Virtual Reality Haptic Interactions (虛擬實境與觸覺回饋互動)

Lecturer: Ray
Week 5 (10/9)

Syllabus

	1	Introduction	9	Unity	
	2	Brain storming & paper reading instruction	10	Unity	
Team up	3	Introducing Arduino	11	Unity, communication and XR	**Check point**
	4	Arduino, coding	12	AI and LLM	
Proposal check point	5	Motors	13	Project discussion	
	6	Motor and encoder	14	3D printing	**Check point**
Proposal check point	7	Paper discussion	15	Project discussion and checkpoint	
	8	Proposal (with lo-fi prototype)	16	Final project	

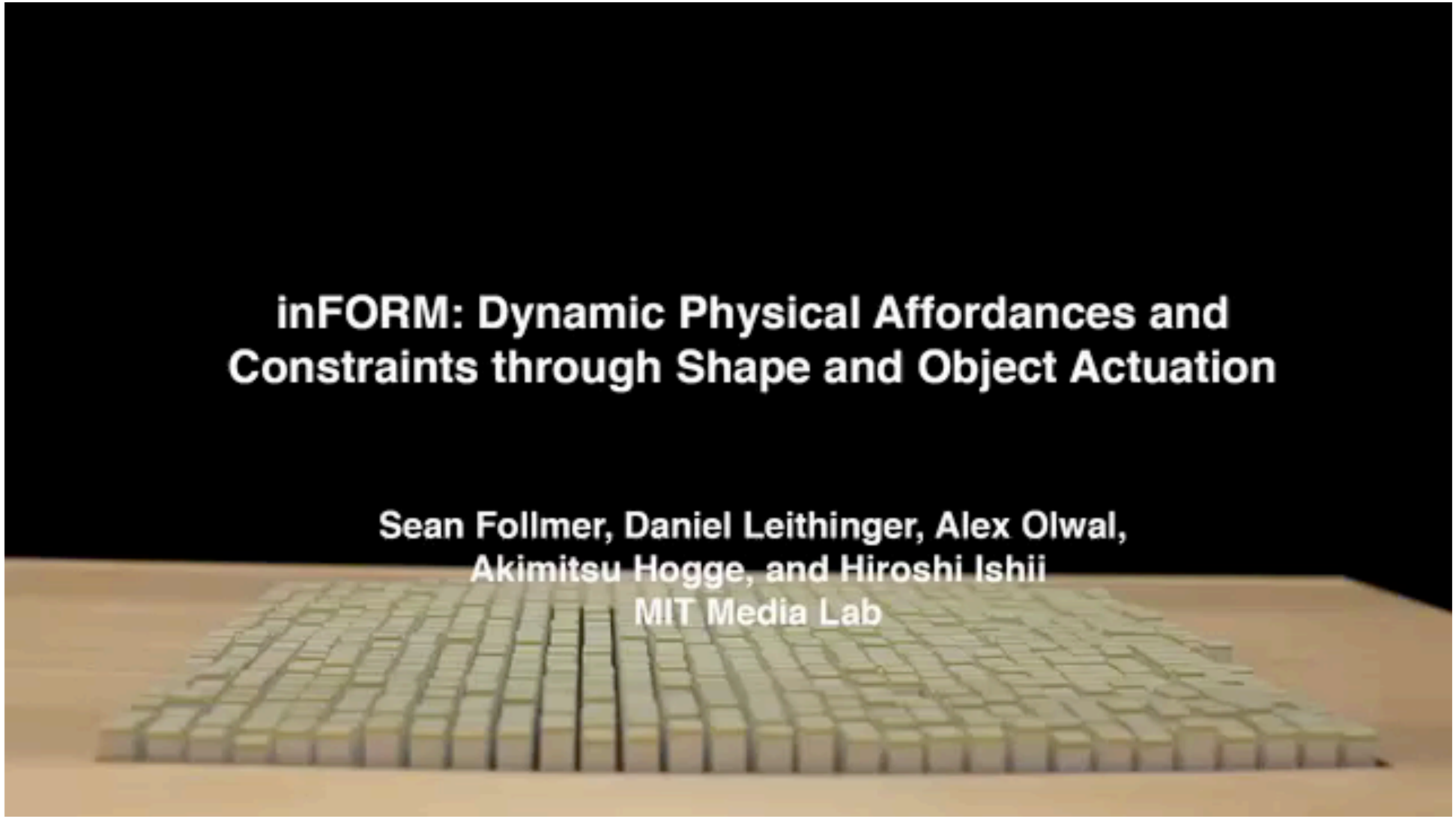
Servo motor

MobiLimb, UIST'18

Drag:on, CHI'19

PaCaPa, CHI'19

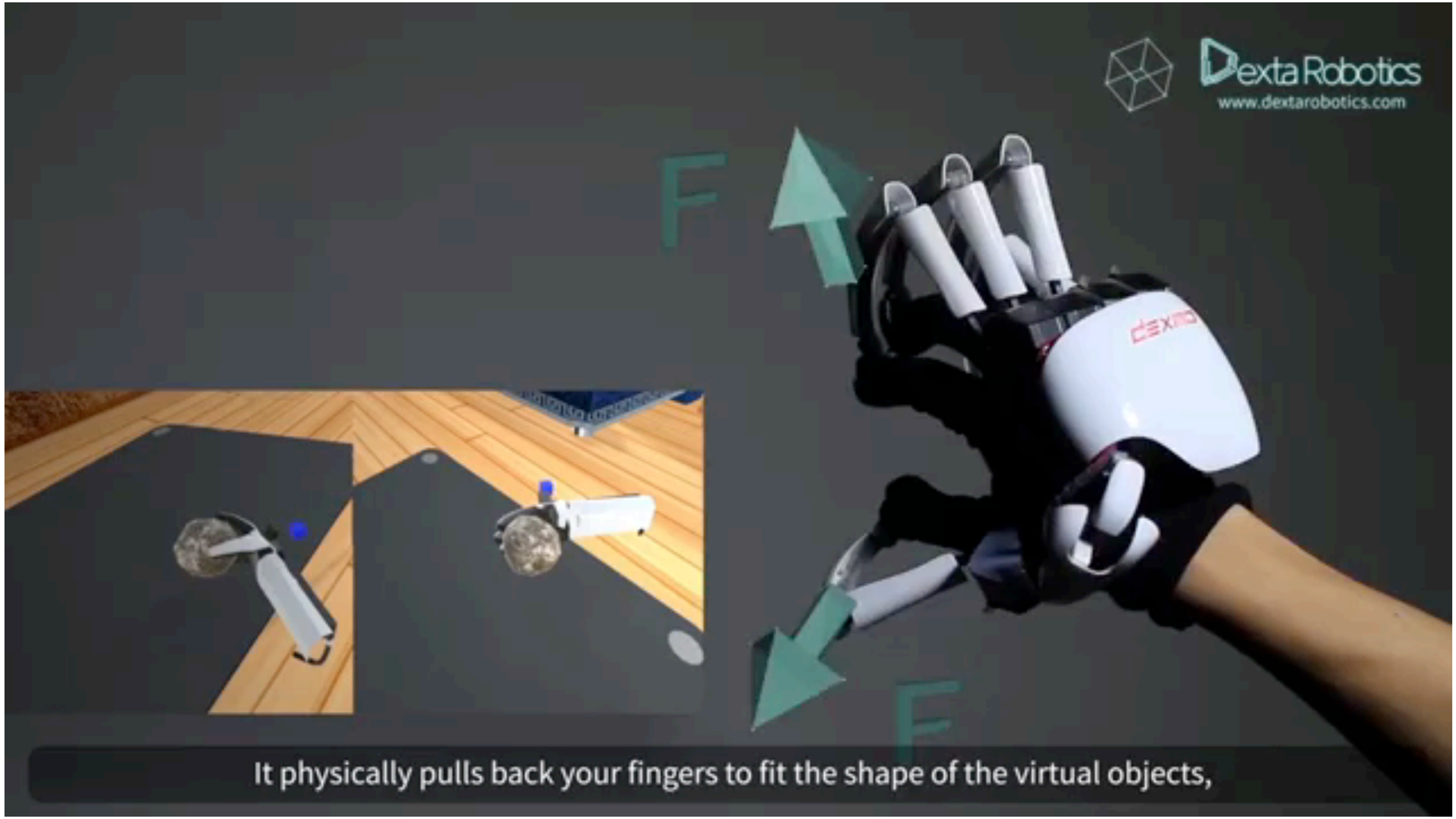
NormalTouch and TextureTouch, UIST'16



inFORM, UIST'13

LineFORM, UIST’15

ChainFORM, UIST'16



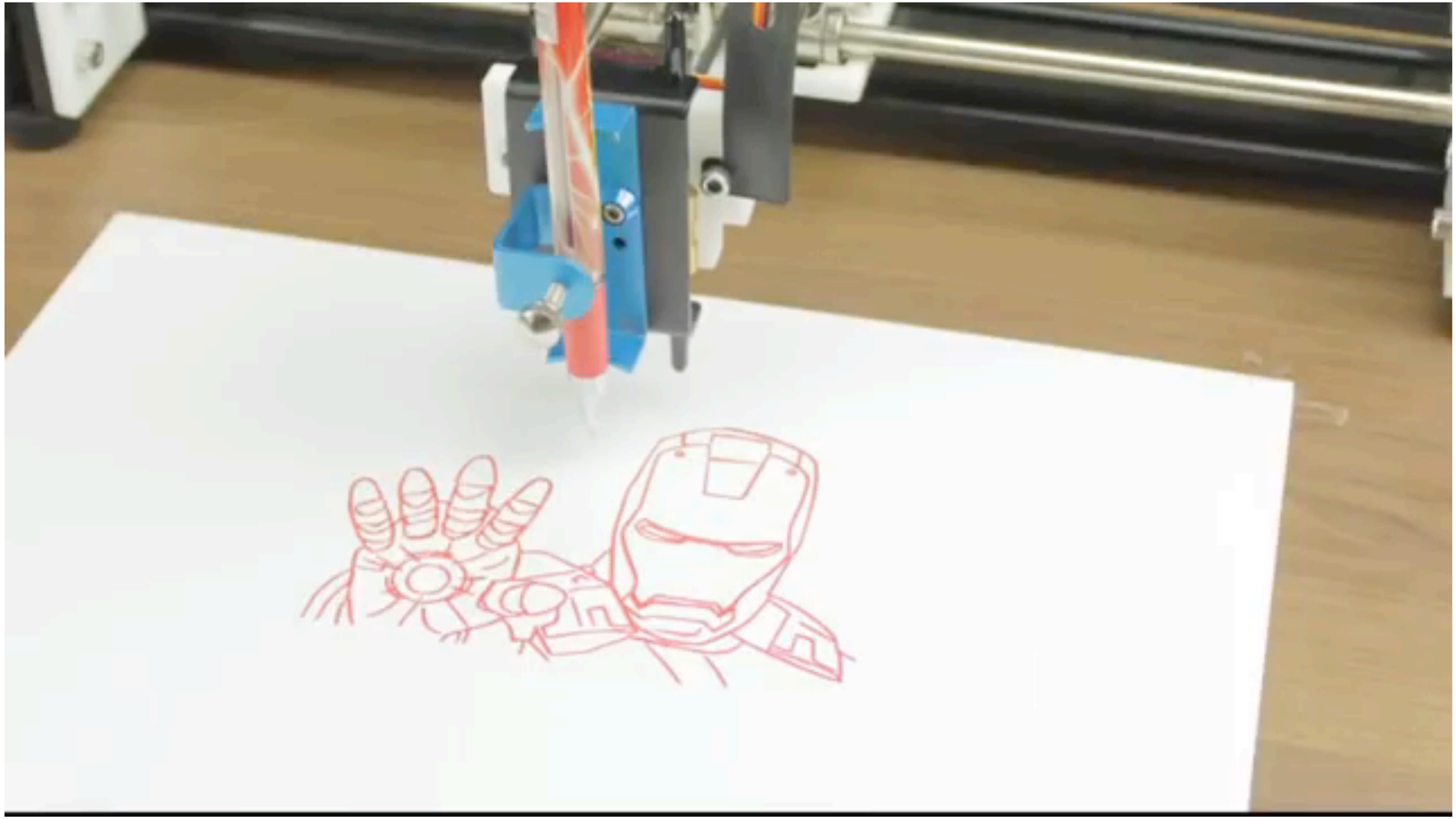
Dexmo, UIST'16

MetaLimbs, SIGGRAPH'17 ET
MetaArms, UIST'18

Fusion, SIGGRAPH’18 ET

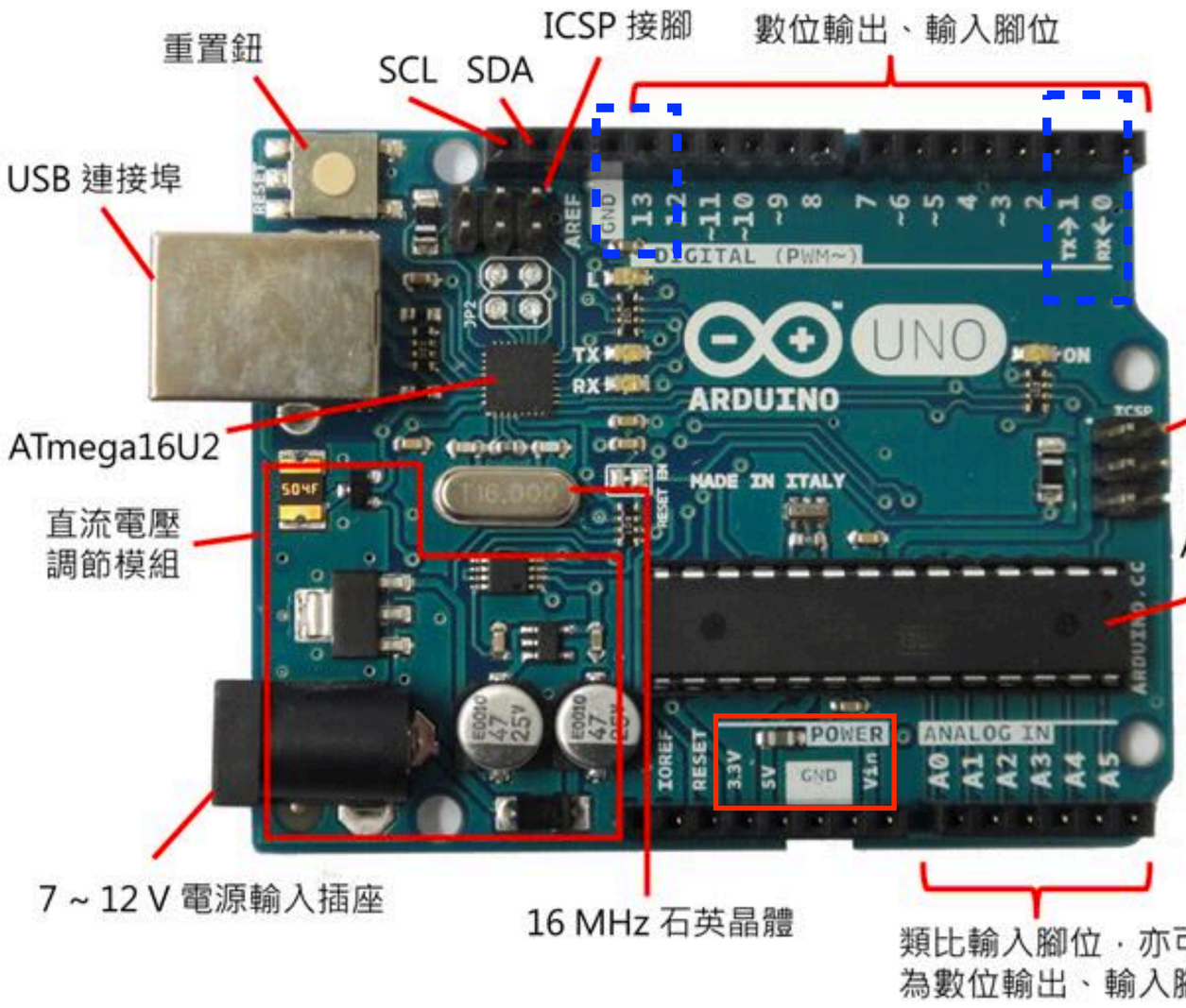
Motion Guidance Sleeve, CHI'16

Shifty, IEEE VR'17



[https://youtu.be/hlgmn2mxVTY]

Arduino



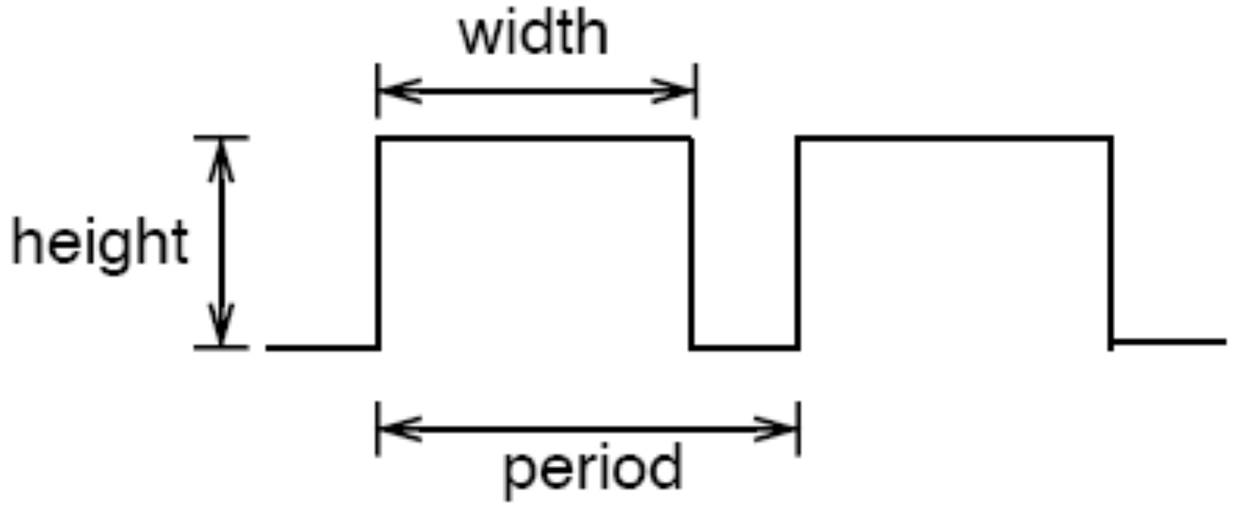
~: PWM (0~255)

Pin 0, 1 : RX, TX

Pin 13: LED

Analog-to-digital converter (ADC) (0~1023)

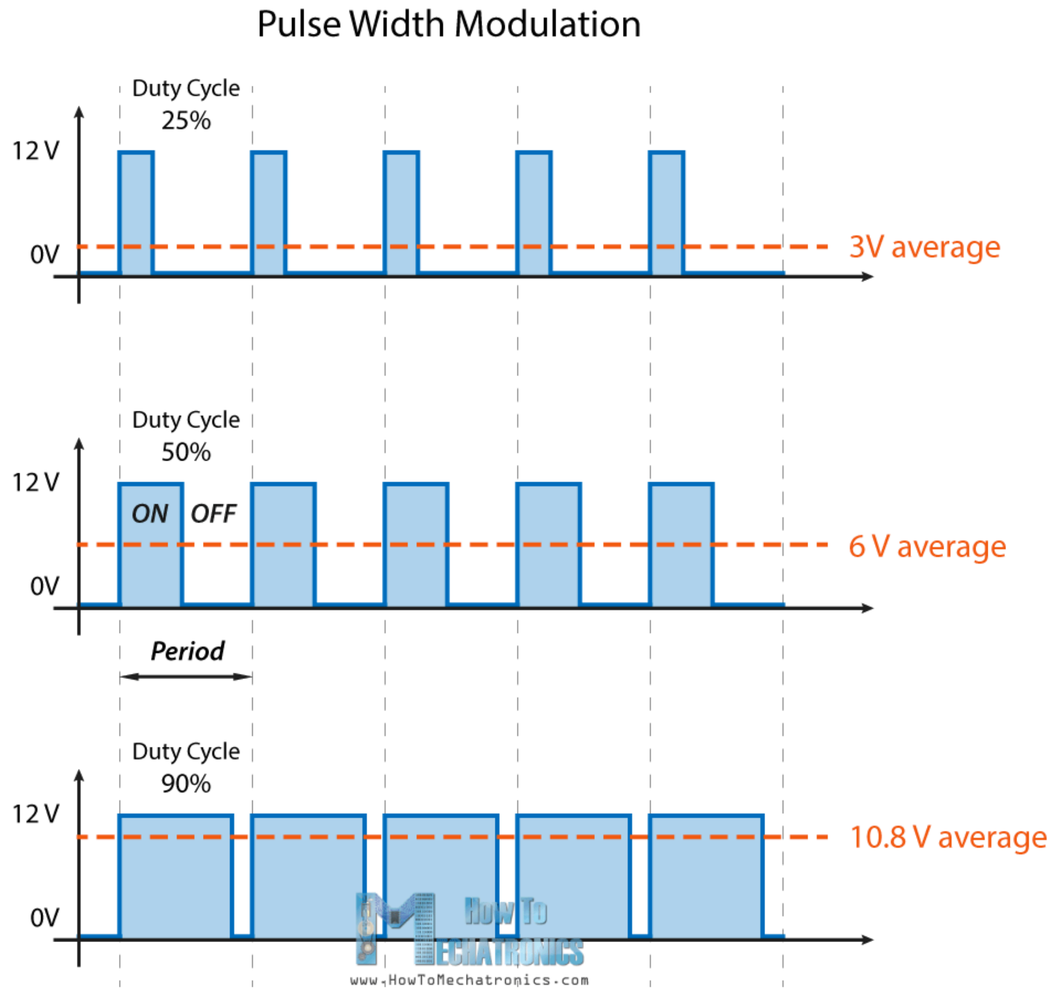
Pulse Width Modulation (PWM)



Three characteristics of PWM signals:

- **Pulse width range (min/max)**
- **Pulse period (=1/pulse per second)**
- **voltage levels (0-5V, for instance)**

Pulse Width Modulation (PWM)

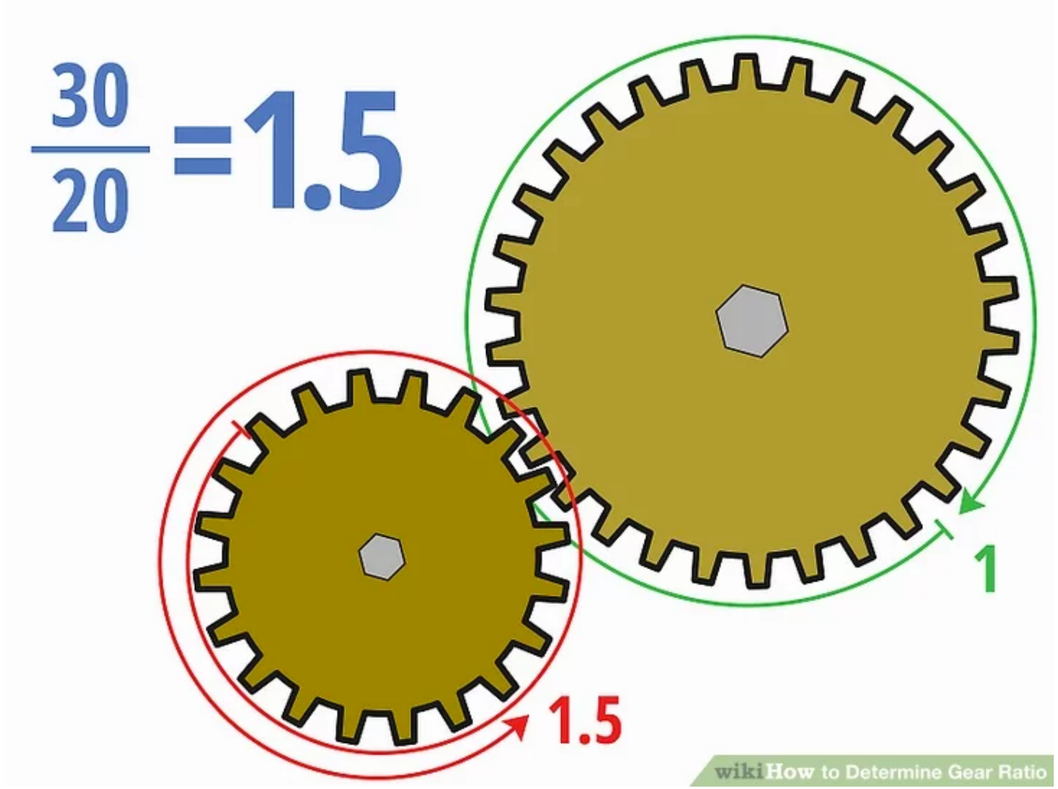


https://goo.gl/hcSV1W

https://goo.gl/vfDVW6

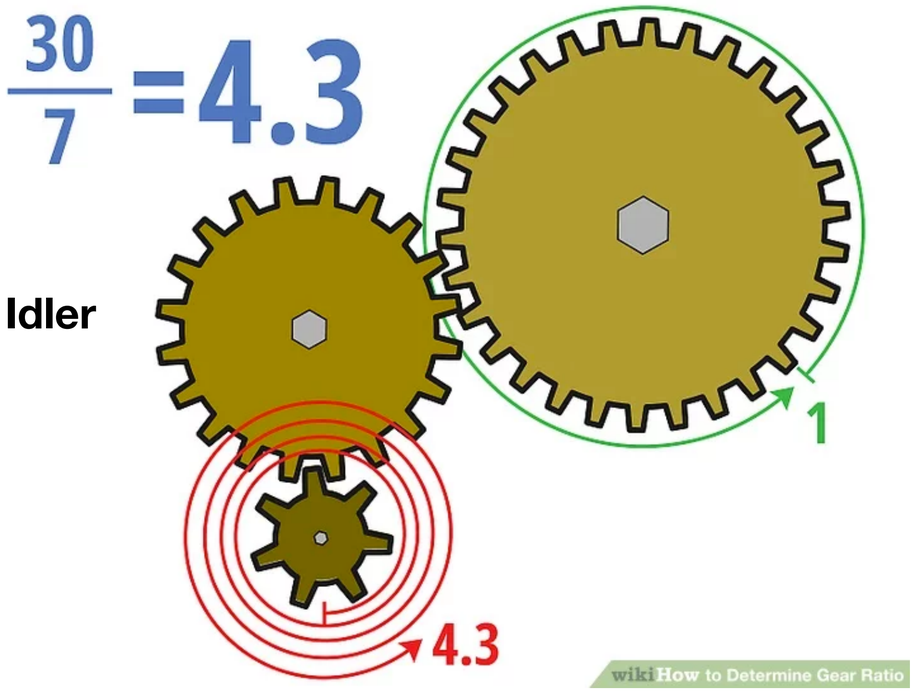
Gear Ratio

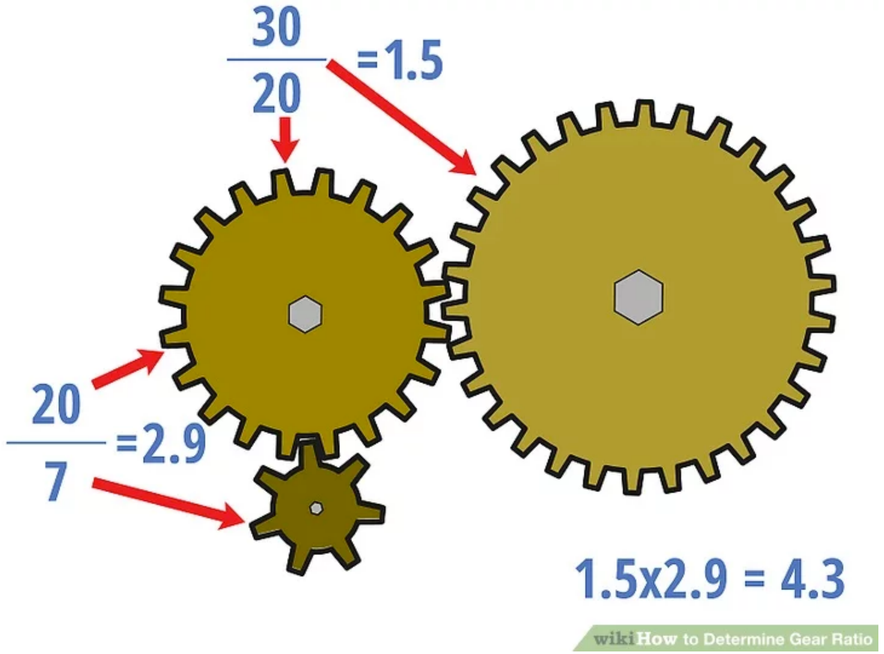
$$\text{Gear ratio} = \frac{\text{Number of teeth on driven gear}}{\text{Number of teeth on driver gear}}$$



Gear Ratio

$$\text{Gear ratio} = \frac{\text{Number of teeth on driven gear}}{\text{Number of teeth on driver gear}}$$

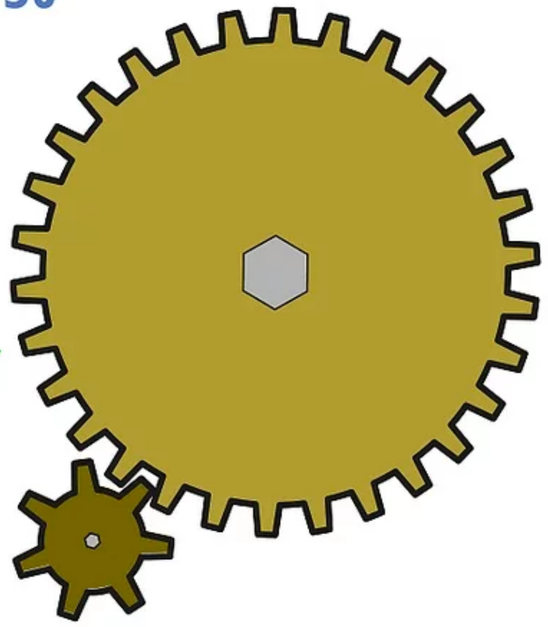


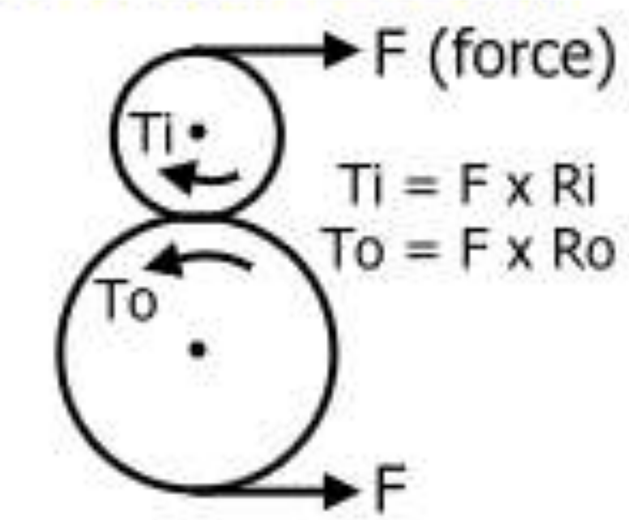


Gear Ratio

130 rpms × 7 = S2 × 30
910 = S2 × 30
910/30 = S2
30.33 rpms = S2

30.33
rpm



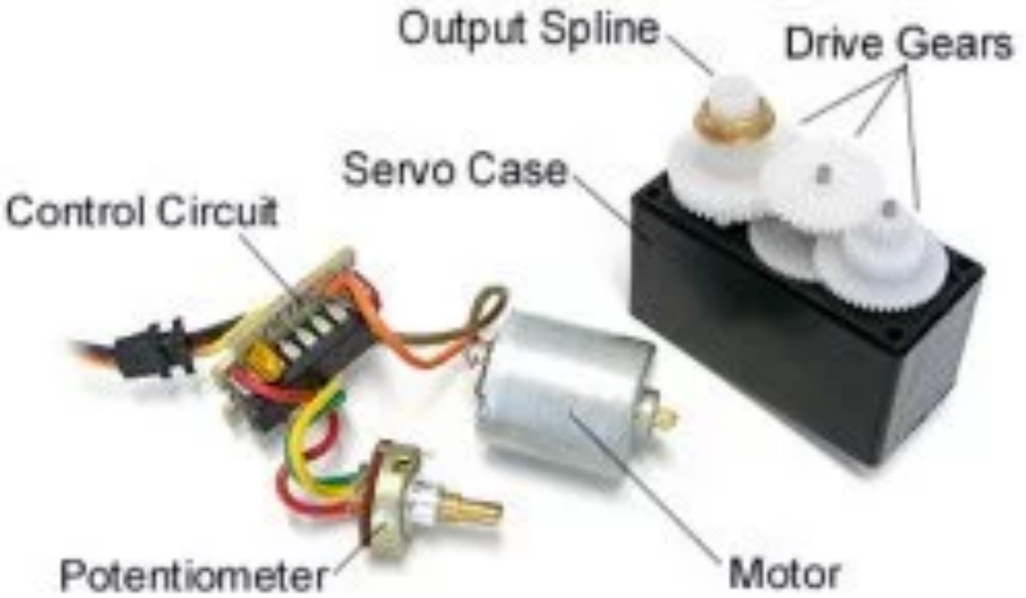


TORQUE

$$To = Ti \times \frac{Ro}{Ri}$$

S1 × T1 = S2 × T2

Servo Motor

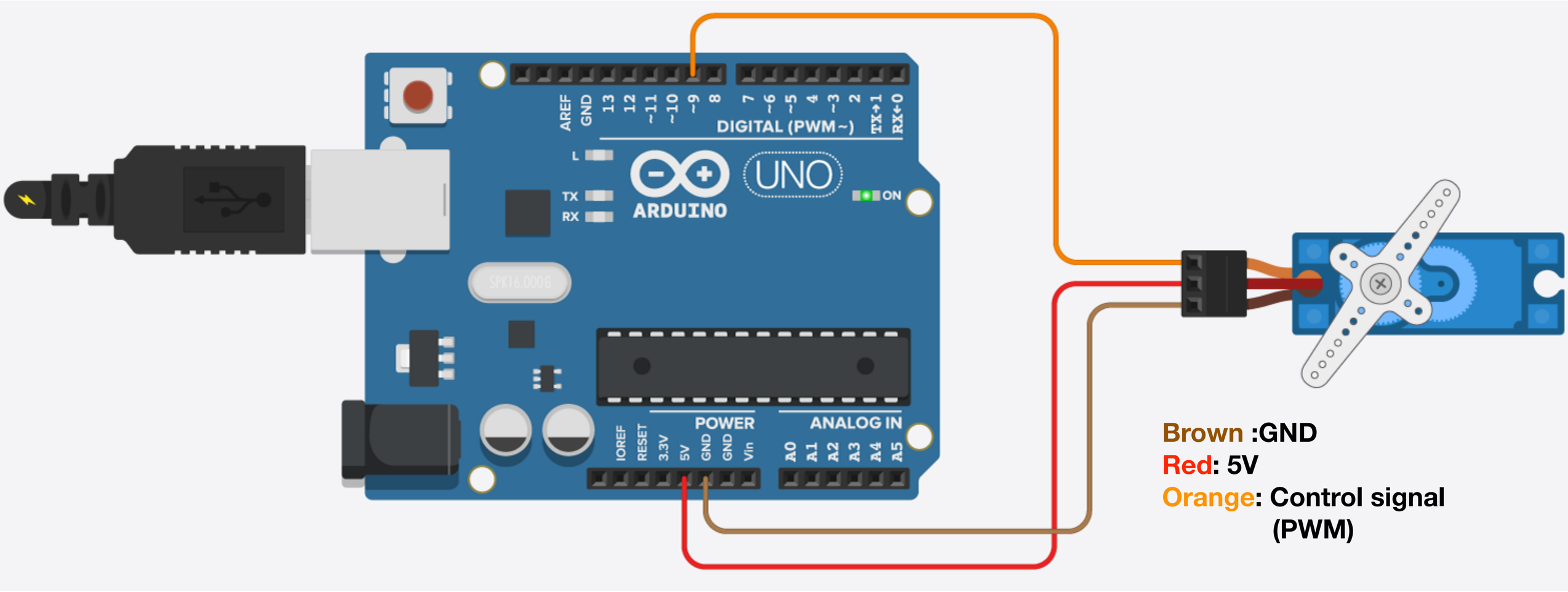

Output Spline
Drive Gears
Servo Case
Control Circuit
Potentiometer
Motor

Servo Motor

- Only rotate from 0 to 180 degree.

- Two control methods:
  - PWM
  - Angle

Servo Motor


AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM~)
L
TX
RX
ARDUINO
UNO
ON
POWER
ANALOG IN
IOREF
RESET
3.3V
5V
GND
GND
Vin
A0
A1
A2
A3
A4
A5
Brown :GND
Red: 5V
Orange: Control signal (PWM)

Servo Motor

```
#include <Servo.h>

Servo myservo; // 建立Servo物件，控制伺服馬達

void setup()
{
  myservo.attach(9); // 連接數位腳位9，伺服馬達的訊號線
}

void loop()
{
  myservo.write(180);
  delay(500);
  myservo.write(0);
  delay(500);
}
```

[https://bit.ly/3URBWR8]

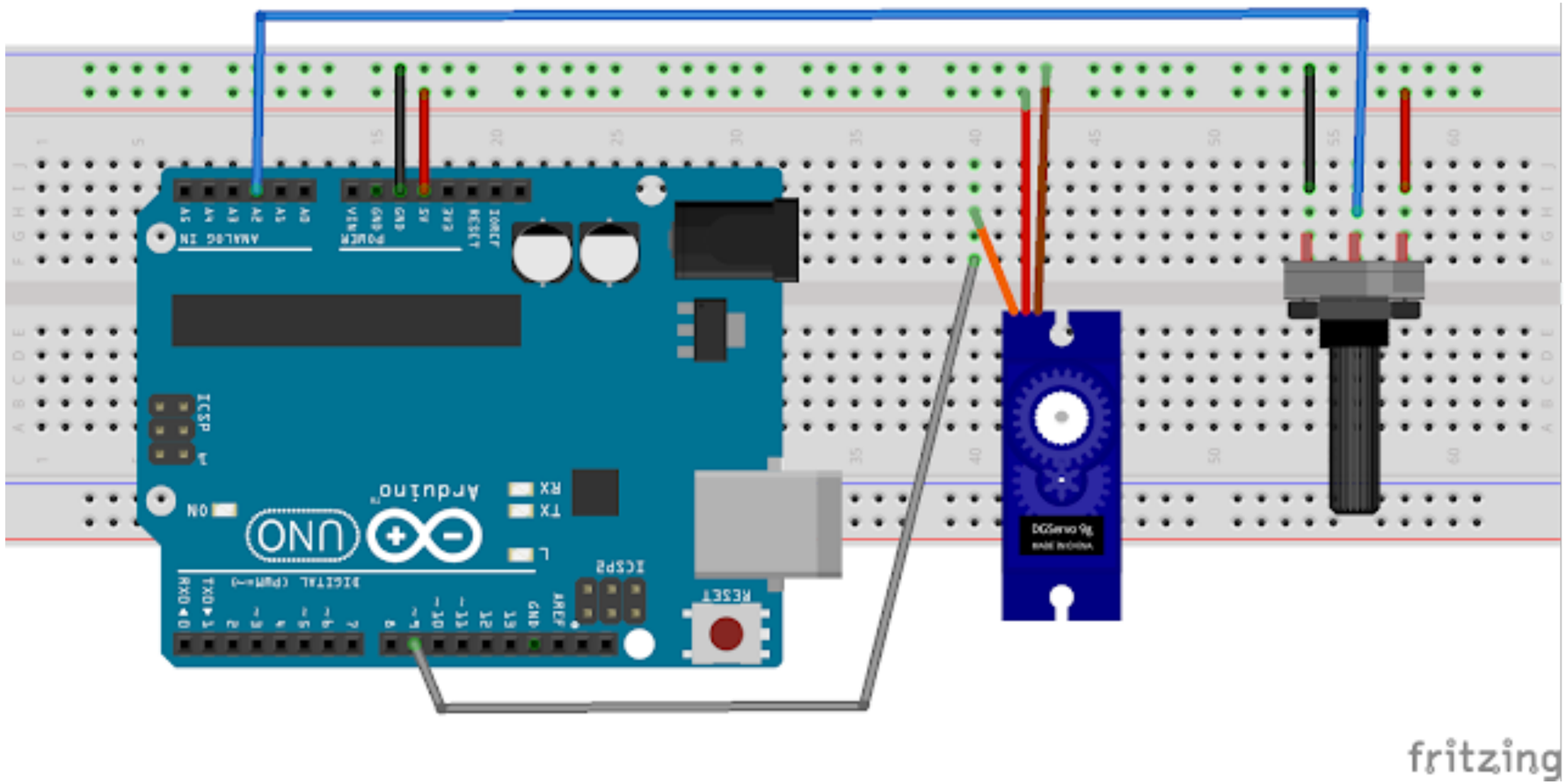
Servo Motor

```
void setup()
{
  myservo.attach(9, 500, 2400); // 連接數位腳位9，伺服馬達的訊號線，修正脈衝寬度範圍
  myservo.write(90); // 一開始先置中90度
  delay(3000);
}

void loop()
{
  for(int i = 500; i <= 2400; i+=100){
    myservo.writeMicroseconds(i); // 直接以脈衝寬度控制
    delay(100);
  }
  for(int i = 2400; i >= 500; i-=100){
    myservo.writeMicroseconds(i);
    delay(100);
  }
}
```

Standard:
1000: fully counter-clockwise
1500: mid-point
2000:fully clockwise

[https://goo.gl/aEKqi4]

Servo Motor Control (Potentiometer)



Servo Motor

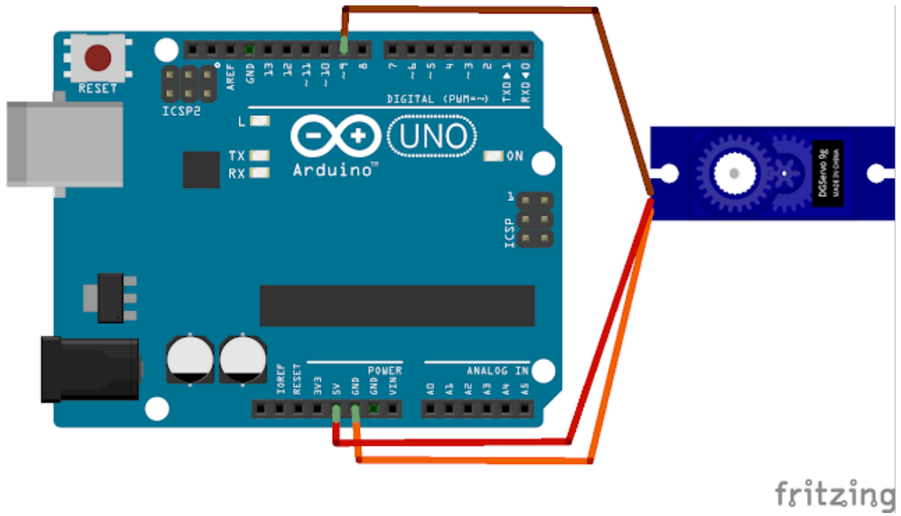
```
#include <Servo.h>
Servo myservo;        // 建立Servo物件，控制伺服馬達

int potpin = 2;       // 可變電阻連接的 pin 腳
int potval;           // 從可變電阻來的數值

void setup()
{
  myservo.attach(9);  // 將 servo 物件連接到 pin 9
}

void loop()
{
  potval = analogRead(potpin);            // 從可變電阻讀取類比數值 (介於 0 跟 1023)
  potval = map(potval, 0, 1023, 0, 179);  // 將類比值轉換為 servo 可讀取的範圍值(介於0 跟 180)
  myservo.write(potval);                  // 設定 servo 位置
  delay(15);                              // 等待 15ms 讓 servo 走到指定位置
}
```

→ analog signal (0-1023)

[https://goo.gl/9HsKgE]

Continuous Rotation Servo Motor

- Control the rotating speed
  0° : counterclockwise
  90° : stop
  180° : clockwise



Brown :GND
Red: 5V
Orange: Control signal (PWM)

Continuous Rotation Servo Motor

- No rotated limitation

```
void loop()
{

  myservo.write(0);                  // rotate counterclockwise full speed
  delay(2000);
  myservo.write(95);                  // stop
  delay(2000);
  myservo.write(180);                 // rotate clockwise full speed
  delay(2000);
  myservo.write(95);                  // stop
  delay(1000);

  myservo.write(89);                  // rotate very slowly counterclockwise
  delay(2000);
  myservo.write(95);                  // stop
  delay(1000);
  myservo.write(100);                  // rotate very slowly clockwise
  delay(2000);
  myservo.write(95);                  // stop
  delay(1000);
}
```

[https://goo.gl/PqAKDA]

Bending Sensor and Servo

Bending sensor

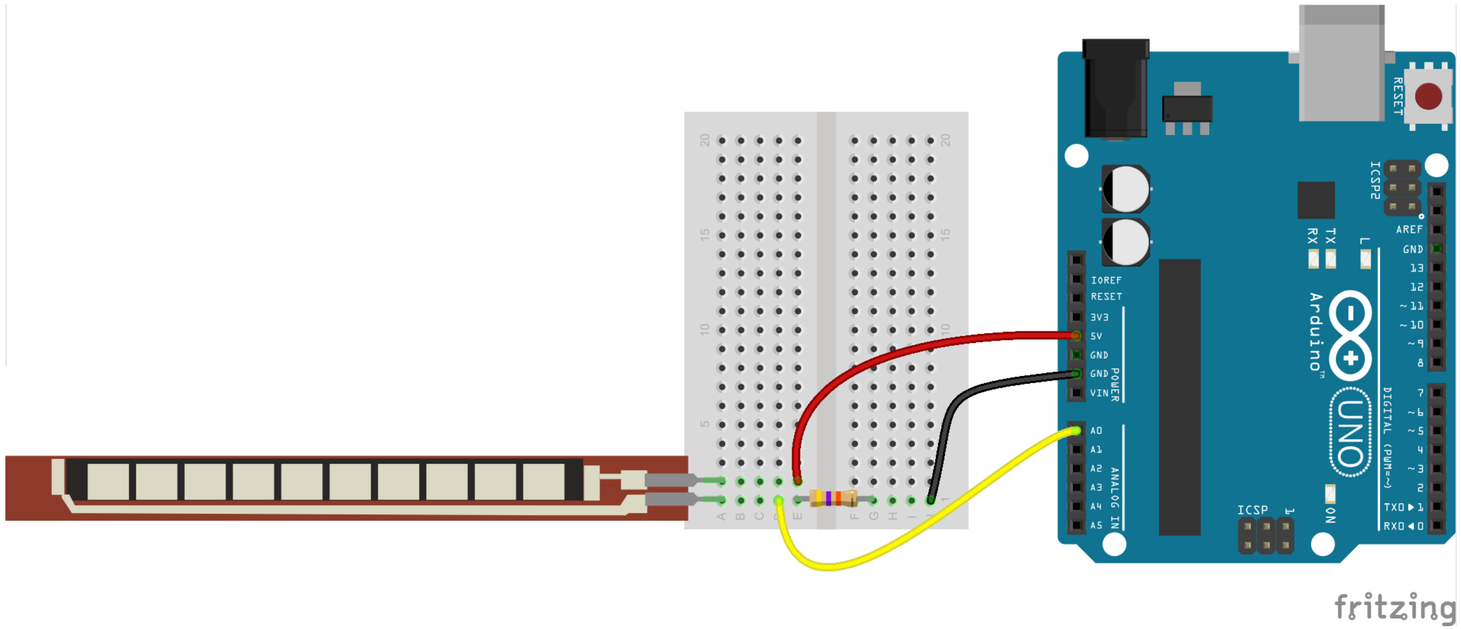
GND	⟷	GND
VCC	⟷	5V

Servo 1

GND	⟷	GND
5V	⟷	5V
PWM	⟷	6

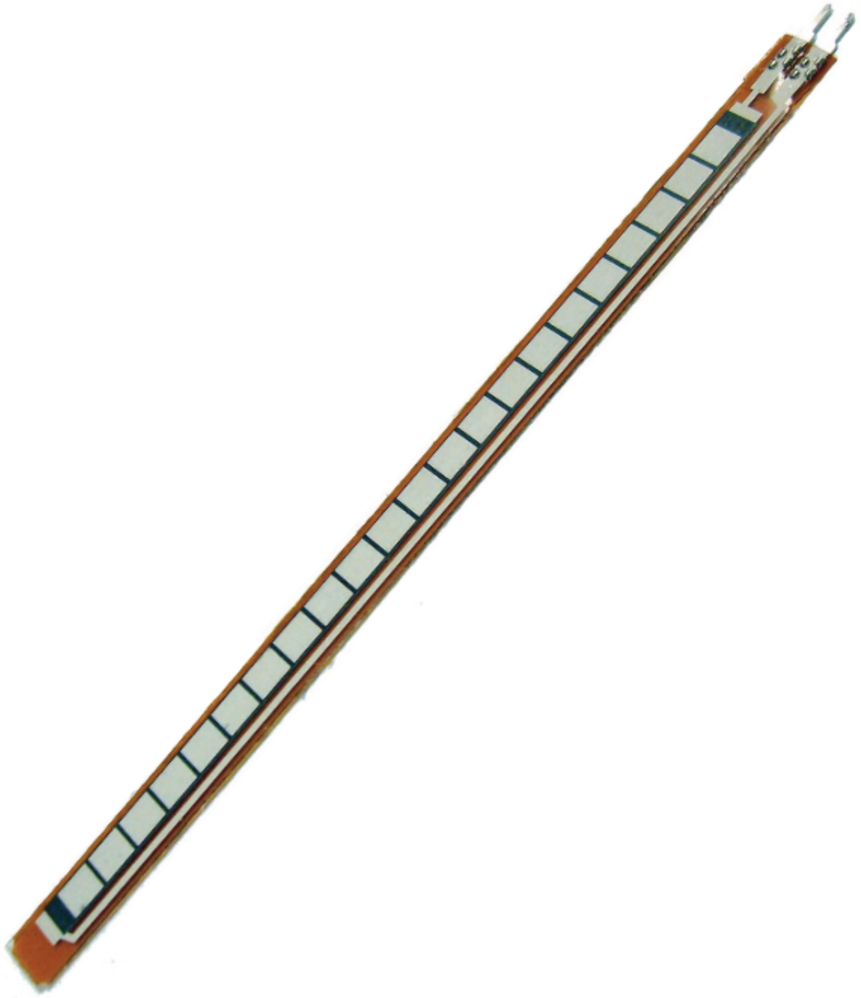
Servo 2

GND	⟷	GND
5V	⟷	5V
PWM	⟷	7

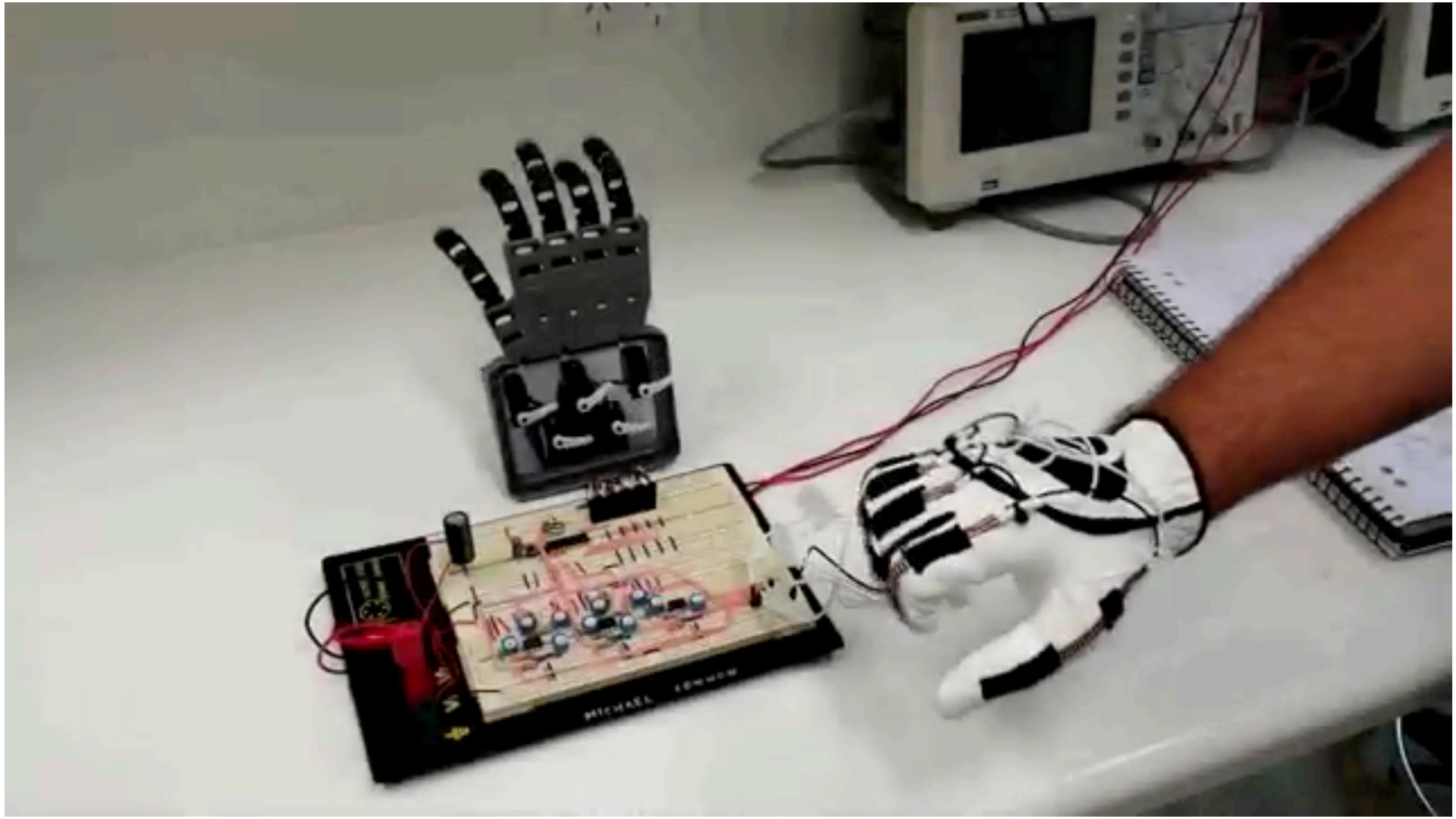


[bit.ly/3TUtOR0]

Flex (Bend) Sensor



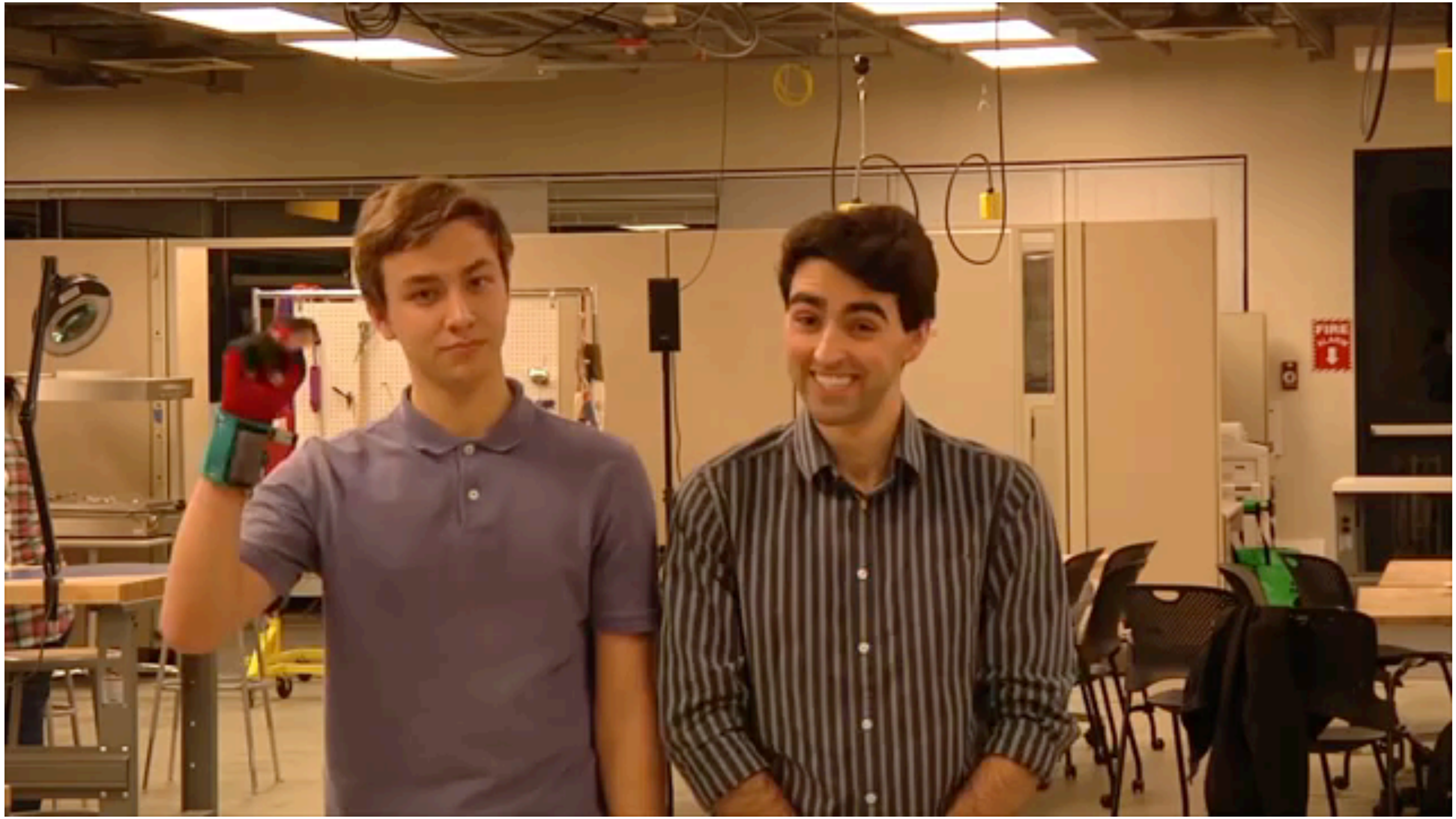
[https://bit.ly/3woF2Sc]



[https://youtu.be/PIhmBszYHOU]

[https://youtu.be/-TOi7datdbs]

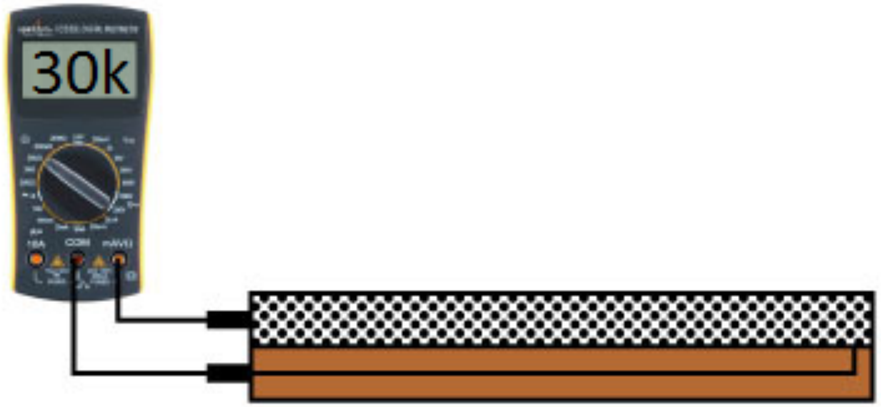
FlexCase, CHI'16



[https://youtu.be/RHTrAXsULOI]

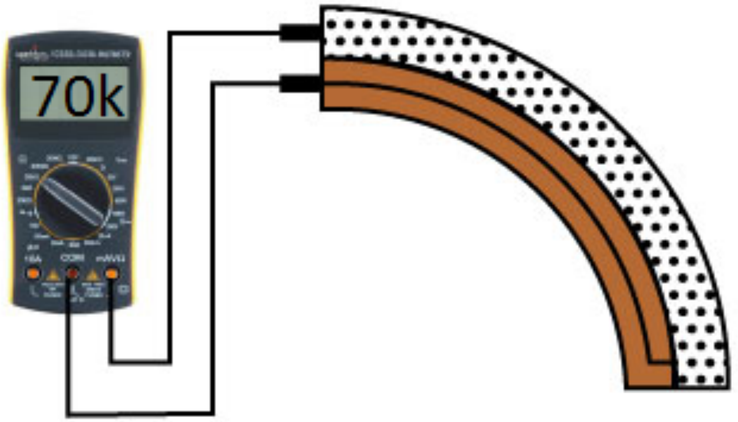
Flex Sensor

Flat



Conductive particles close together - 30kΩ.

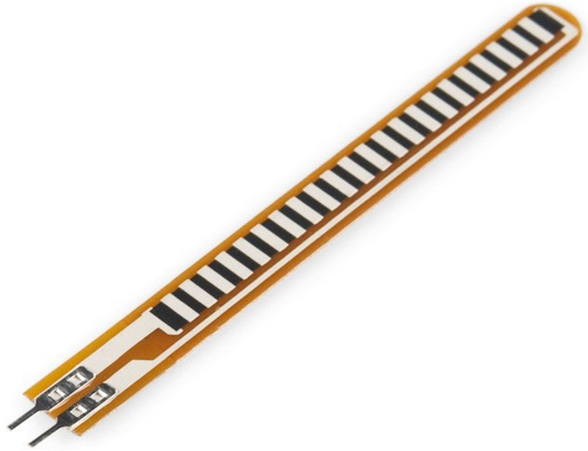
Bend



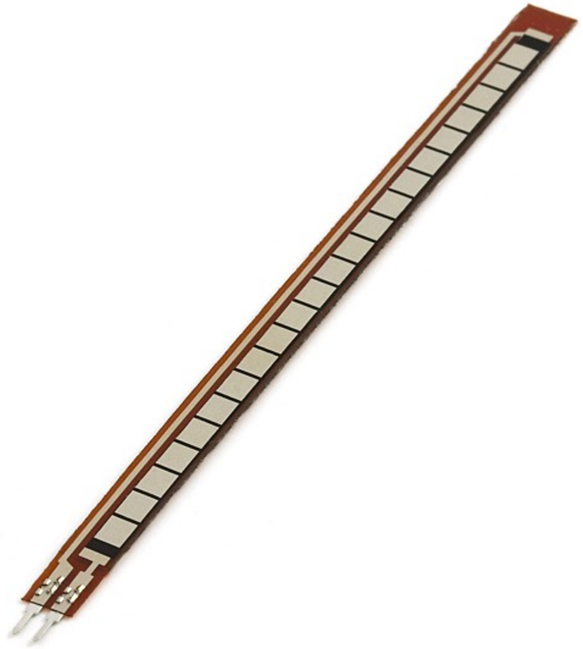
Conductive particles further apart - 70kΩ.

Flex Sensor

- Two sizes



Flex sensor 2.2 (5.888 cm)

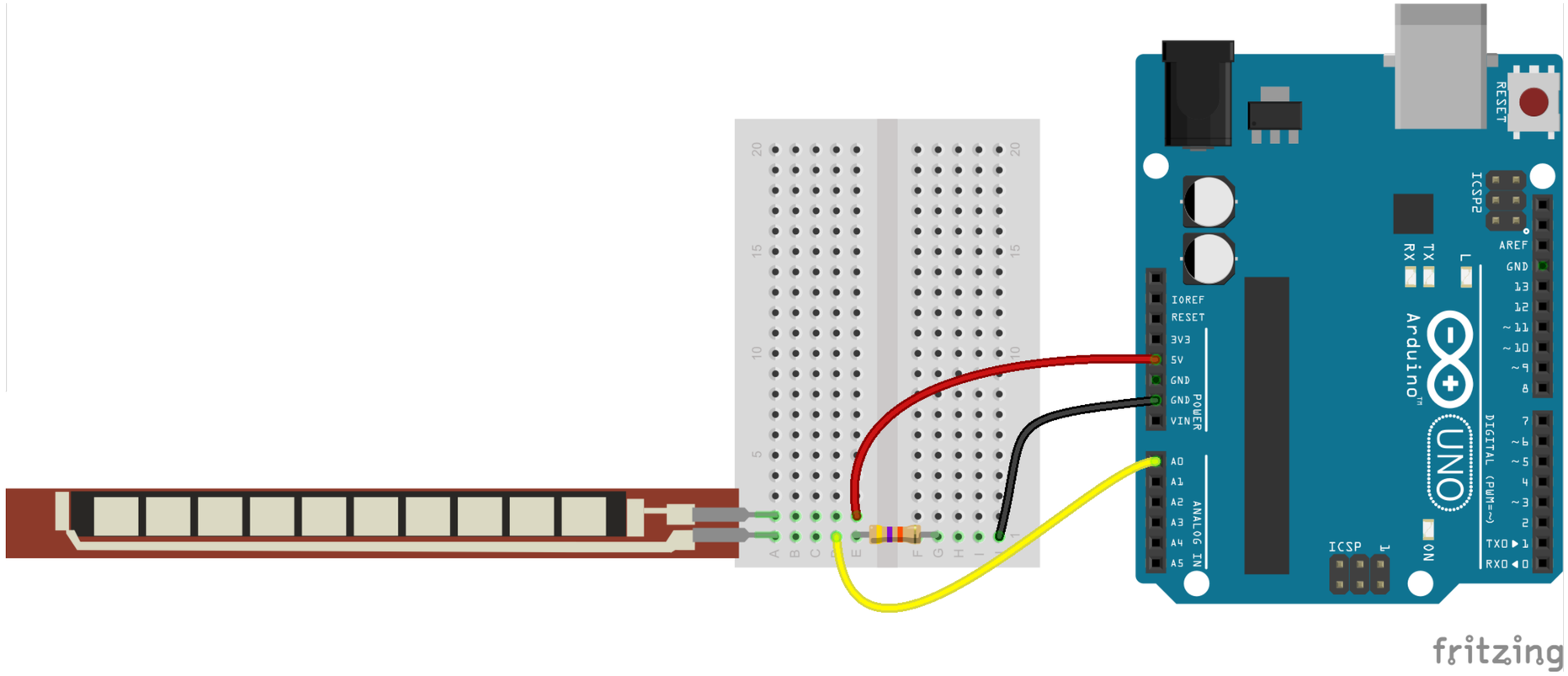


Flex sensor 4.5 (11.43 cm)

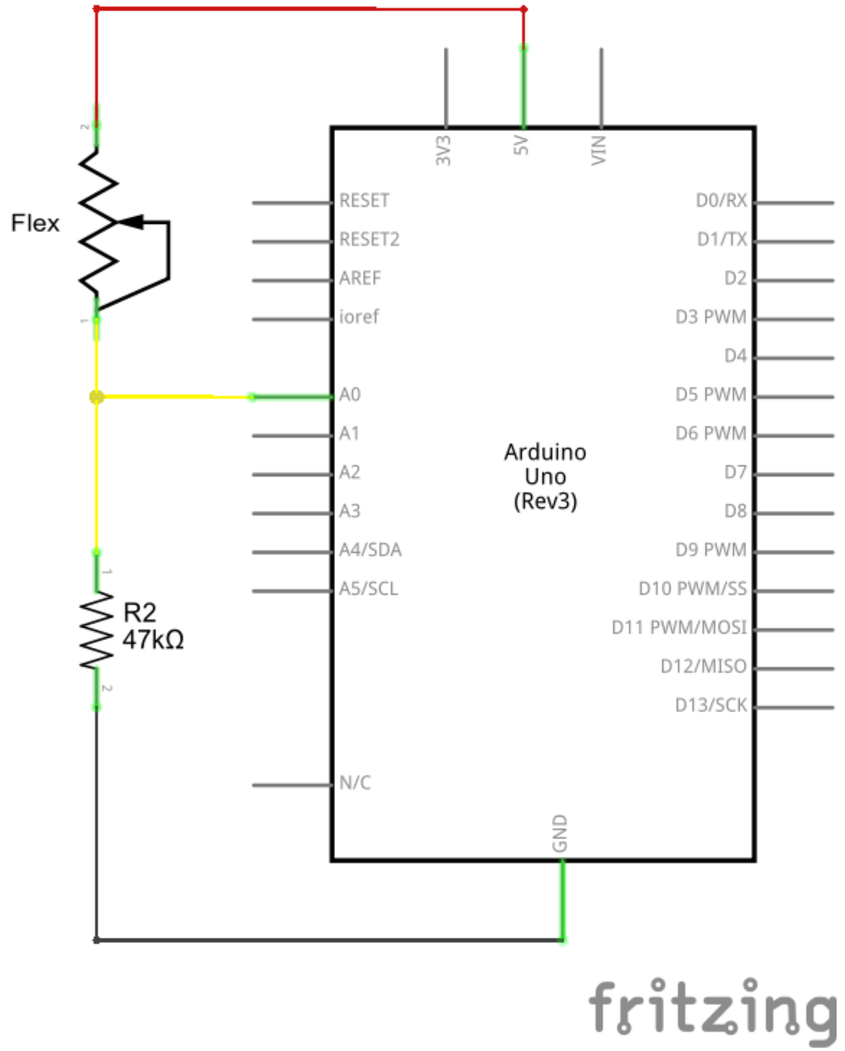
What we need

- Flex sensor
- Arduino Uno
- Breadboard & USB cable
- Dupont line (male to male) * 3
- 47 KΩ ohm resistor * 1

Flex Sensor Circuit



Flex Sensor Circuit



Flex Sensor Code

- Flex_Sensor_Example.ino
  - Upload it and open the serial monitor (9600 baud rate)
  - Press FSR sensor

[https://bit.ly/3tsNXjV]

Flex Sensor Code

- Flex_Sensor_Example.ino

```
const int FLEX_PIN = A0; // Pin connected to voltage divider output

// Measure the voltage at 5V and the actual resistance of your
// 47k resistor, and enter them below:
const float VCC = 4.98; // Measured voltage of Ardunio 5V line
const float R_DIV = 47500.0;

// Upload the code, then try to adjust these values to more
// accurately calculate bend degree.
const float STRAIGHT_RESISTANCE = 37300.0; // resistance when straight
const float BEND_RESISTANCE = 90000.0; // resistance at 90 deg

void setup()
{
  Serial.begin(9600);
  pinMode(FLEX_PIN, INPUT);
}
```

Flex Sensor Code

- Flex_Sensor_Example.ino

```
void loop()
{
  // Read the ADC, and calculate voltage and resistance from it
  int flexADC = analogRead(FLEX_PIN);
  float flexV = flexADC * VCC / 1023.0;
  float flexR = R_DIV * (VCC / flexV - 1.0);
  Serial.println("Resistance: " + String(flexR) + " ohms");

  // Use the calculated resistance to estimate the sensor's
  // bend angle:
  float angle = map(flexR, STRAIGHT_RESISTANCE, BEND_RESISTANCE,
                    0, 90.0);
  Serial.println("Bend: " + String(angle) + " degrees");
  Serial.println();

  delay(500);
}
```

** same to FSR code **

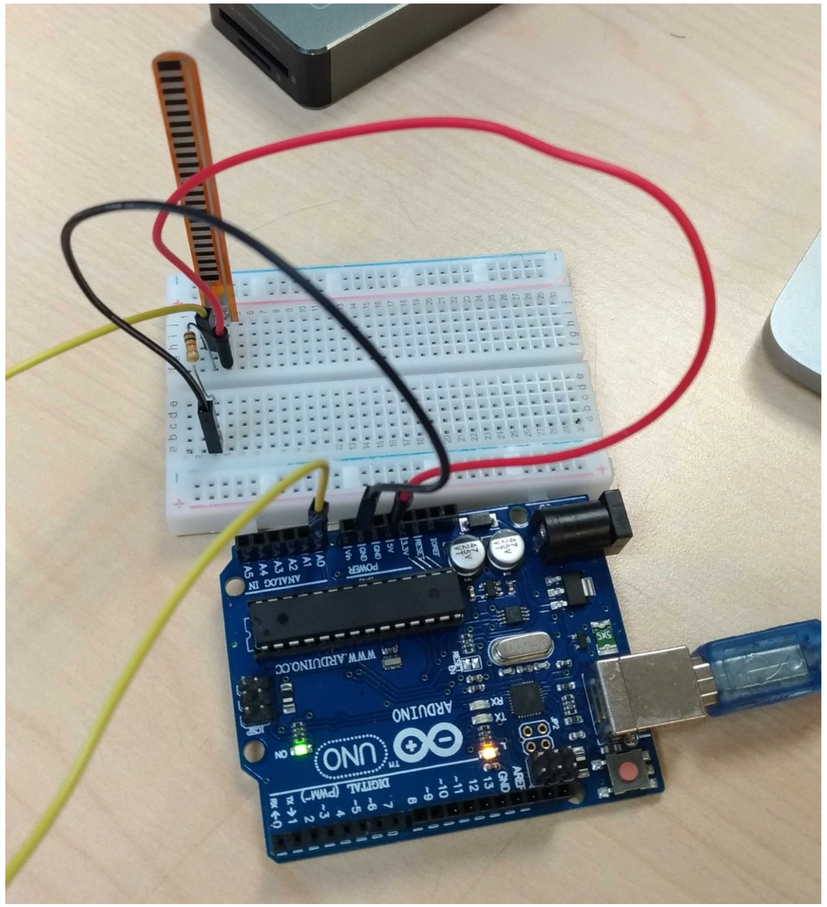
Flex Sensor Code

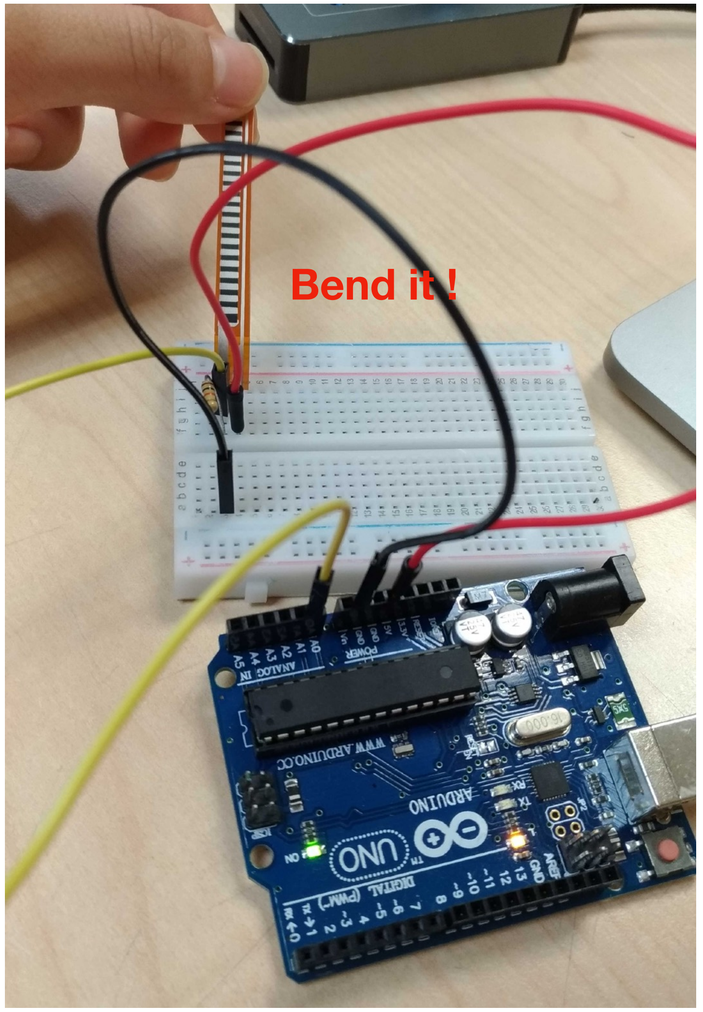
- Flex_Sensor_Example.ino

```
void loop()
{
  // Read the ADC, and calculate voltage and resistance from it
  int flexADC = analogRead(FLEX_PIN);
  float flexV = flexADC * VCC / 1023.0;
  float flexR = R_DIV * (VCC / flexV - 1.0);
  Serial.println("Resistance: " + String(flexR) + " ohms");

  // Use the calculated resistance to estimate the sensor's
  // bend angle:
  float angle = map(flexR, STRAIGHT_RESISTANCE, BEND_RESISTANCE,
                    0, 90.0);
  Serial.println("Bend: " + String(angle) + " degrees");
  Serial.println();

  delay(500);
}
```

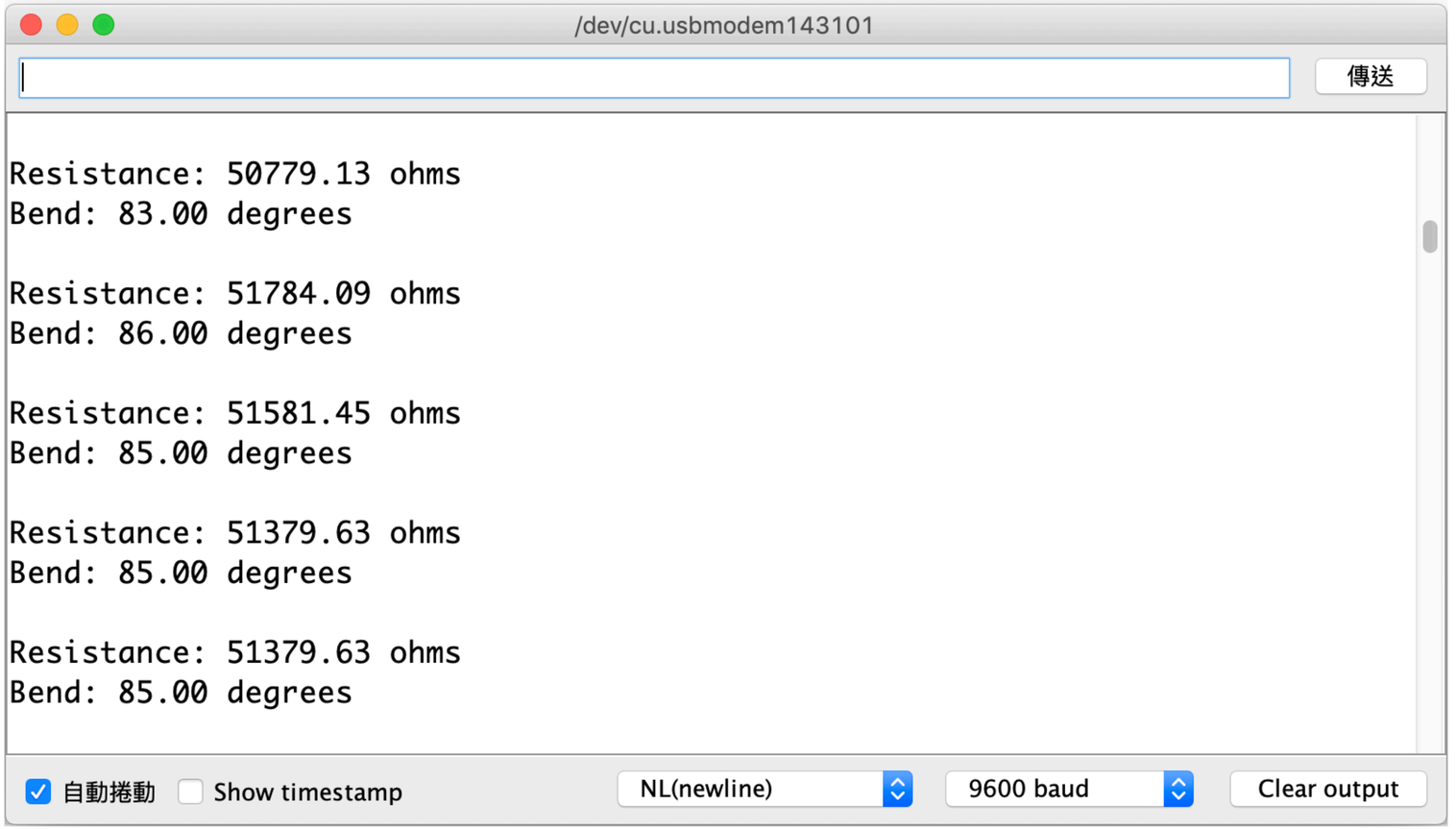
map value to 0 ~ 90 angle

Flex Sensor Code





Flex Sensor Code



Bending in Unity

Unity: https://bit.ly/3JUcCol
Bending: https://bit.ly/3xBWSBN

FSR Sensor and Servo

FSR

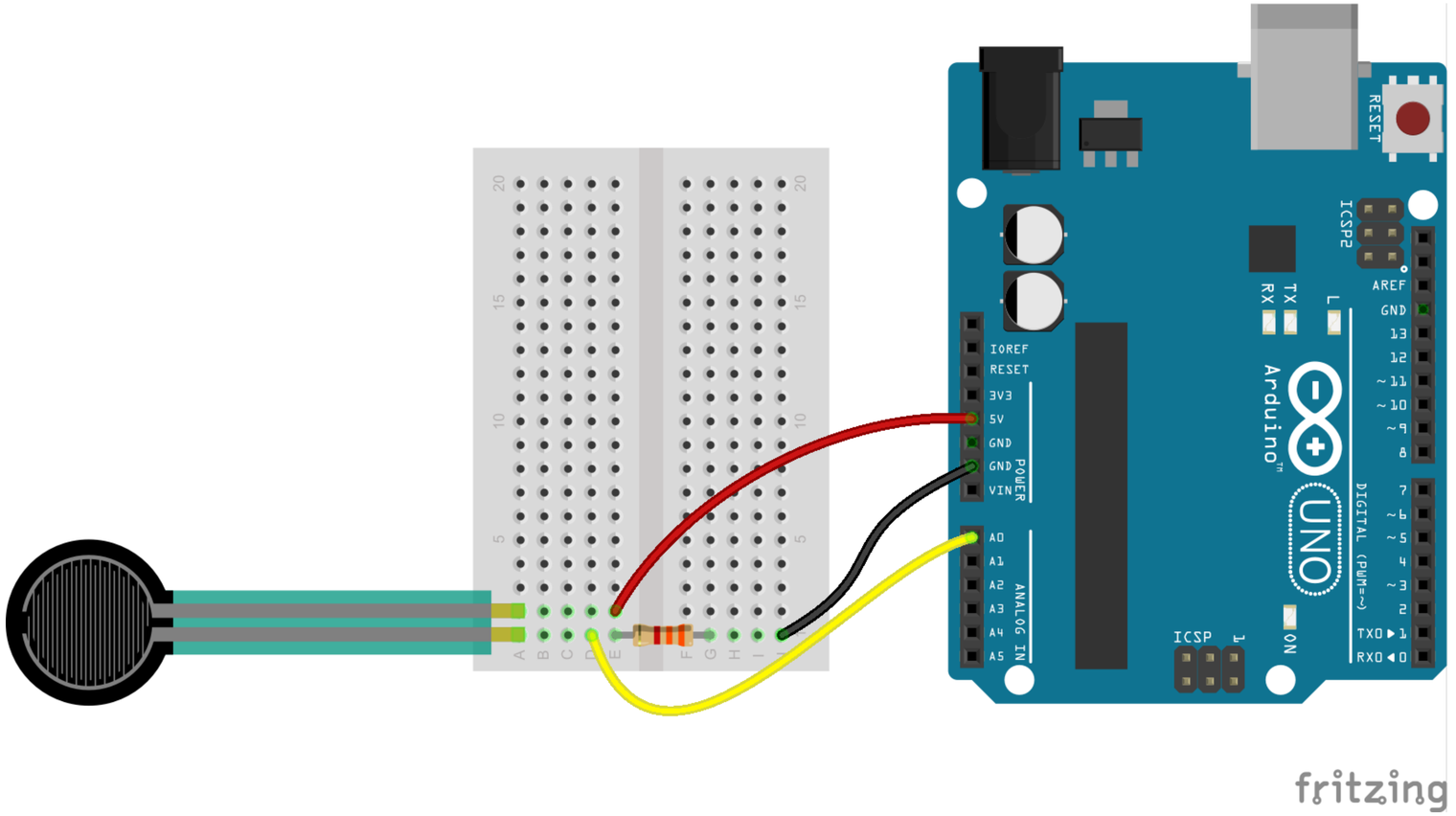
GND ⟷ GND
VCC ⟷ 5V

Servo 1

GND ⟷ GND
5V ⟷ 5V
PWM ⟷ 6

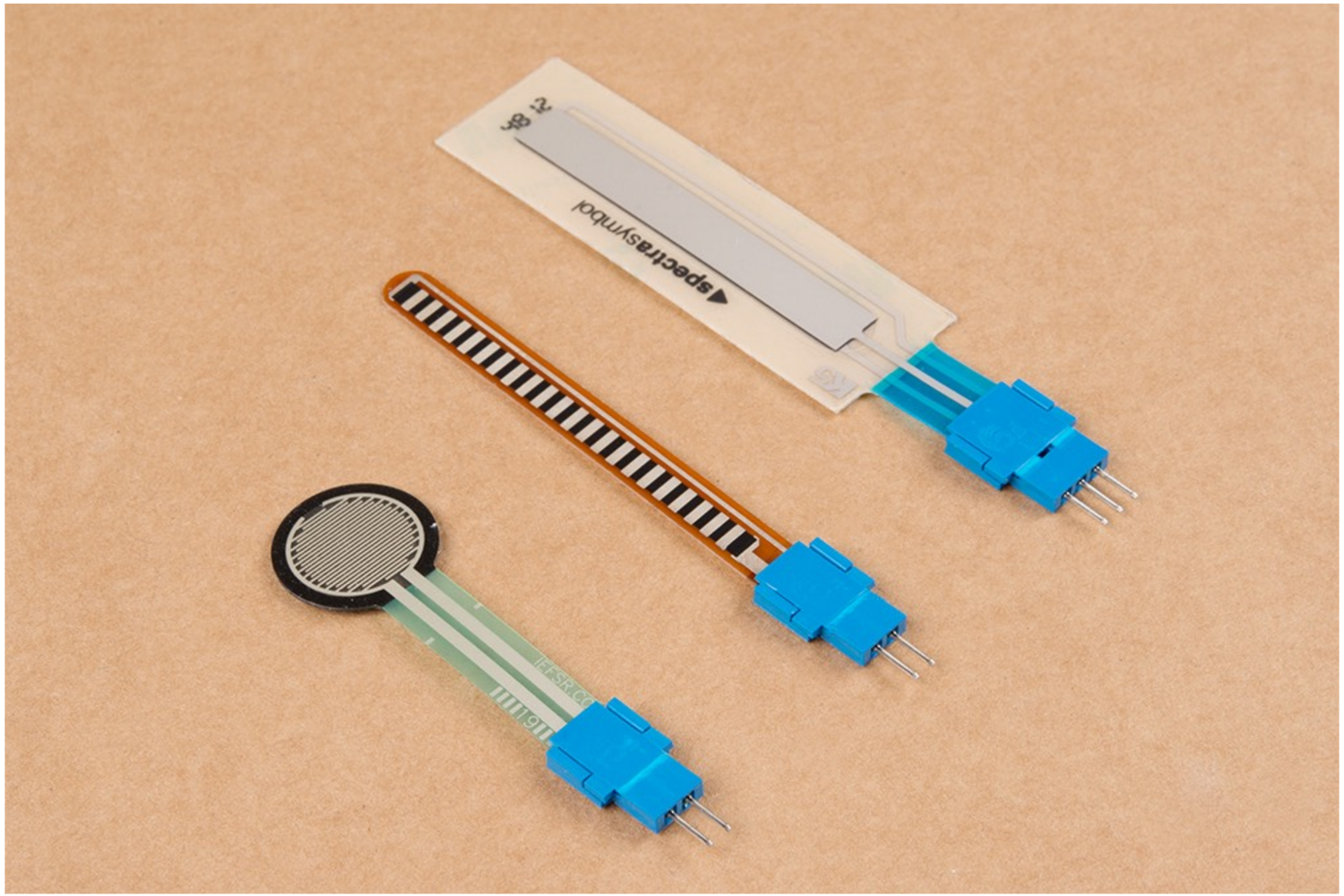
Servo 2

GND ⟷ GND
5V ⟷ 5V
PWM ⟷ 7



[bit.ly/4cARZLE]

Force-Sensitive Resistor (FSR) Sensor



[https://bit.ly/3tsmMWe]

Force Gestures, UIST'11

VRsneaky: Increasing Presence in VR Through Gait-Aware Auditory Feedback

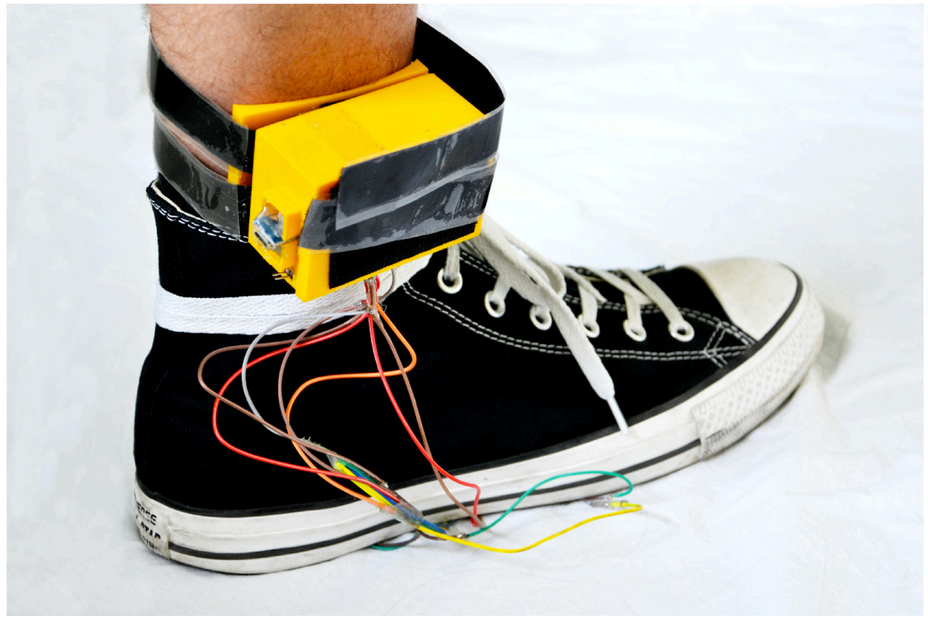
Matthias Hoppe[1], Jakob Karolus[1], Felix Dietz[1], Paweł W. Woźniak[2], Albrecht Schmidt[1], Tonja Machulla[1]

[1] LMU Munich. Munich. Germany. {firstname.lastname}@ifi.lmu.de

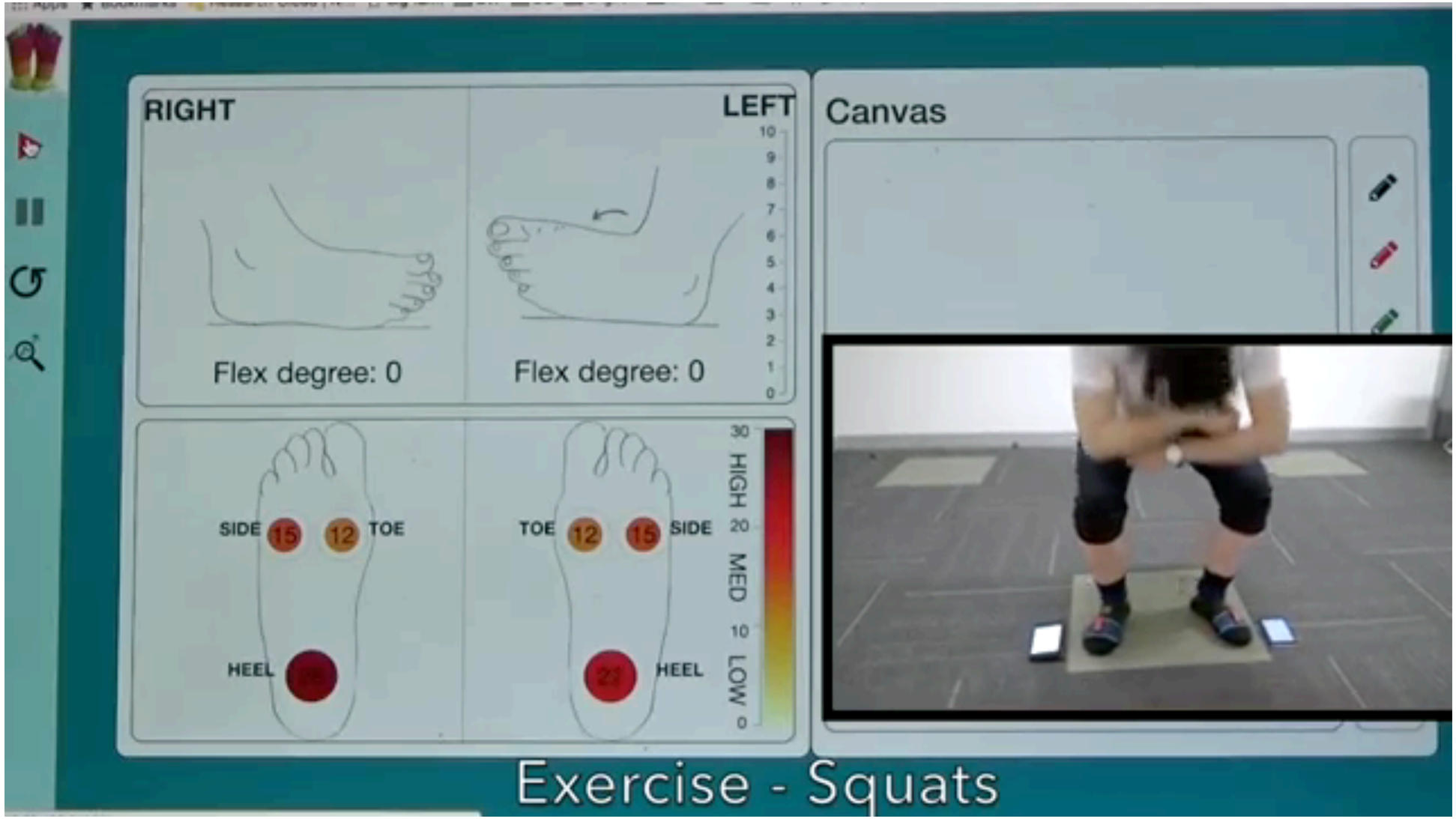
[2] University Utrecht. Utrecht. the Netherlands. p.w.wozniak@uu.nl

VRsneaky, CHI'19

Force Resistive Sensing (FSR)

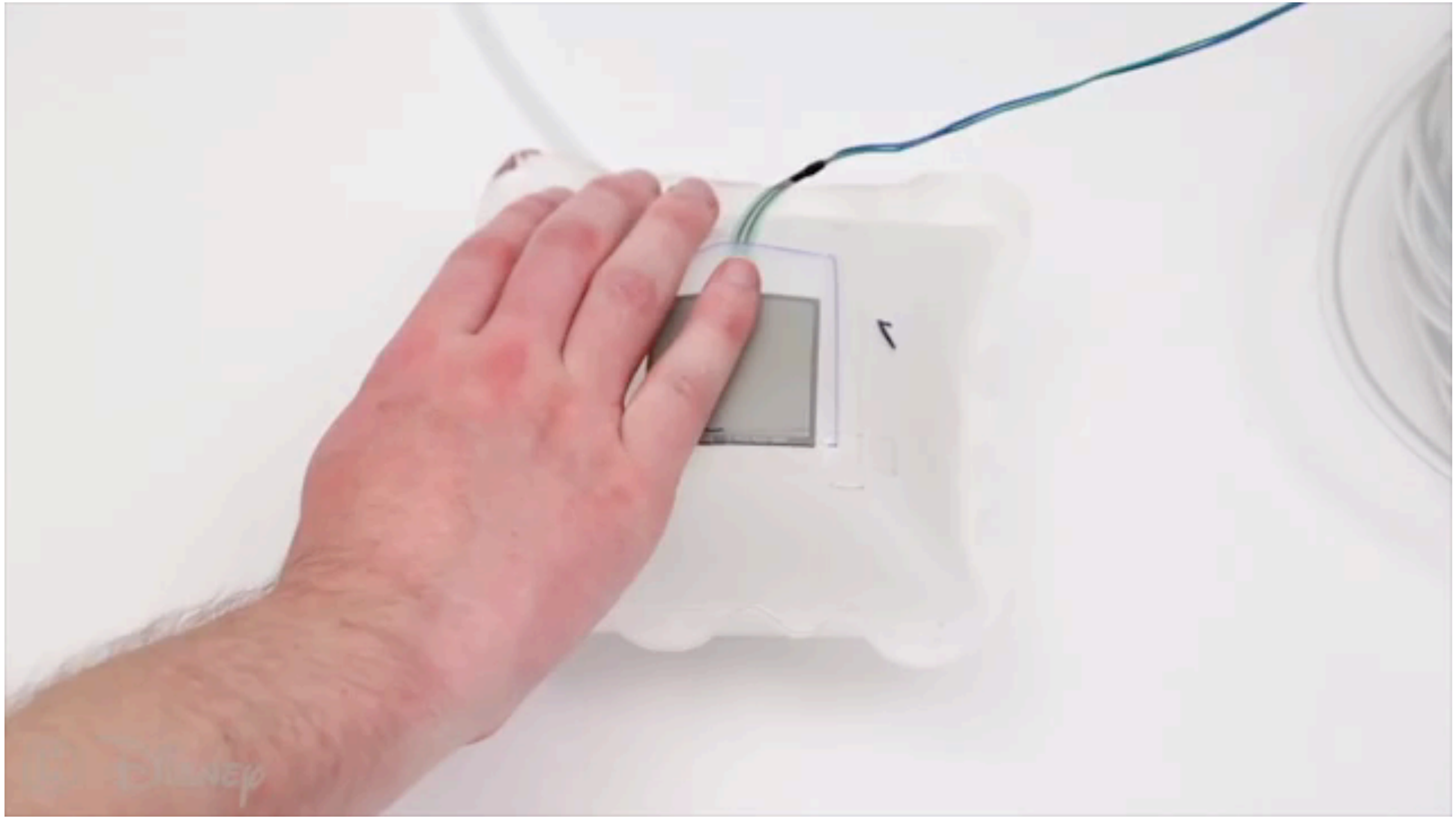


VRsneaky, CHI’19


RIGHT
LEFT
Canvas
Flex degree: 0
Flex degree: 0
10
9
8
7
6
5
4
3
2
1
0
SIDE
15
12
TOE
TOE
12
15
SIDE
HEEL
21
HEEL
30
HIGH
20
MED
10
LOW
0
Exercise - Squats

SoPhy, CHI'17

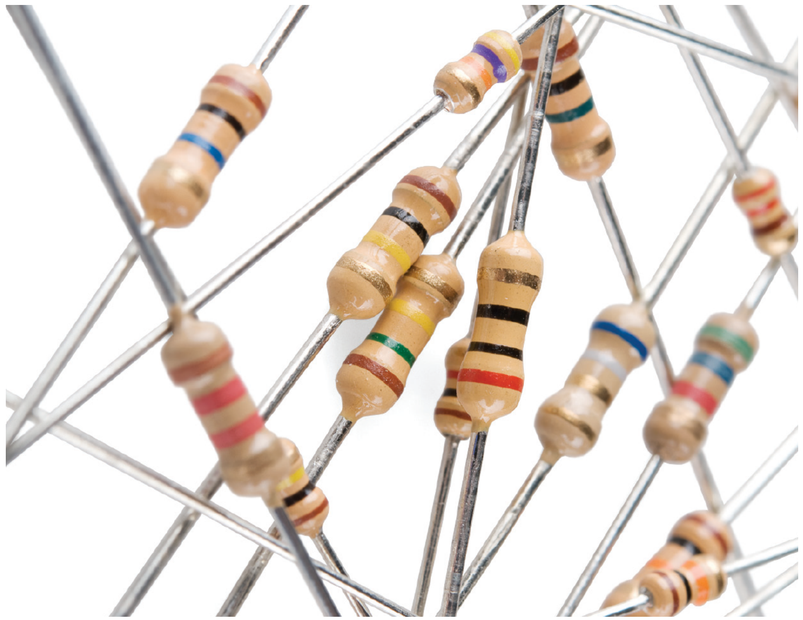
PuPoP, UIST'18

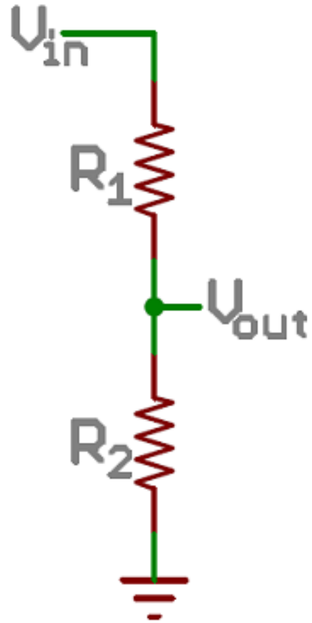


Force Jacket, CHI’18

Force-Sensitive Resistor (FSR) Sensor

- Static resistor
- Voltage Dividers



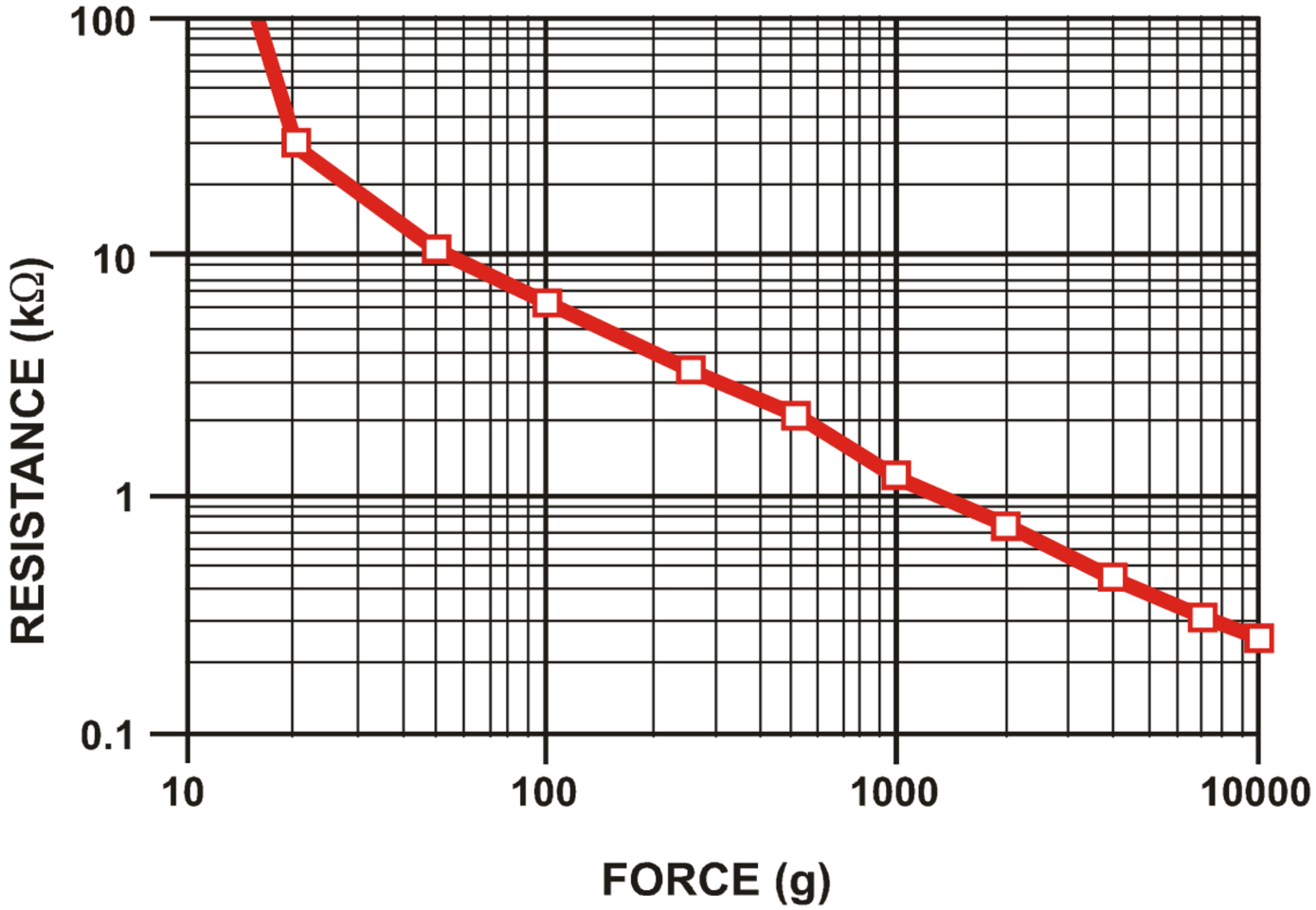


$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

What we need

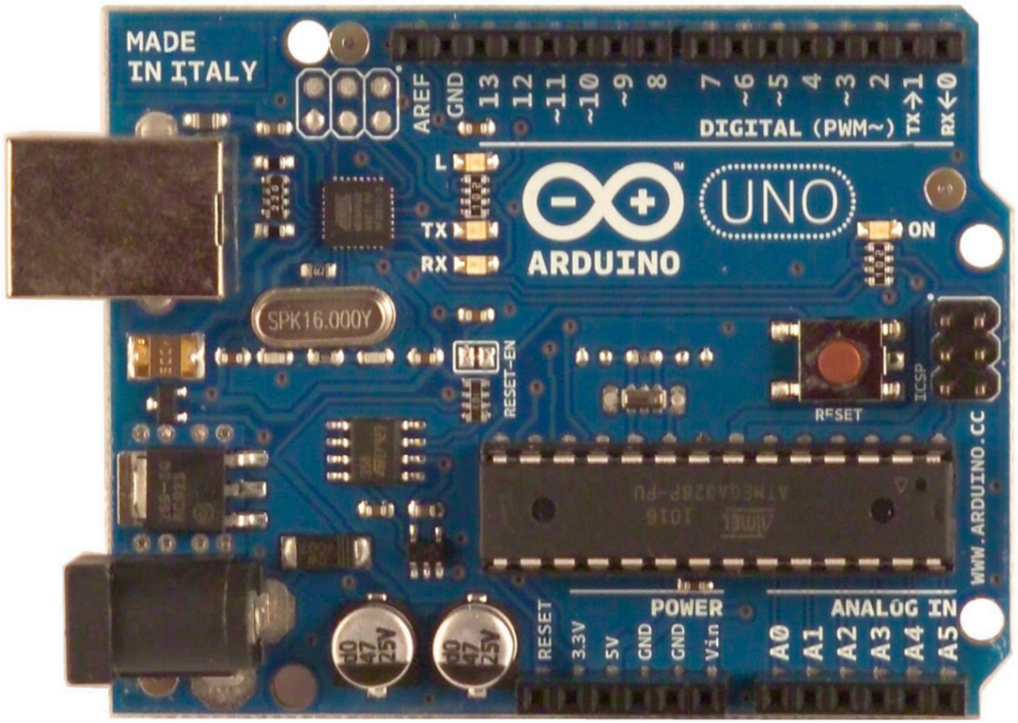
- FSR sensor
- Arduino Uno
- Breadboard & USB cable
- Dupont line (male to male) * 3
- 3.3K ohm resistor * 1

Force v.s. Resistance



Resistance & Conductance

Analog-To-Digital Pin

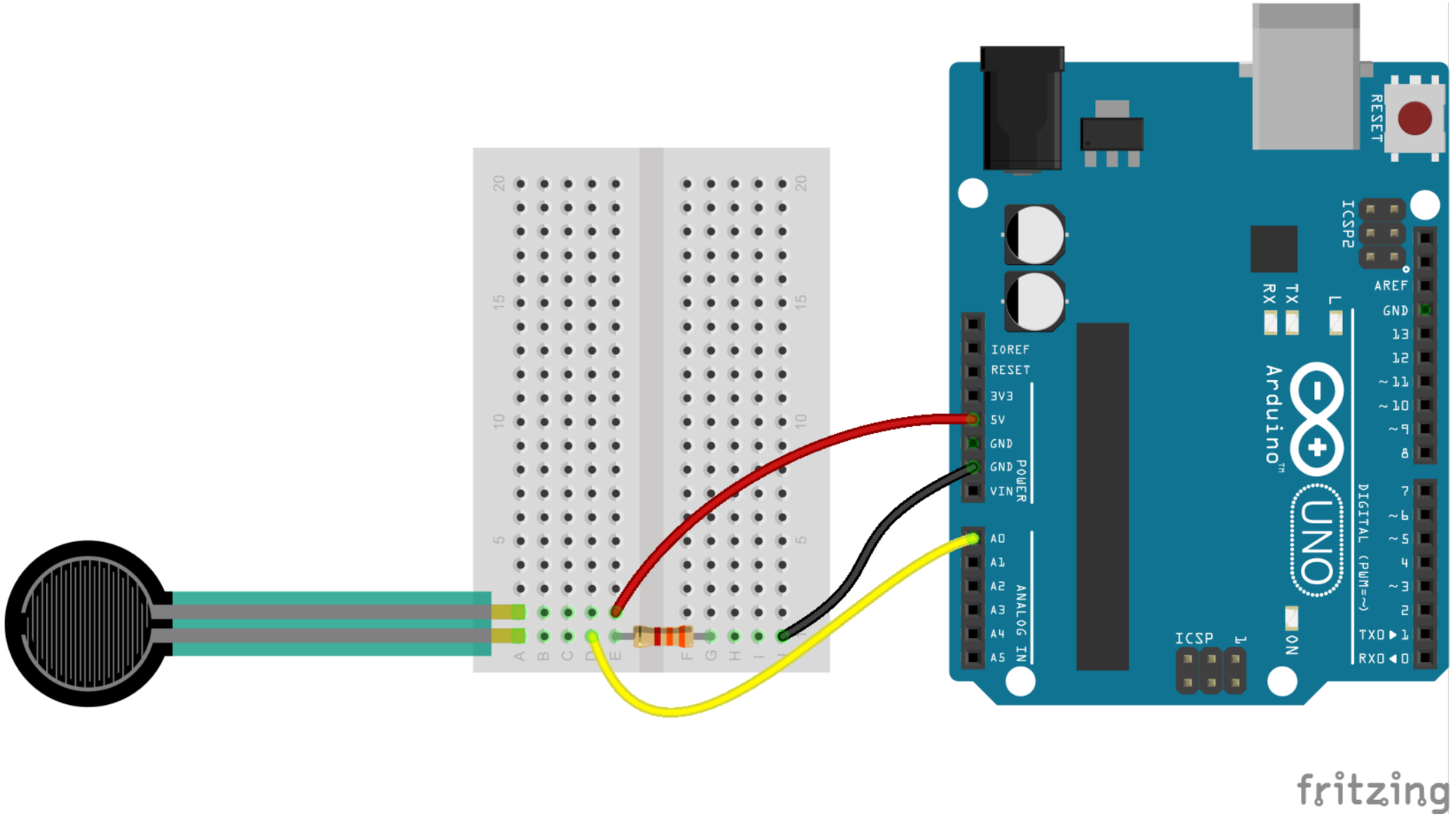


`analogRead(pin_number)`

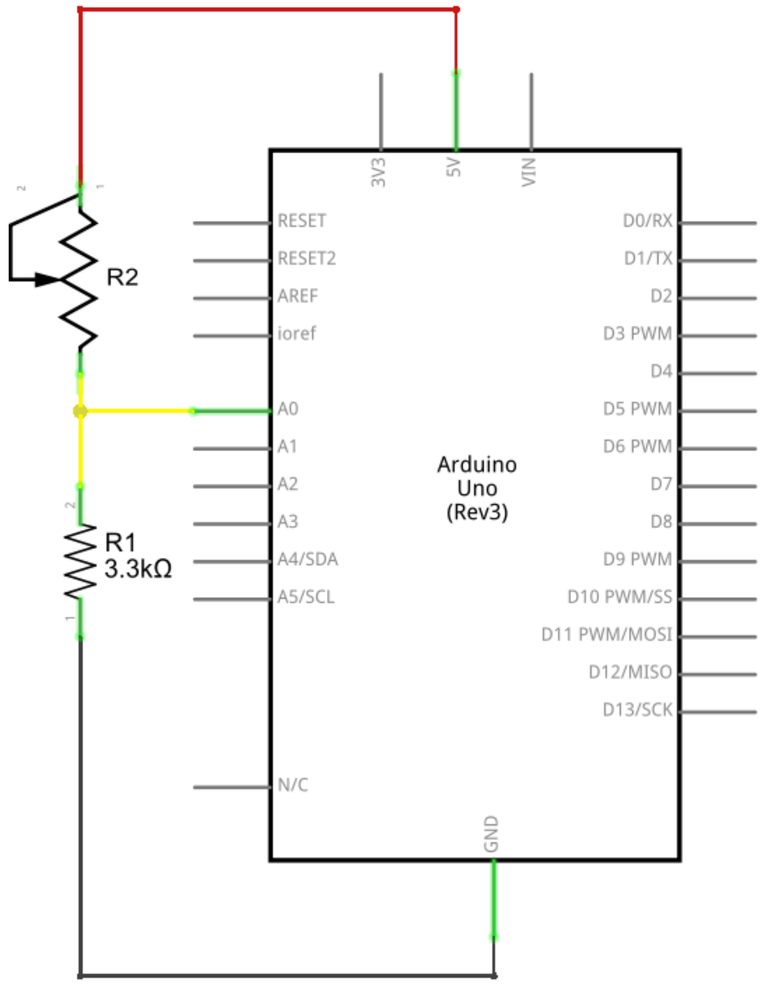
returns 0 if voltage = 0V
returns 1023 if voltage = 5V
…and any value in between

analog pins are **analog input only**

FSR Circuit


RESET
ICSP2
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM=~)
TX
RX
L
ON
ICSP
1
IOREF
RESET
3V3
5V
GND
GND
VIN
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN
UNO
Arduino™
fritzing

FSR Circuit


3V3
5V
VIN
RESET
RESET2
AREF
ioref
A0
A1
A2
A3
A4/SDA
A5/SCL
N/C
GND
D0/RX
D1/TX
D2
D3 PWM
D4
D5 PWM
D6 PWM
D7
D8
D9 PWM
D10 PWM/SS
D11 PWM/MOSI
D12/MISO
D13/SCK
Arduino
Uno
(Rev3)
R2
R1
3.3kΩ
fritzing

FSR Code

- Force_Sensitive_Resistor_Example.ino
  - Upload it and open the serial monitor (9600 baud rate)
  - Press FSR sensor

[https://bit.ly/3CWr37D]

FSR Code

- Force_Sensitive_Resistor_Example.ino

```
const int FSR_PIN = A0; // Pin connected to FSR/resistor divider

// Measure the voltage at 5V and resistance of your 3.3k resistor, and enter
// their value's below:
const float VCC = 4.98; // Measured voltage of Ardunio 5V line
const float R_DIV = 3230.0; // Measured resistance of 3.3k resistor

void setup()
{
  Serial.begin(9600);
  pinMode(FSR_PIN, INPUT);
}
```

FSR Code

- Force_Sensitive_Resistor_Example.ino

```
void loop()
{
  int fsrADC = analogRead(FSR_PIN);
  // If the FSR has no pressure, the resistance will be
  // near infinite. So the voltage should be near 0.
  if (fsrADC != 0) // If the analog reading is non-zero
  {
    // Use ADC reading to calculate voltage:
    float fsrV = fsrADC * VCC / 1023.0;
    // Use voltage and static resistor value to
    // calculate FSR resistance:
    float fsrR = R_DIV * (VCC / fsrV - 1.0);
    Serial.println("Resistance: " + String(fsrR) + " ohms");
```

analog pin
(0 ~ 1023)

FSR Code

- Force_Sensitive_Resistor_Example.ino

```
void loop()
{
  int fsrADC = analogRead(FSR_PIN);
  // If the FSR has no pressure, the resistance will be
  // near infinite. So the voltage should be near 0.
  if (fsrADC != 0) // If the analog reading is non-zero
  {
    // Use ADC reading to calculate voltage:
    float fsrV = fsrADC * VCC / 1023.0;
    // Use voltage and static resistor value to
    // calculate FSR resistance:
    float fsrR = R_DIV * (VCC / fsrV - 1.0);
    Serial.println("Resistance: " + String(fsrR) + " ohms");
```

FSR Code

- Force_Sensitive_Resistor_Example.ino

**** Voltage Dividers ****

Vout = Vcc * R / (R + Rfsr)

→ Rfsr = R (Vcc / Vout - 1)

```
void loop()
{
  int fsrADC = analogRead(FSR_PIN);
  // If the FSR has no pressure, the resistance will be
  // near infinite. So the voltage should be near 0.
  if (fsrADC != 0) // If the analog reading is non-zero
  {
    // Use ADC reading to calculate voltage:
    float fsrV = fsrADC * VCC / 1023.0;
    // Use voltage and static resistor value to
    // calculate FSR resistance:
    float fsrR = R_DIV * (VCC / fsrV - 1.0);
    Serial.println("Resistance: " + String(fsrR) + " ohms");
```

[https://bit.ly/3iFXtKl]

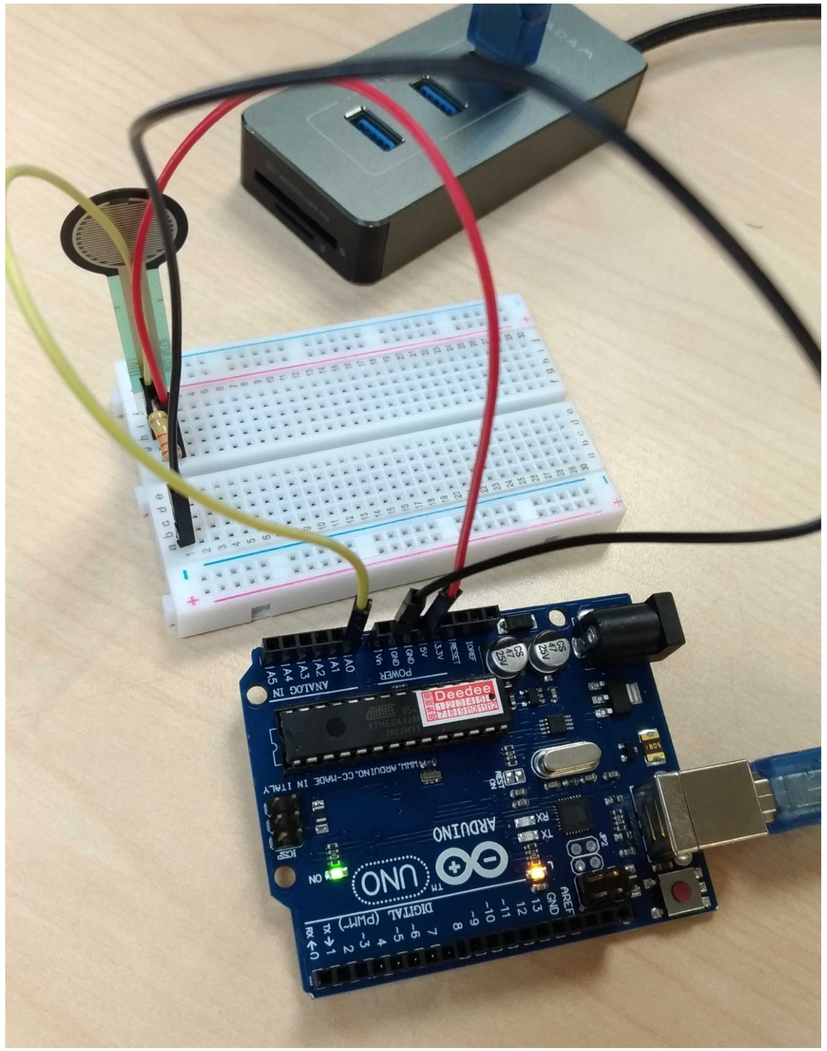
FSR Code

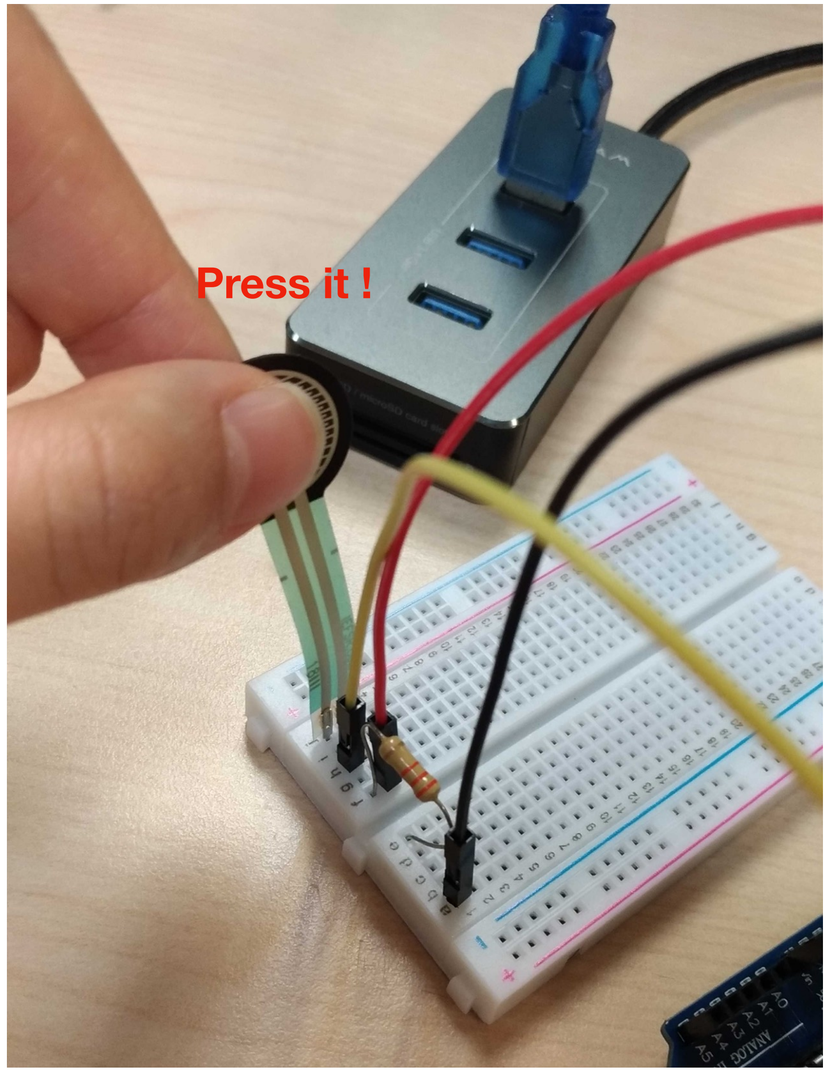
- Force_Sensitive_Resistor_Example.ino

```
  // Guesstimate force based on slopes in figure 3 of
  // FSR datasheet:
  float force;
  float fsrG = 1.0 / fsrR; // Calculate conductance
  // Break parabolic curve down into two linear slopes:
  if (fsrR <= 600)
    force = (fsrG - 0.00075) / 0.00000032639;
  else
    force =  fsrG / 0.000000642857;
  Serial.println("Force: " + String(force) + " g");
  Serial.println();

  delay(500);
}
```

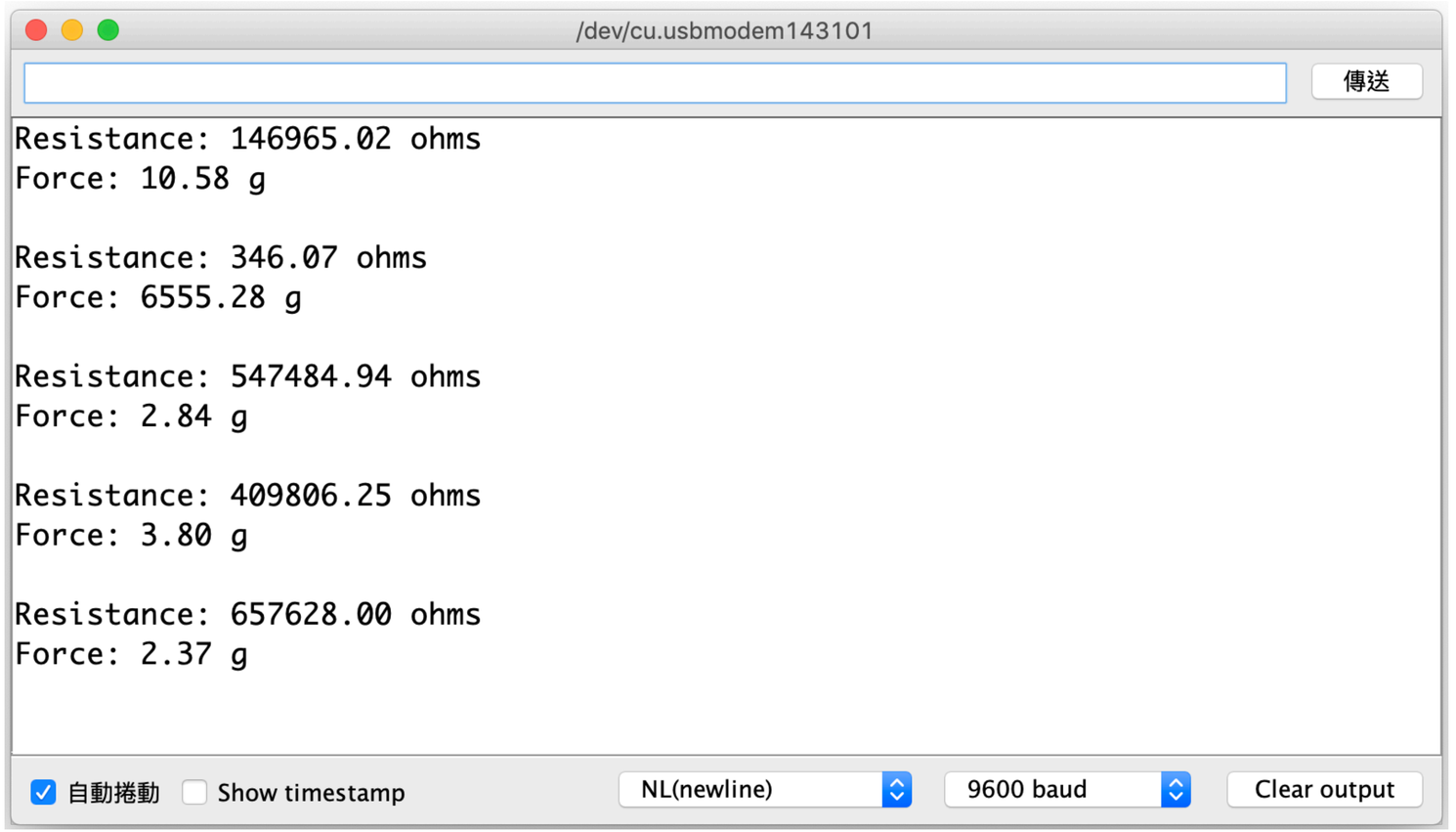
$$G = 1/R$$

FSR Code





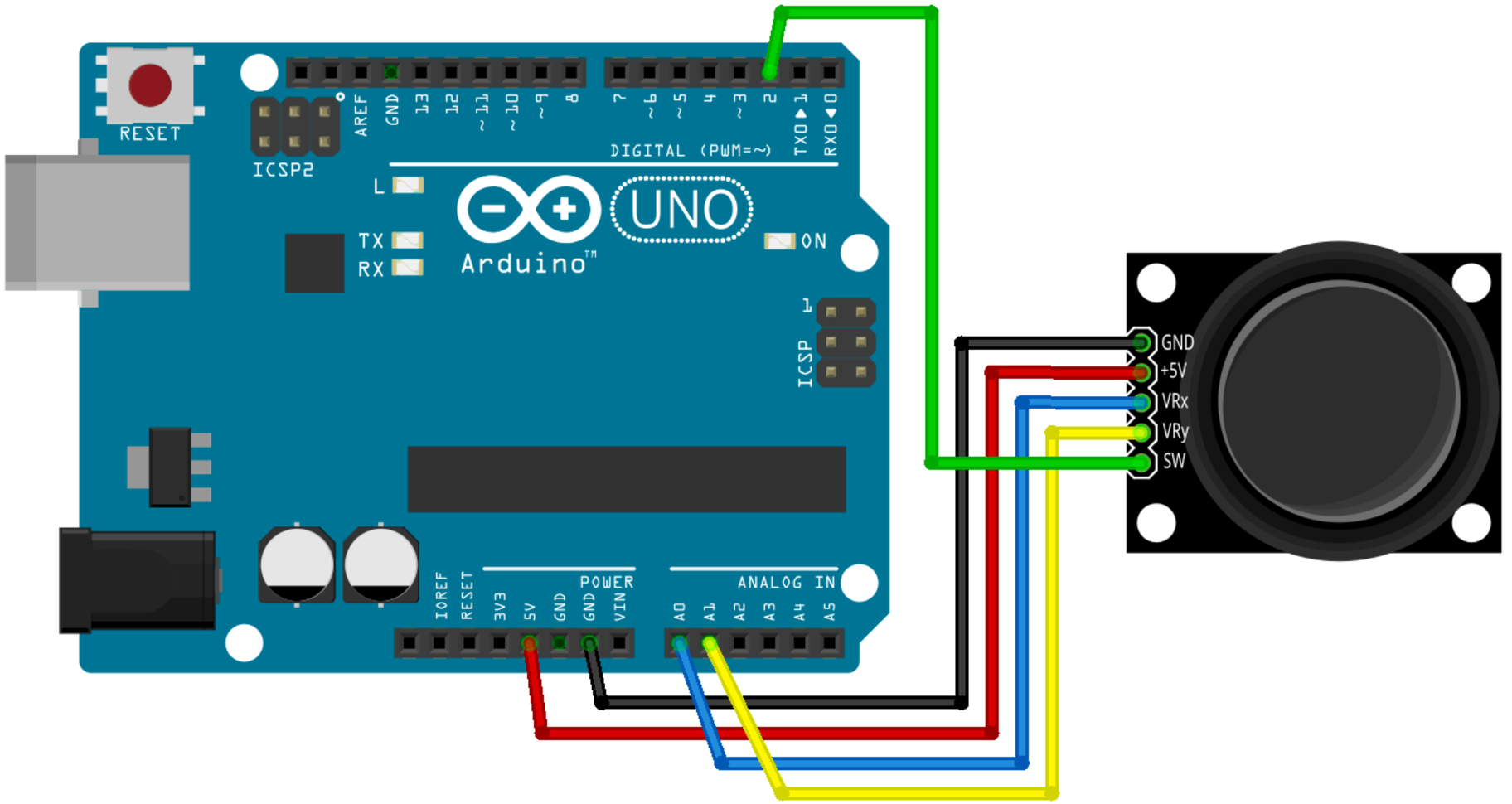
FSR Code


/dev/cu.usbmodem143101
傳送
Resistance: 146965.02 ohms
Force: 10.58 g
Resistance: 346.07 ohms
Force: 6555.28 g
Resistance: 547484.94 ohms
Force: 2.84 g
Resistance: 409806.25 ohms
Force: 3.80 g
Resistance: 657628.00 ohms
Force: 2.37 g
自動捲動
Show timestamp
NL(newline)
9600 baud
Clear output

FSR in Unity

FSR Arduino code : https://bit.ly/3rEQtC3

Joystick


RESET
ICSP2
AREF
GND
13
12
~11
~10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
DIGITAL (PWM=~)
L
TX
RX
Arduino™
UNO
ON
ICSP
IOREF
RESET
3V3
5V
GND
GND
VIN
POWER
A0
A1
A2
A3
A4
A5
ANALOG IN
GND
+5V
VRx
VRy
SW
fritzing

Joystick

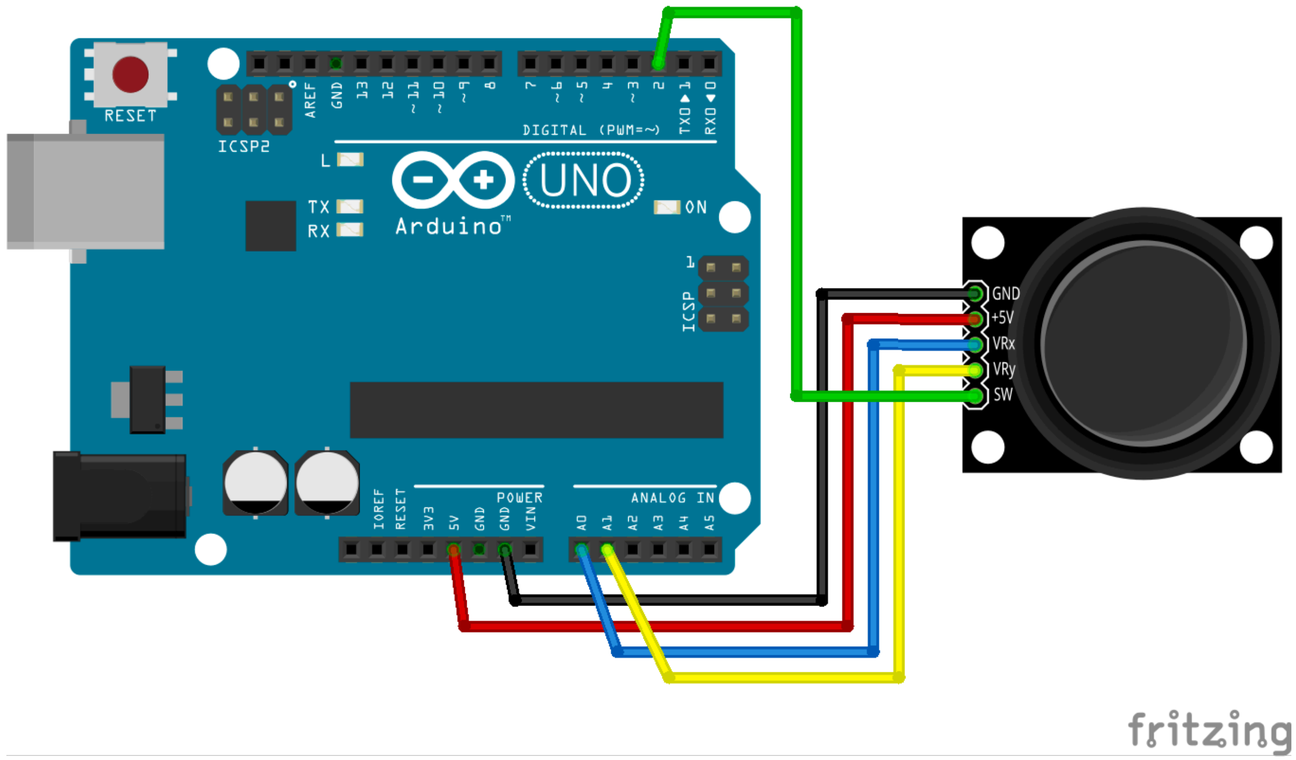
```
void setup() {
  Serial.begin(115200);
  pinMode(2,INPUT);
}

void loop() {
  int vrx, vry, sw;

  vrx = analogRead(A0);
  vry = analogRead(A1);
  sw = digitalRead(2);

  Serial.print("VRx=");
  Serial.print(vrx);
  Serial.print(",VRy=");
  Serial.println(vry);
  Serial.print("SW= ");
  Serial.println(sw);
  delay(100);
}
```

[https://bit.ly/3ClBUGV]

Joystick Servo

[https://bit.ly/3Fi8qMC]

Joystick and Servo



Joystick

GND	⟷	GND
5V	⟷	5V
VRx	⟷	A0
VRy	⟷	A1
SW	⟷	2

Servo 1

GND	⟷	GND
5V	⟷	5V
PWM	⟷	6

Servo 2

GND	⟷	GND
5V	⟷	5V
PWM	⟷	7

[bit.ly/4abPNs4]

Servo Motor Control
(外接電源)