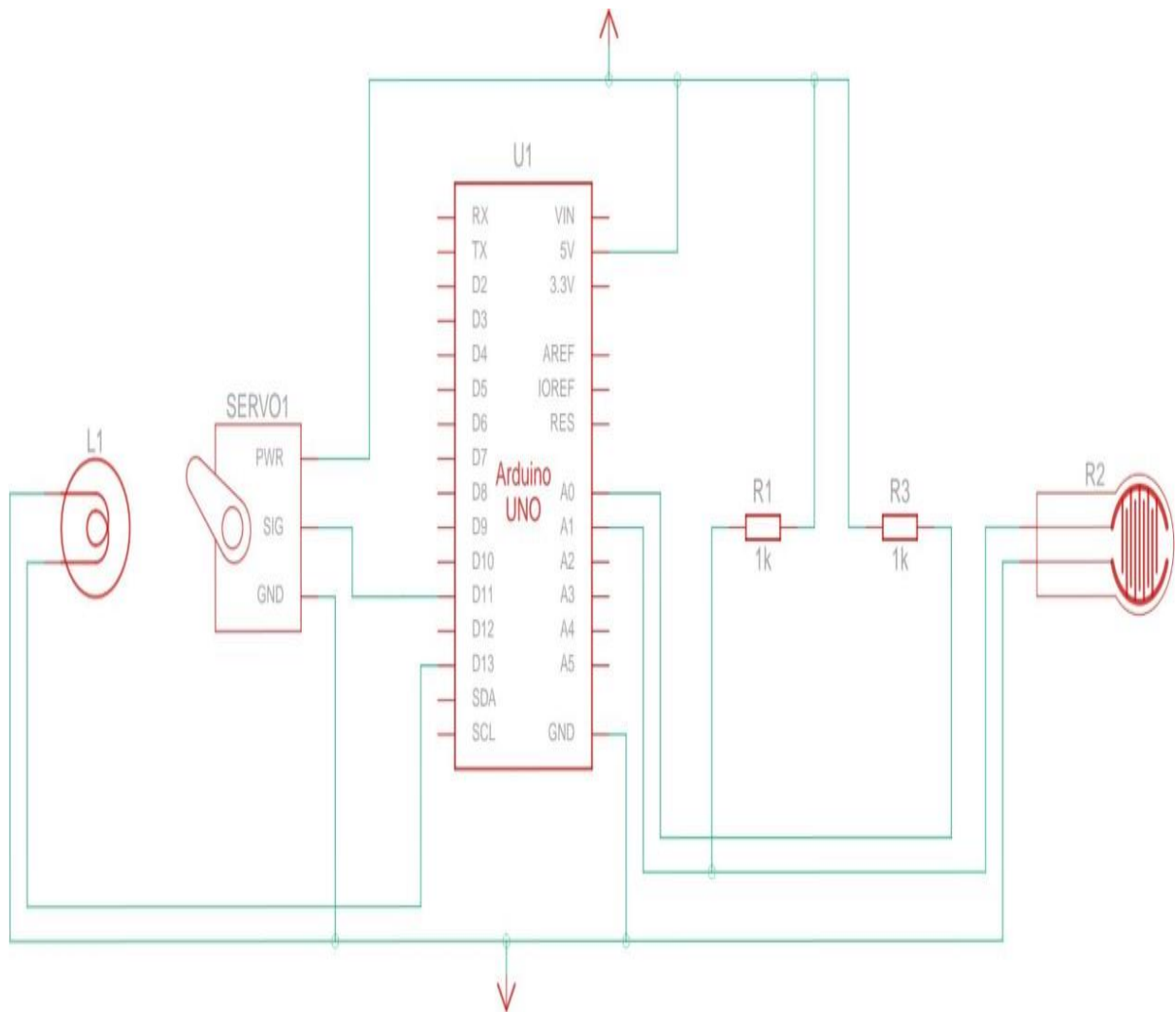


Assignment 1

(Smaert Home)



Circuit design Funky Hango | Tin: x +

https://www.tinkercad.com/things/aklrFmxc7u-funky-hango/editel

Funky Hango

All changes saved

Code Start Simulation Send To

Text 1 (Arduino Uno R3)

```
19 void loop()
20 {
21   int s1 = analogRead(A0);
22   int s2 = analogRead(A1);
23   //Serial.println(s1);
24   Serial.println(s2);
25   delay(10);
26   if(s2<700)
27   { for (pos = 0; pos <= 180; pos += 2) {
28
29     myservo.write(pos);
30     delay(5);
31   }
32   for (pos = 180; pos >= 0; pos -= 2) {
33     myservo.write(pos);
34     delay(5);
35   }
36   delay(2000);
37 }
38 if(s1>500)
39 {
40   digitalWrite(13, HIGH);
41 }
42 if(s1<500)
43 {
44   digitalWrite(13, LOW);
45 }
46 }
```

Serial Monitor

1023
1023
1023
1023

Send Clear

The image shows a Tinkercad workspace with an Arduino Uno R3 board. It is connected to a light bulb (digital output), a servo motor (PWM output), and a buzzer (digital output). Two potentiometers are connected to the analog input pins A0 and A1. The code in the 'Code' tab implements the following logic: 1. Read A0 and A1. 2. Print A1 to the serial monitor. 3. Delay 10ms. 4. If A2 < 700, move the servo from 0 to 180 degrees and back in 2-degree increments, with 5ms delays between steps. 5. Delay 2000ms. 6. If A1 > 500, turn the buzzer on (HIGH). 7. If A1 < 500, turn the buzzer off (LOW). The Serial Monitor shows the value 1023 being printed repeatedly.

Program:

```
#include <Servo.h>
Servo myservo;
int pos = 0;
void setup()
{
  Serial.begin(9600);
  pinMode(13, OUTPUT);
  pinMode(11, OUTPUT);
  pinMode(A0, INPUT);
  pinMode(A1, INPUT);
  myservo.attach(11);
}

void loop()
{
  int s1 = analogRead(A0);
  int s2 = analogRead(A1);
  //Serial.println(s1);
  Serial.println(s2);
  delay(10);
  if(s2<700)
  { for (pos = 0; pos <= 180; pos += 2) {

    myservo.write(pos);
    delay(5);
  }
  for (pos = 180; pos >= 0; pos -= 2) {
    myservo.write(pos);
    delay(5);
  }
  delay(2000);
}
  if(s1>500)
  {
    digitalWrite(13, HIGH);
  }
  if(s1<500)
  {
    digitalWrite(13, LOW);
  }
}
```