
Arduino Workshop

Content

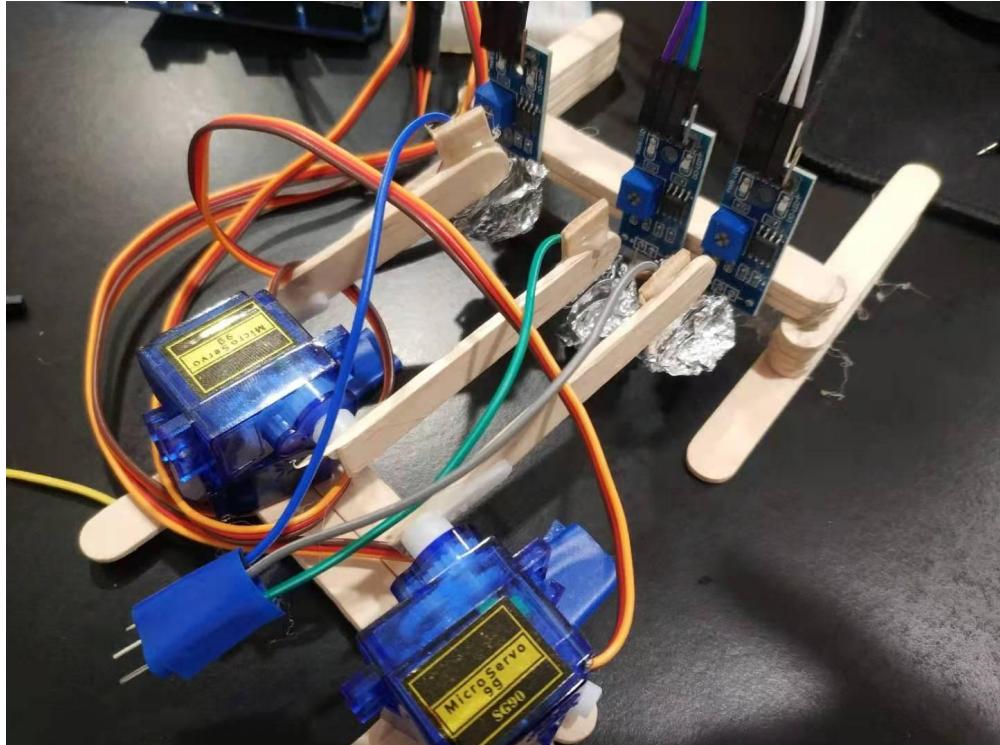
First part:

1. [IDE installation](#)
2. [Board Description & Set up](#)
3. Program Structure
4. I/O function
5. Turn on your first LED
6. Fading LED

Content

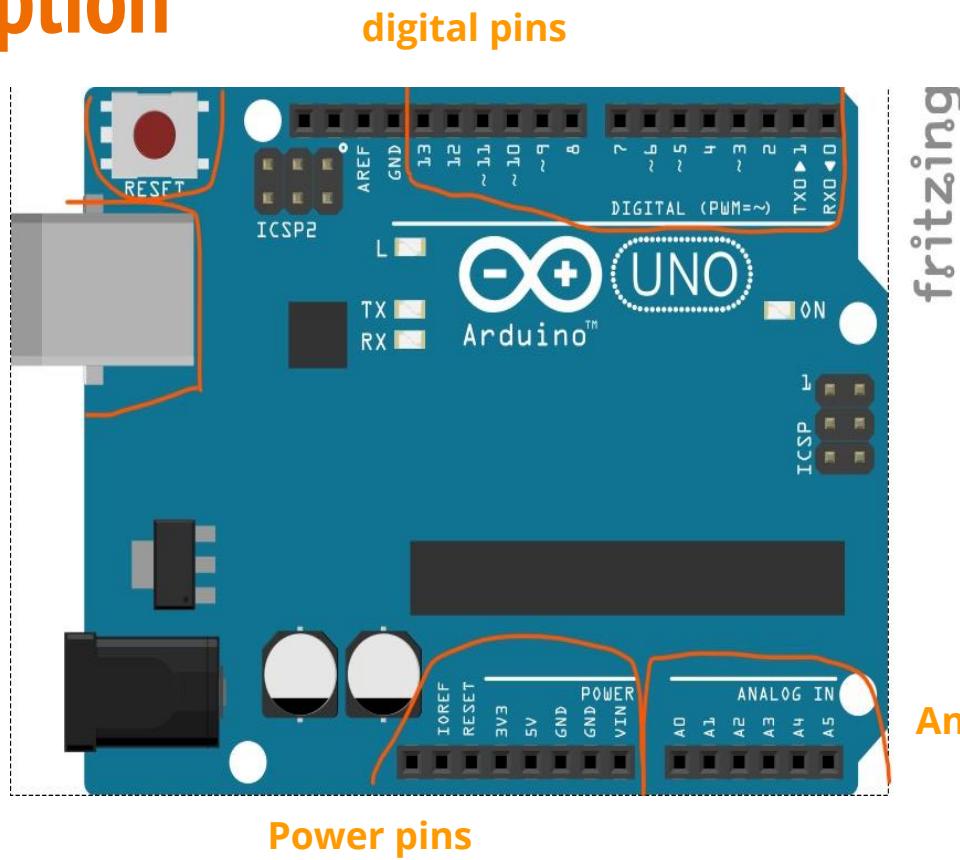
Second part:

1. Servo
2. LDR module
3. mini project



Board description

Power USB



Analog input pins

I/O Function

pinMode() : set pin mode.

pinMode(pin,mode)

Input: pinMode(pin,INPUT)

Output: pinMode(pin,OUTPUT)

digitalWrite(): output HIGH or LOW

digitalWrite(pin,HIGH/LOW)

digitalRead(): read digital input(HIGH or LOW) from pin and return the value (1/0)

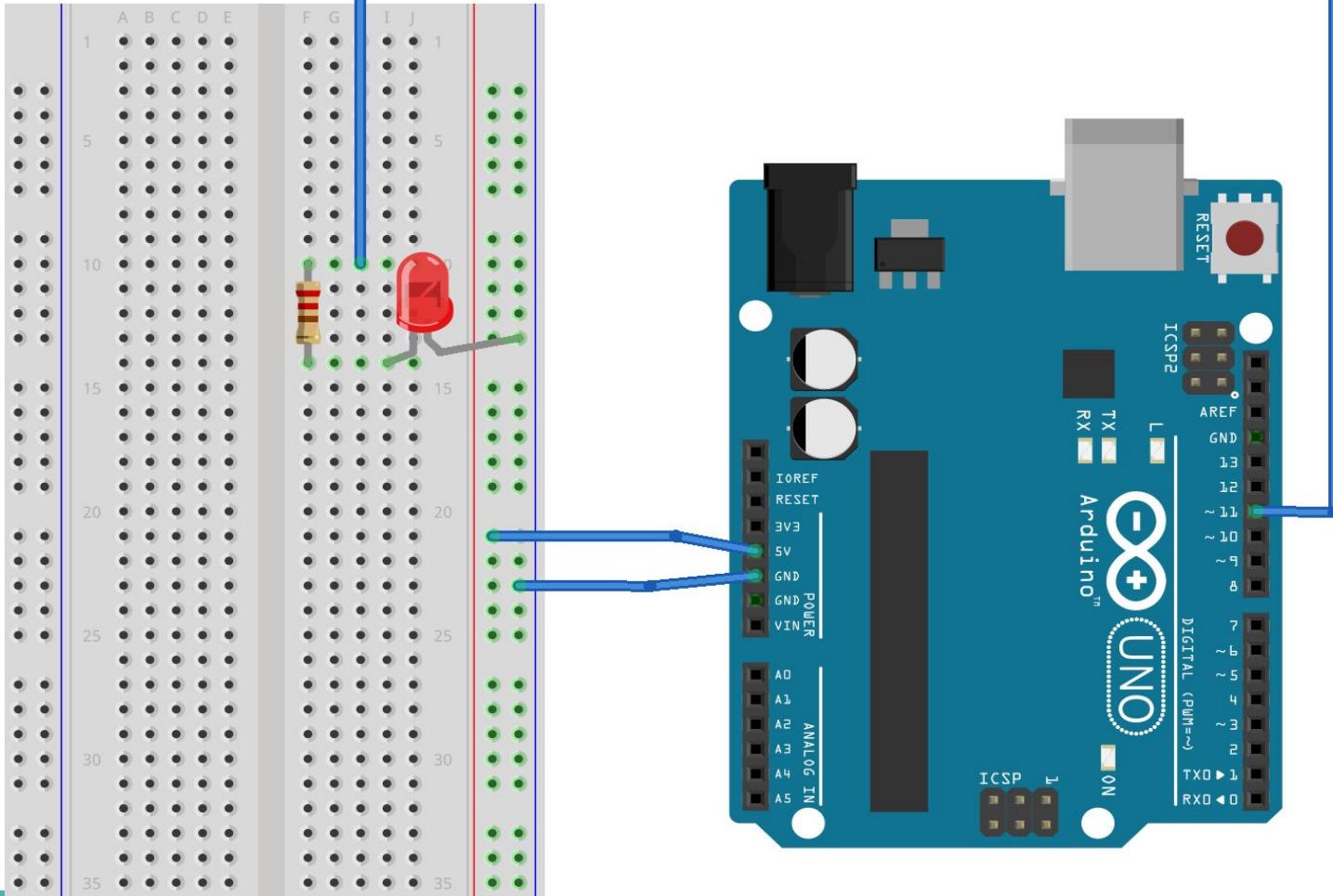
digitalRead(pin)

analogWrite(): output square wave with different duty cycle ([PWM](#))

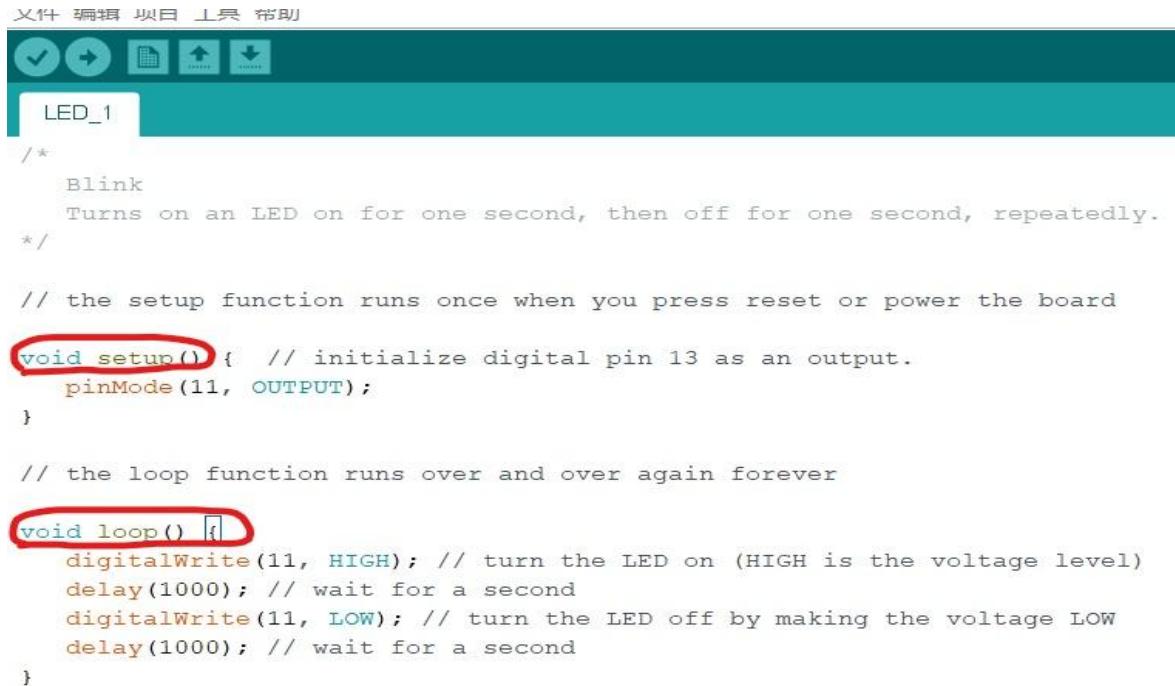
analogWrite(pin,duty cycle(0-255))

analogRead(): read analog input form pin and return the value (0-255)

analogRead(pin)



Program Structure



```
LED_1

/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.
*/

// the setup function runs once when you press reset or power the board

void setup() { // initialize digital pin 13 as an output.
  pinMode(11, OUTPUT);
}

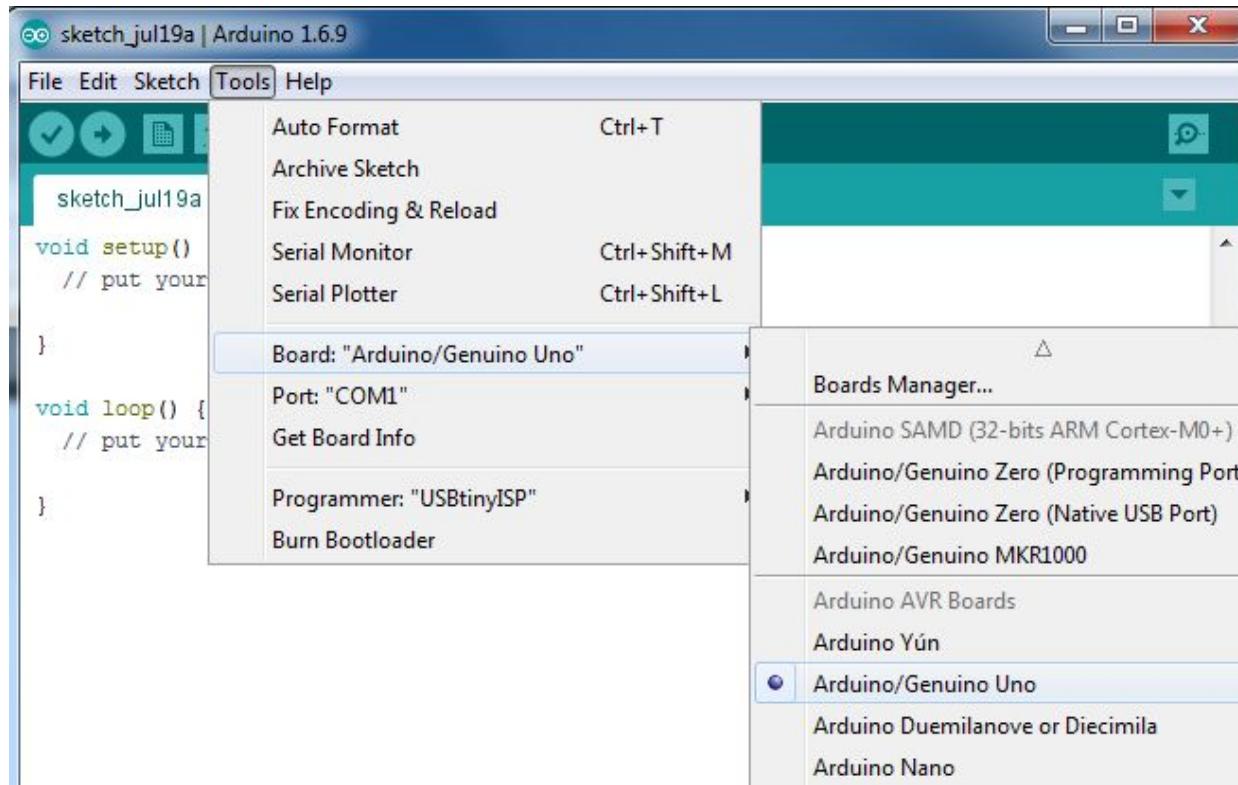
// the loop function runs over and over again forever

void loop() {
  digitalWrite(11, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(11, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

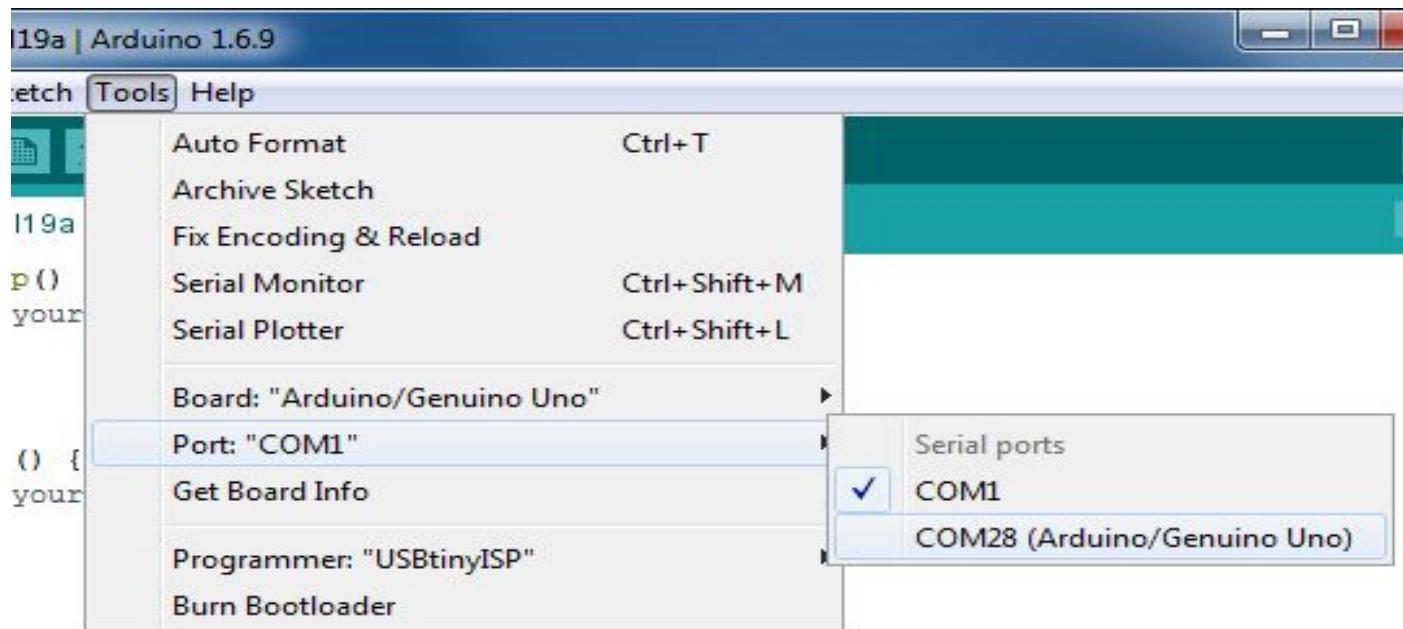
How to upload your first code

1. Connect your Arduino using the USB cable.
2. Choose Tools→Board→Arduino Uno to find your board in the Arduino menu.
3. Choose the correct serial port for your board.
4. Click the Upload button.

How to upload your first code



How to upload your first code



Code

文件 编辑 项目 工具 帮助



```
LED_1

/*
Blink
Turns on an LED on for one second, then off for one second, repeatedly.
*/

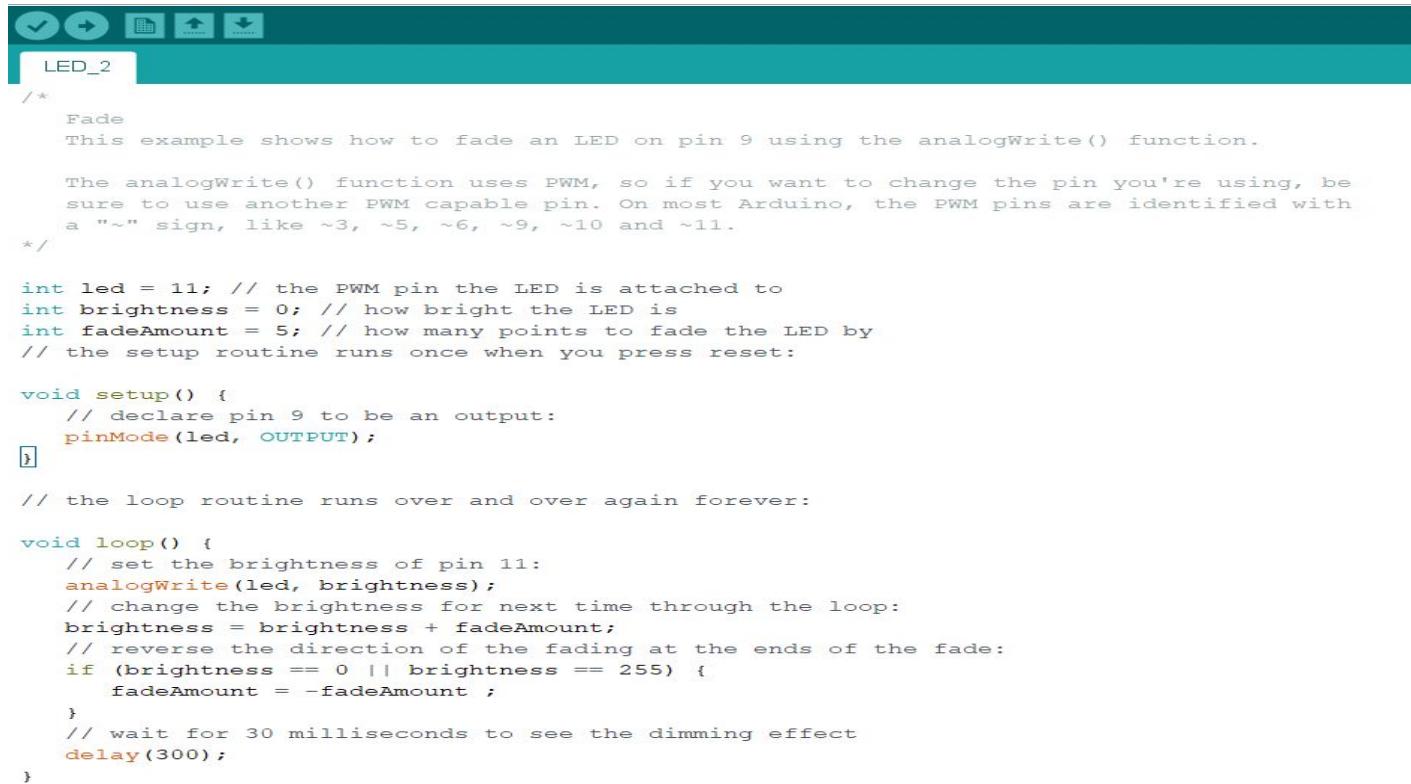
// the setup function runs once when you press reset or power the board

void setup() { // initialize digital pin 13 as an output.
  pinMode(11, OUTPUT);
}

// the loop function runs over and over again forever

void loop() {
  digitalWrite(11, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(11, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

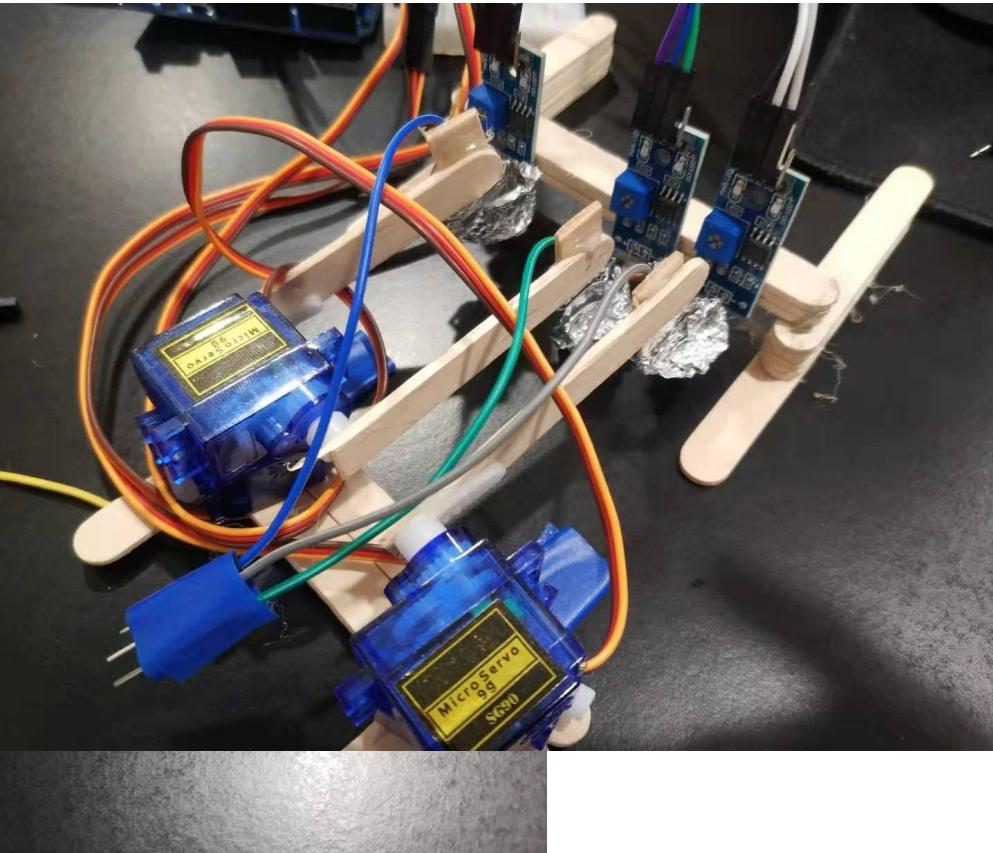
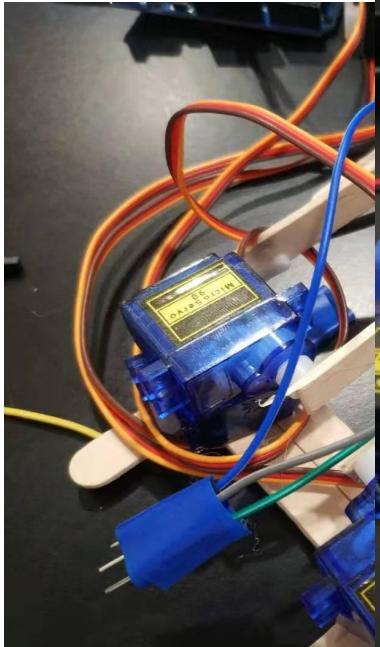
Code

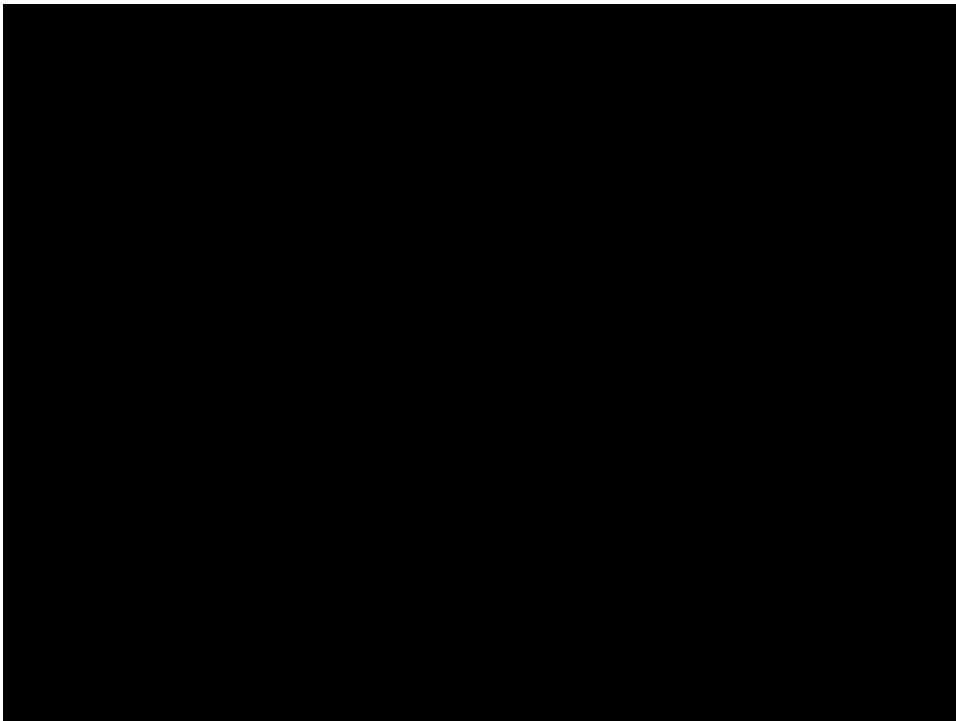


The screenshot shows the Arduino IDE interface with the title bar "LED_2". The code area contains the following:

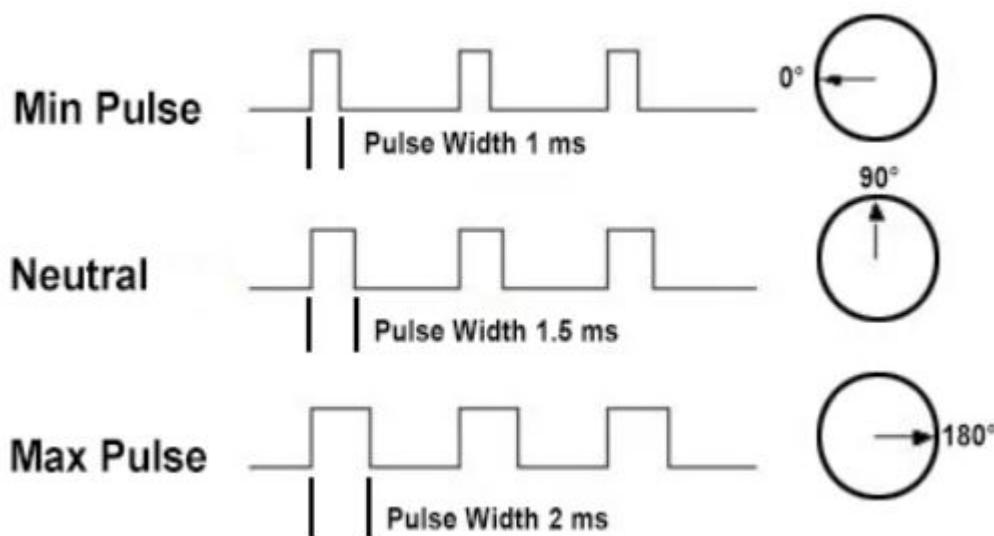
```
/*  
Fade  
This example shows how to fade an LED on pin 9 using the analogWrite() function.  
  
The analogWrite() function uses PWM, so if you want to change the pin you're using, be  
sure to use another PWM capable pin. On most Arduino, the PWM pins are identified with  
a "~" sign, like ~3, ~5, ~6, ~9, ~10 and ~11.  
*/  
  
int led = 11; // the PWM pin the LED is attached to  
int brightness = 0; // how bright the LED is  
int fadeAmount = 5; // how many points to fade the LED by  
// the setup routine runs once when you press reset:  
  
void setup() {  
    // declare pin 9 to be an output:  
    pinMode(led, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
  
void loop() {  
    // set the brightness of pin 11:  
    analogWrite(led, brightness);  
    // change the brightness for next time through the loop:  
    brightness = brightness + fadeAmount;  
    // reverse the direction of the fading at the ends of the fade:  
    if (brightness == 0 || brightness == 255) {  
        fadeAmount = -fadeAmount;  
    }  
    // wait for 30 milliseconds to see the dimming effect  
    delay(300);  
}
```

Mini Project





9g Servo



Brown: GND

Red: VCC

Orange: input

sketch_

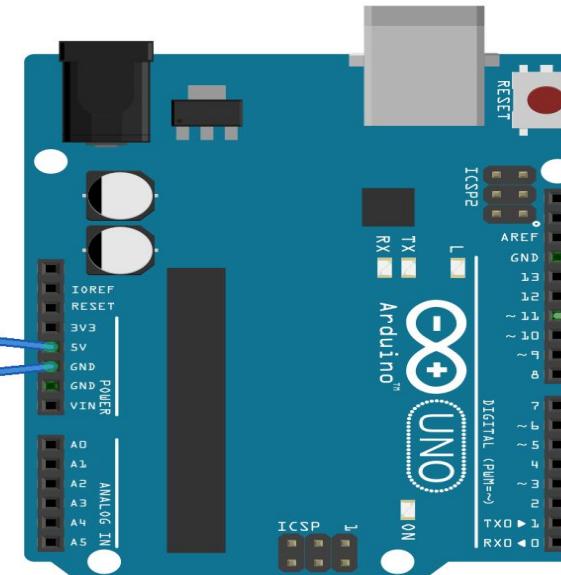
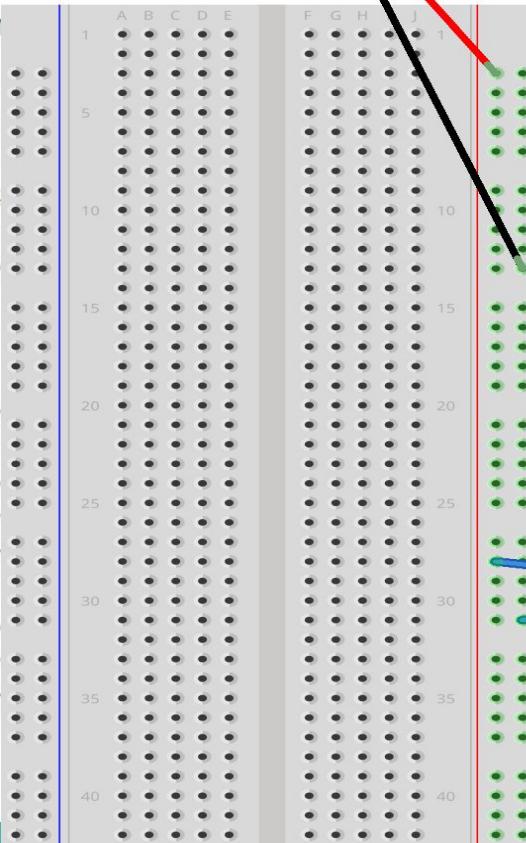
文件 编辑



sketch_r

```
#include  
Servo mys  
int pos;  
void setup  
// put  
myservo  
{  
}
```

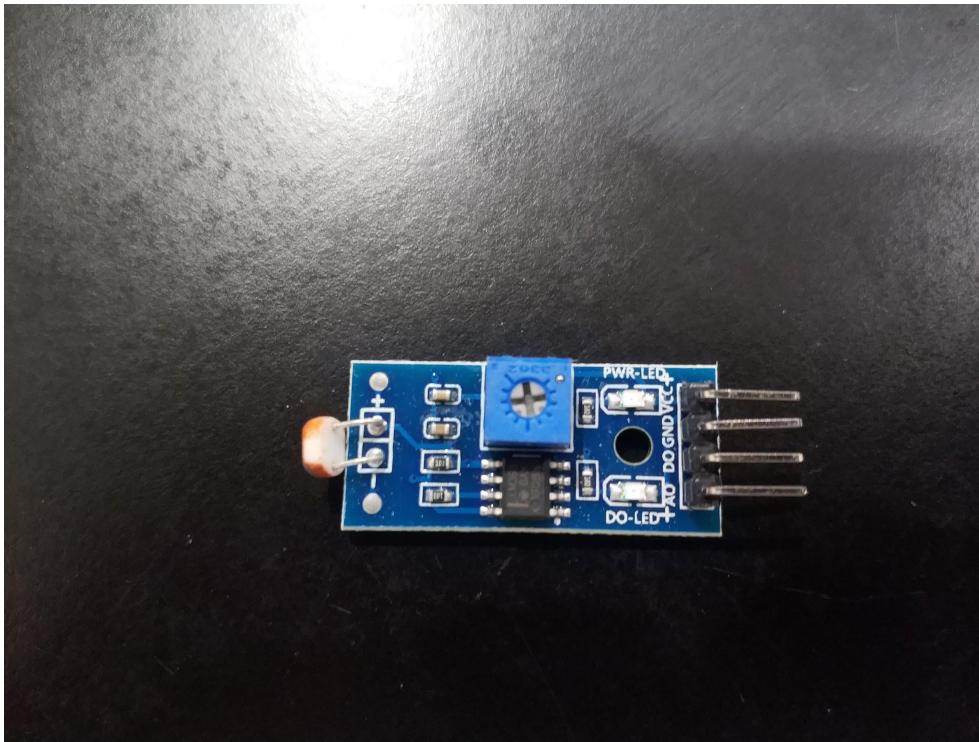
```
void loop  
// put  
for (po  
myser  
delay  
}  
for (po  
myser  
delay  
}  
}
```



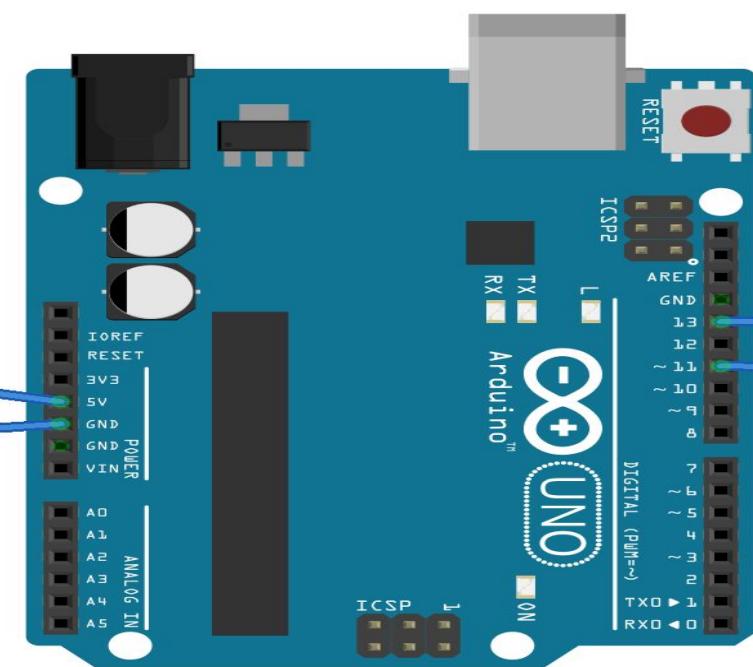
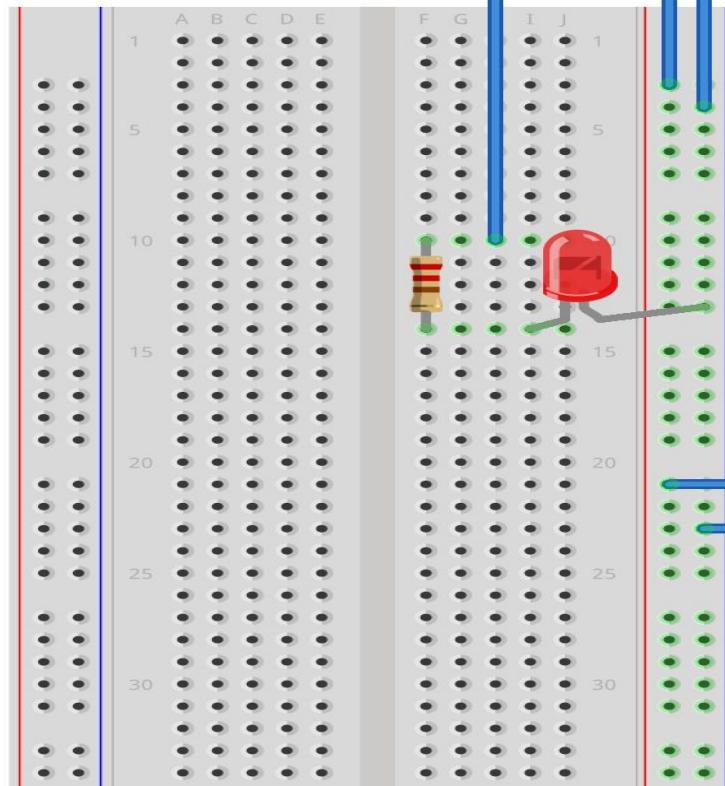
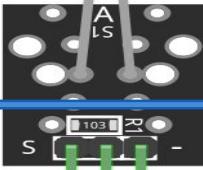
variable 'pos'
reach the position

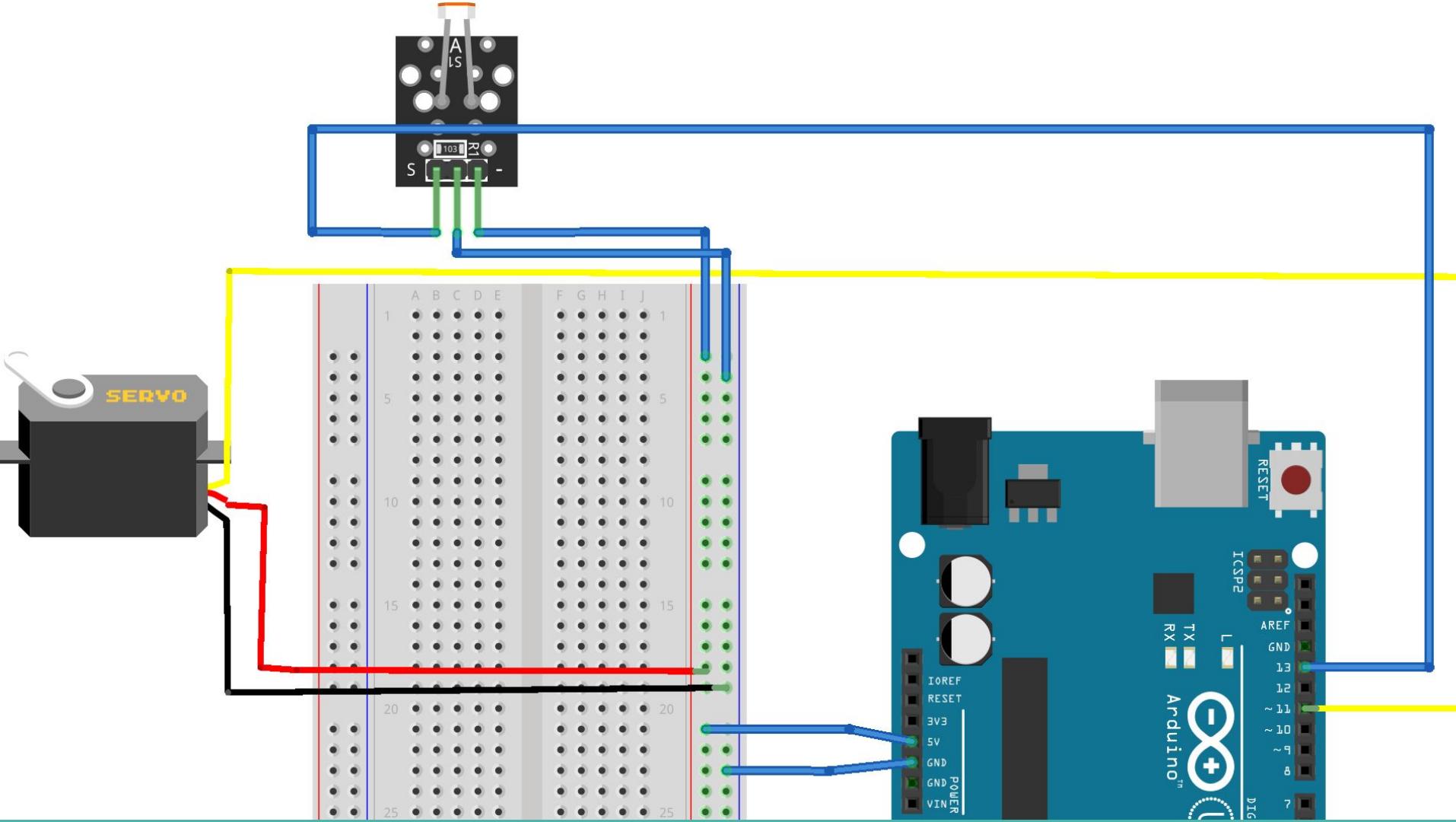
variable 'pos'
reach the position

LDR Module



AO: Analog output
DO: Digital output





Mini Project



take a long stick and cut it into
two parts

Mini Project



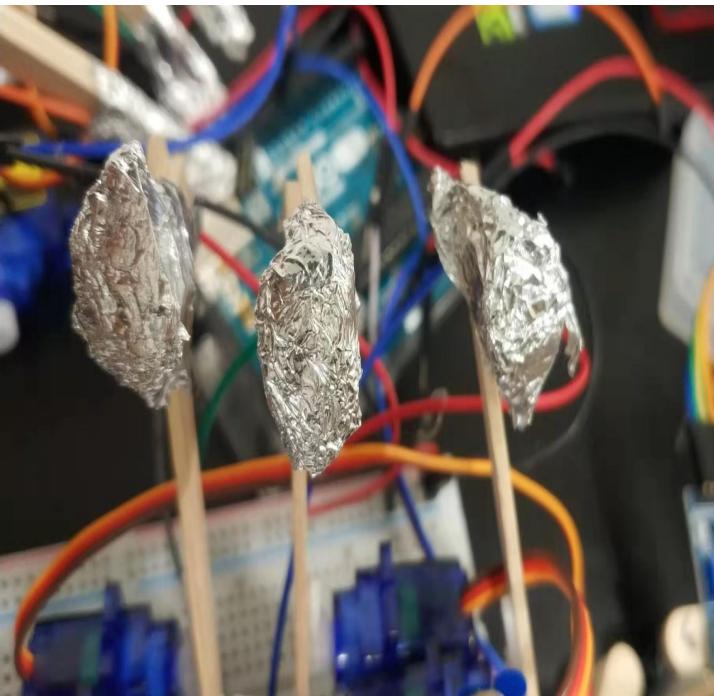
glue them together
like this

Mini Project



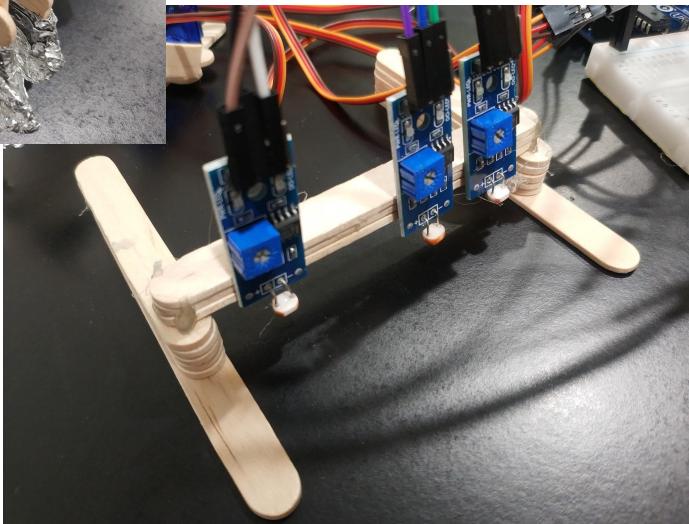
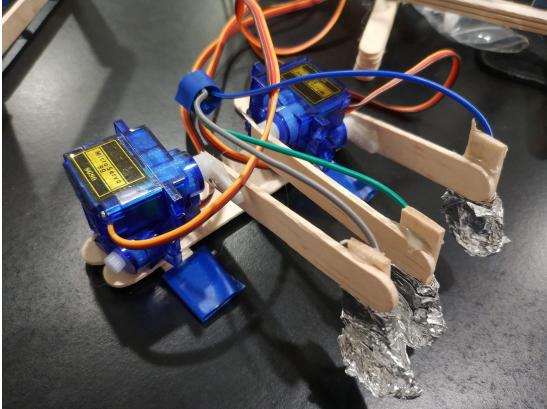
put a male to male jumper
wire on the one end

Mini Project



1. cover it with aluminium foil tightly
2. make sure the contact area is large enough

Mini Project

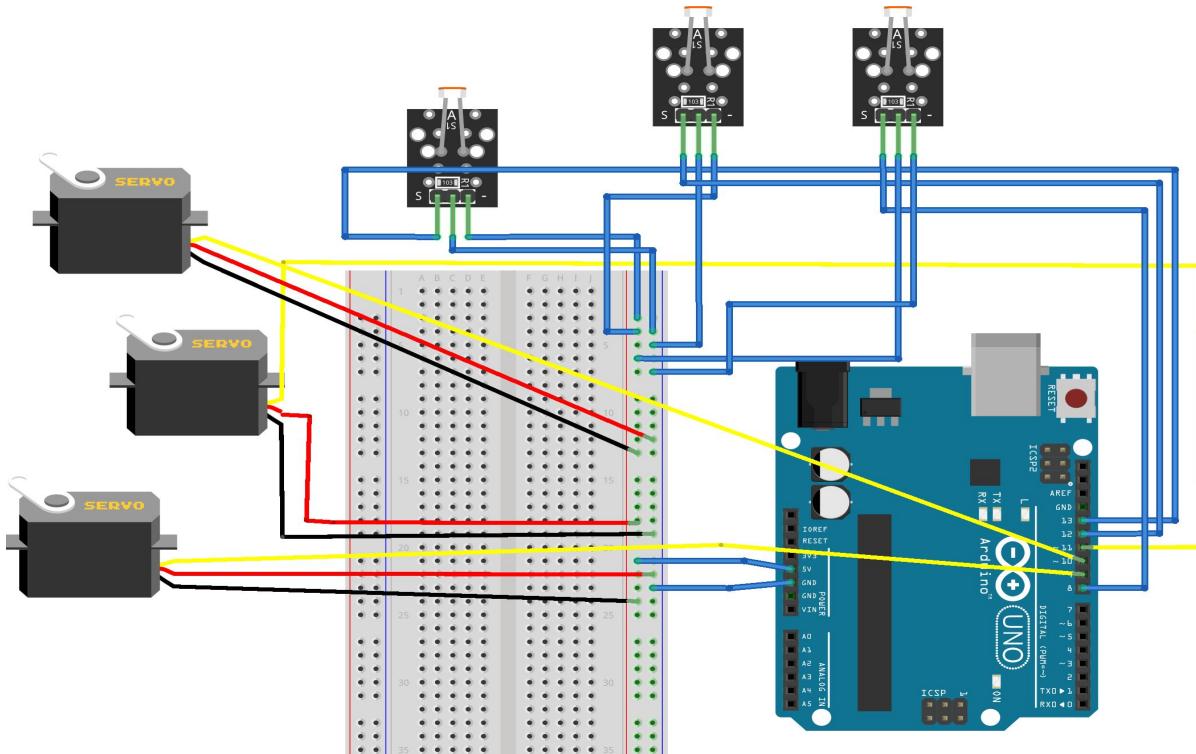


the shape is for reference

Tips:

1. make the arm short and close to screen
2. make sure the base are stable and servos are glued tightly
3. make the LDR close to your screen
4. make sure the shape fit your phone size

Mini Project



Servo connected to 11,10,9
LDR connected to 13,12,8

Mini Project

Servos connected to 11 and LDR connected 13 are one group

Servos connected to 10 and LDR connected 12 are one group

Servos connected to 9 and LDR connected 8 are one group