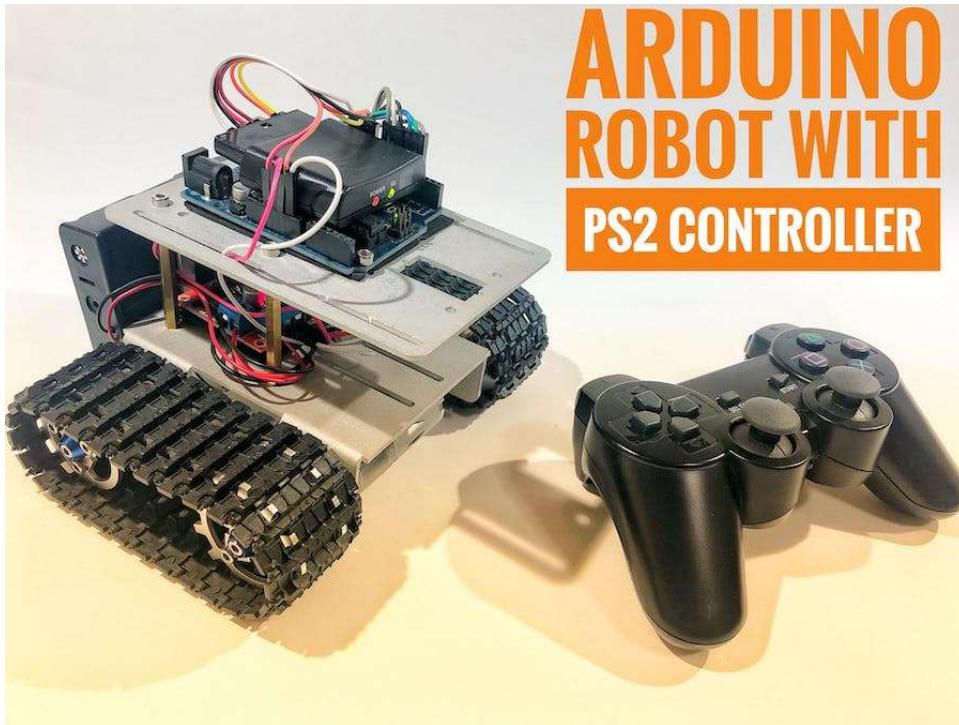


How To Make Simple RC Airplane For Simple Radio Control. DIY RC Aiplane & Arduino RC

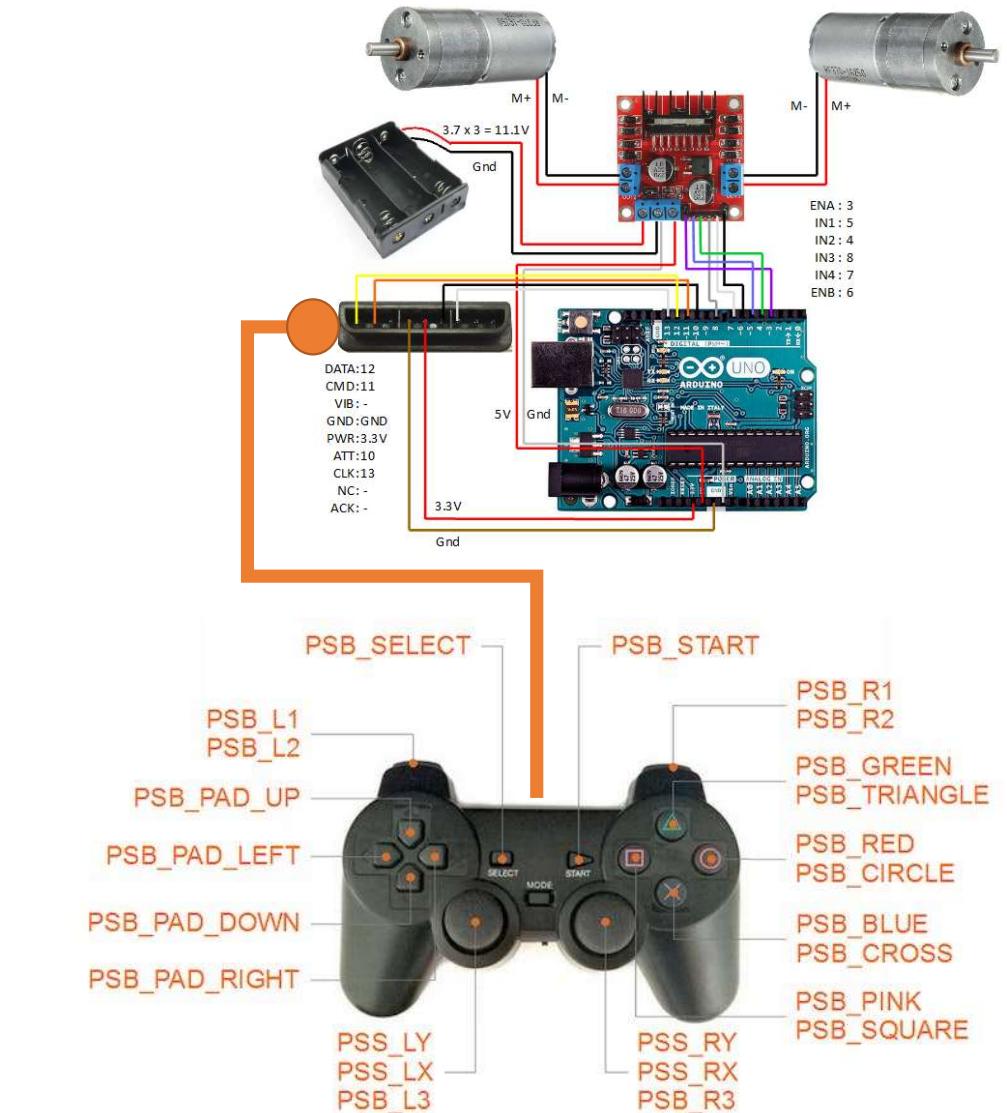
<https://www.circuits-diy.com/joystick-with-servo-motor-arduino-tutorial/>

Joystick Servo Motor Arduino Tutorial





<https://www.hackster.io/igorF2/arduino-robot-with-ps2-controller-playstation-2-joystick-85bddc>





Making a 2400 meters LONG RANGE 8-Channel & Digital Trim Radio Control For RC...

475K views 6 months ago #DiyRC

How to make long range remote control with 100mW NRF24L01 PA LN (E01-ML01DP5) 2.4Ghz module and Arduino. Show more



KendinYap

376K subscribers

Subscribe



7.7K



Comments 544

Please can you make a video on this same transmitter with 0.96" Oled display and arduino nano without...

<https://www.youtube.com/watch?v=RSDFMaanKXM>