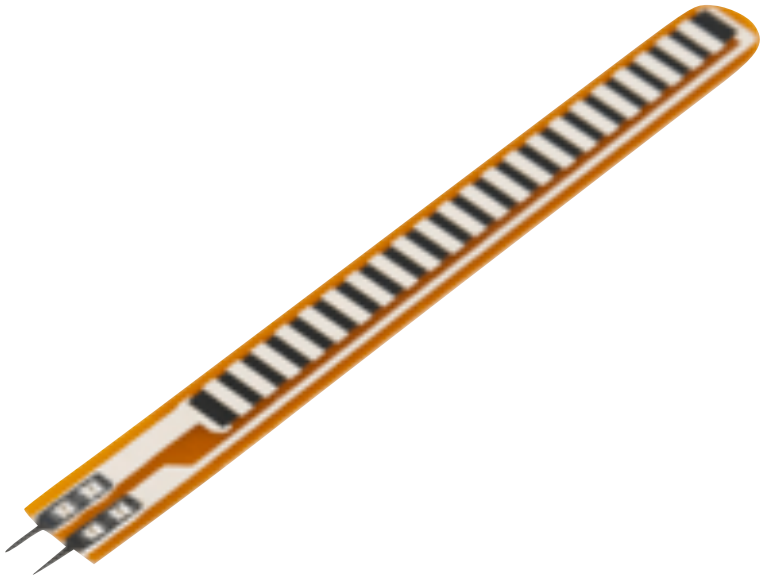
Timeless Lifeskills
FOUNDATION
www.TimelessLifeskills.org

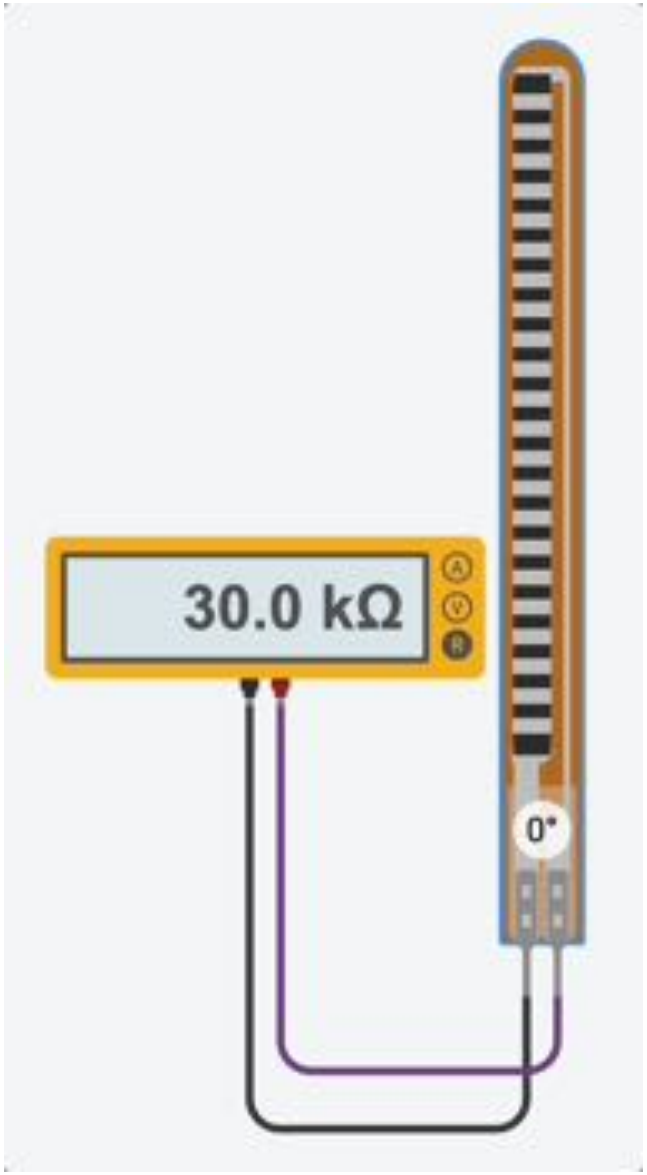
ARDUINO
Analog Sensors
Instructor Guide

PROJECTS

1. Flex Sensor to control Servo
2. Force Sensor to control Servo
3. MQ2 Smoke Sensor Alarm - Analog Pin Out
4. MQ2 Smoke Sensor - Digital Pin Out



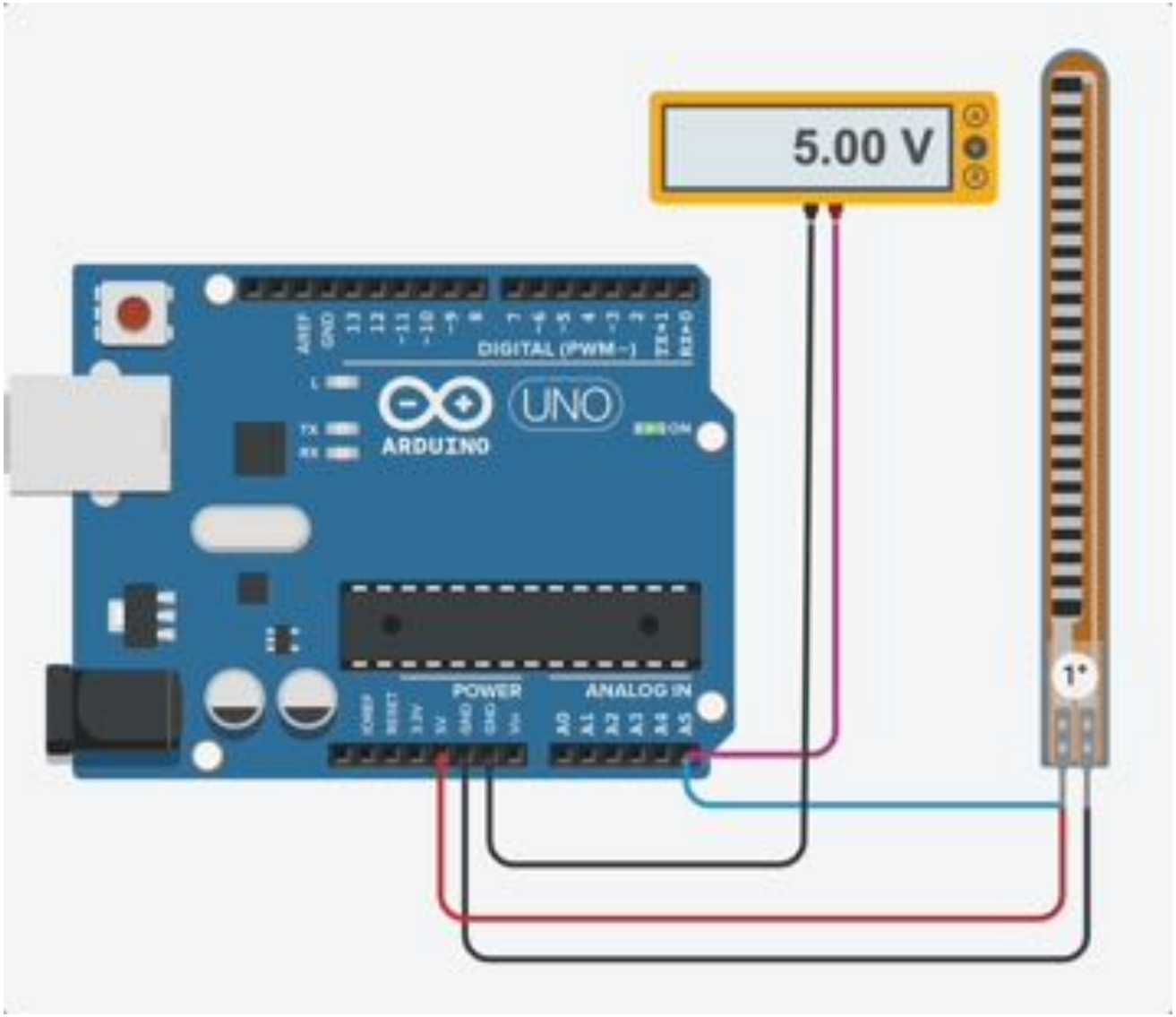
Flex Sensor



The Flex sensor is essentially a variable resistor.

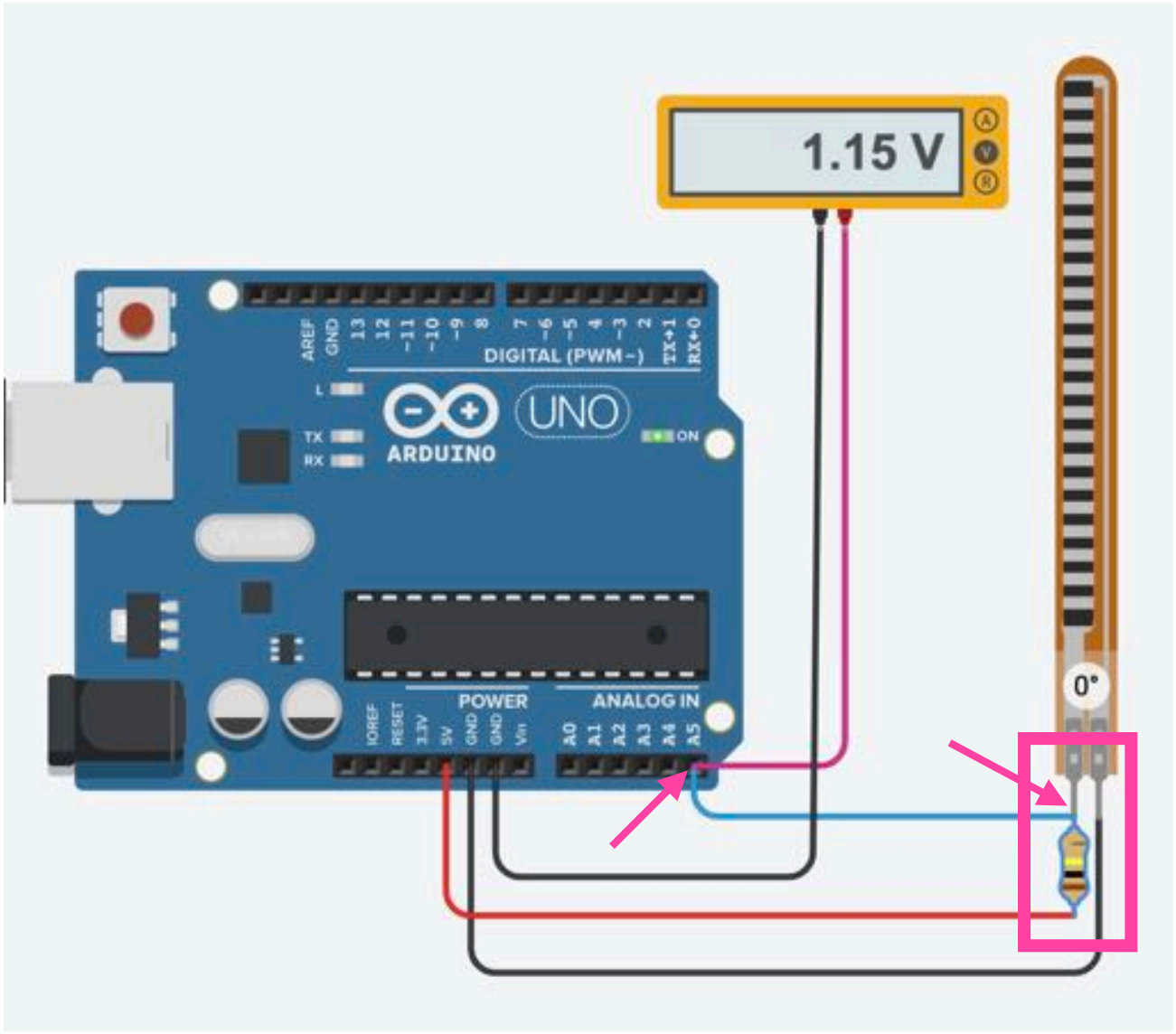
As the sensor is flexed (bent), the resistance across the sensor increases.

Note that, as the flex sensor is bent, the range of resistance varies from 30 kilohm to 160 kilohm.

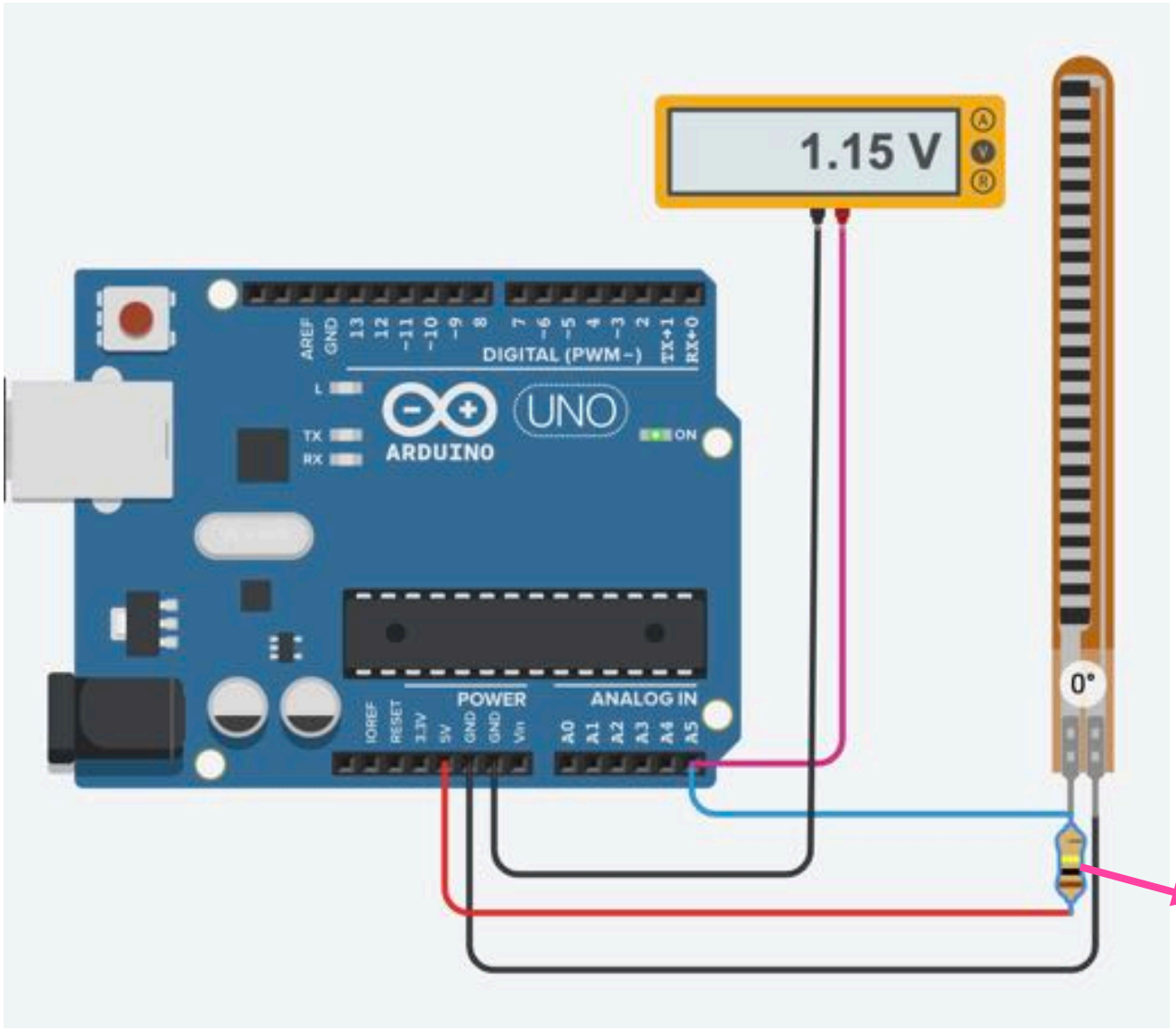


If we connect the Flex Sensor directly to the Arduino (to an analog pin because flex sensor generates an analog signal not a digital signal), even when we bend the sensor, there is no change in voltage.

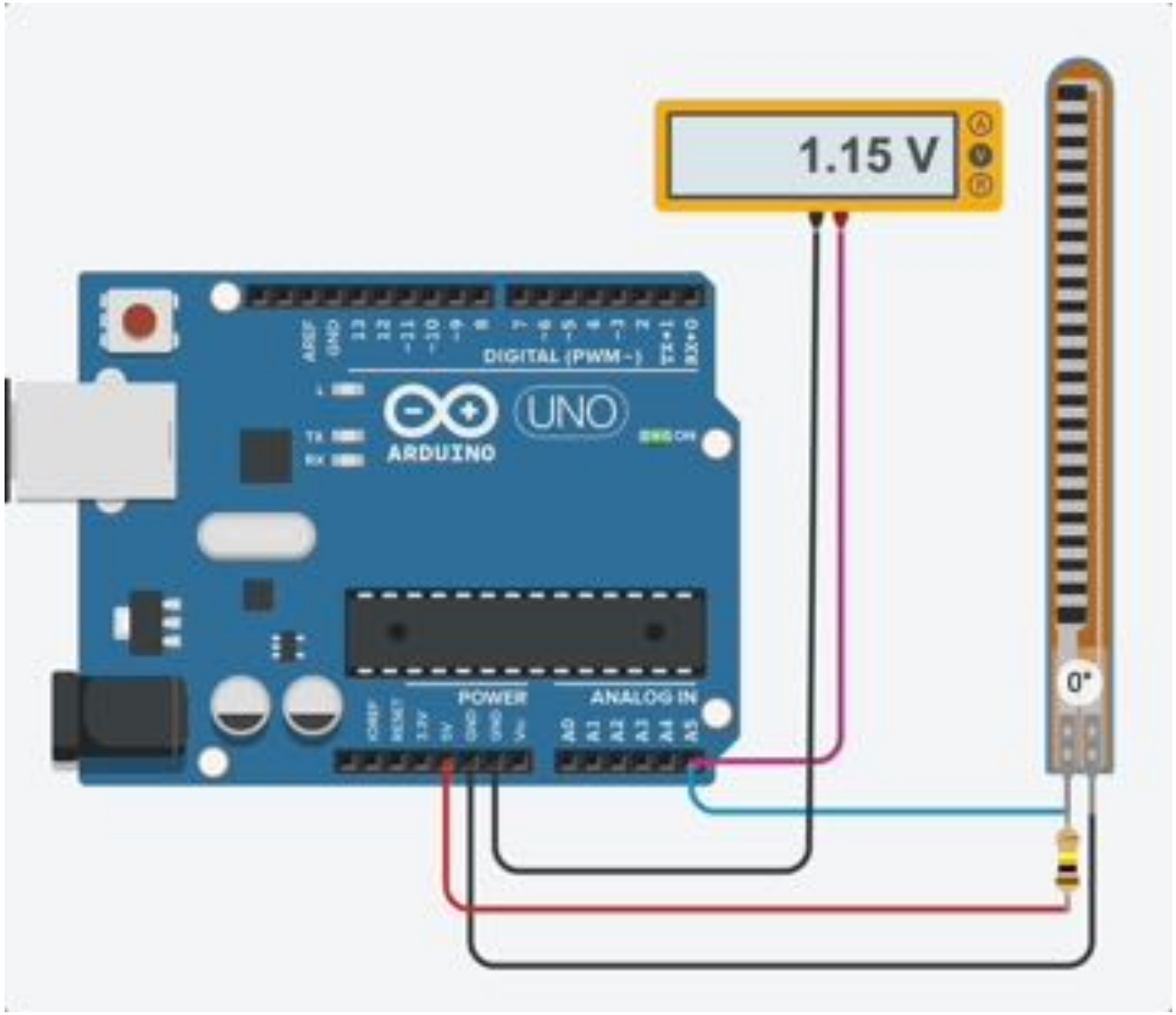
Hence, this is of no use to us because only when there is a change in voltage, as we bend the sensor, can we write a programme to do something with the sensor as it is bent.



However, if we add a resistor in series with the flex sensor and then connect the flex sensor to the Arduino (from the middle of the flex sensor and resistor), we can generate a change in voltage that can be read by the Arduino using the in-built Analog to Digital Converter (ADC).

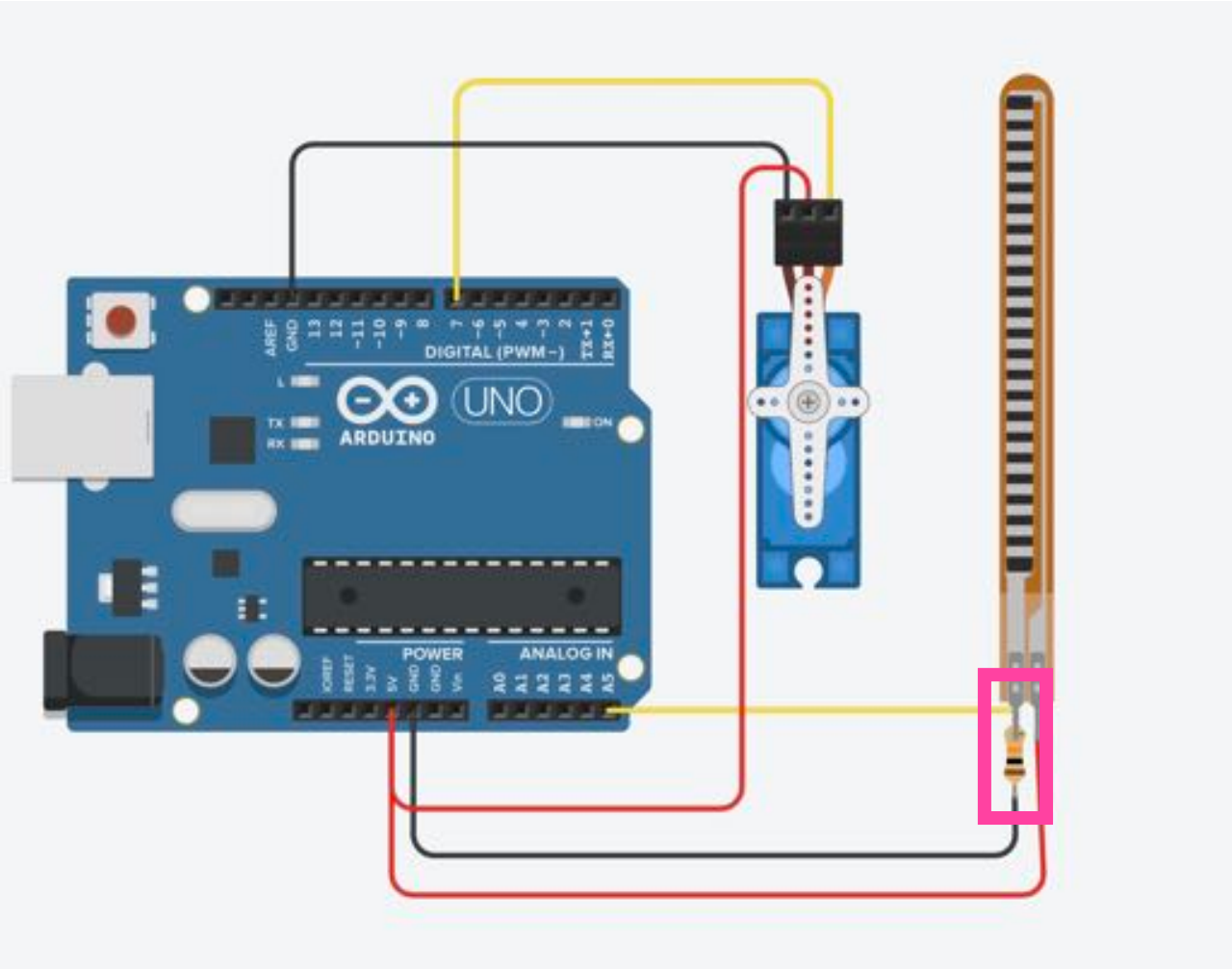


Since the range of resistance of the flex sensor is between 30-160 kΩ (as we saw earlier), if we add a resistor that is somewhere in the middle of this, around 100 kΩ, the range of voltage generated by this circuit will be enough for the ADC in the Arduino to read.



In this circuit, the voltage change as we bend the flex sensor ranges from around 1 volt to around 3 volt, a swing of 2 volt, which the Arduino ADC will be able to distinguish.

Now, we can write a programme where decision-making can be done as the flex sensor is bent. For example, turn a servo, sound an alarm, or light a LED as the sensor is bent.



Summary: Voltage Divider

The Flex sensor is essentially a variable resistor. As the sensor is flexed (bent), the resistance across the sensor increases.

Devices like Arduino, that have an ADC (analog to digital converter), are good at detecting changes in voltage but not that good at detecting changes in resistance.

However, by adding another resistor to the flex sensor, in series, we can create a Voltage Divider. Then, by reading the change in the output of the voltage divider, we can write a programme that can do decision-making based on how much the flex sensor is being bent.

Flex Sensor based Robotic Arm

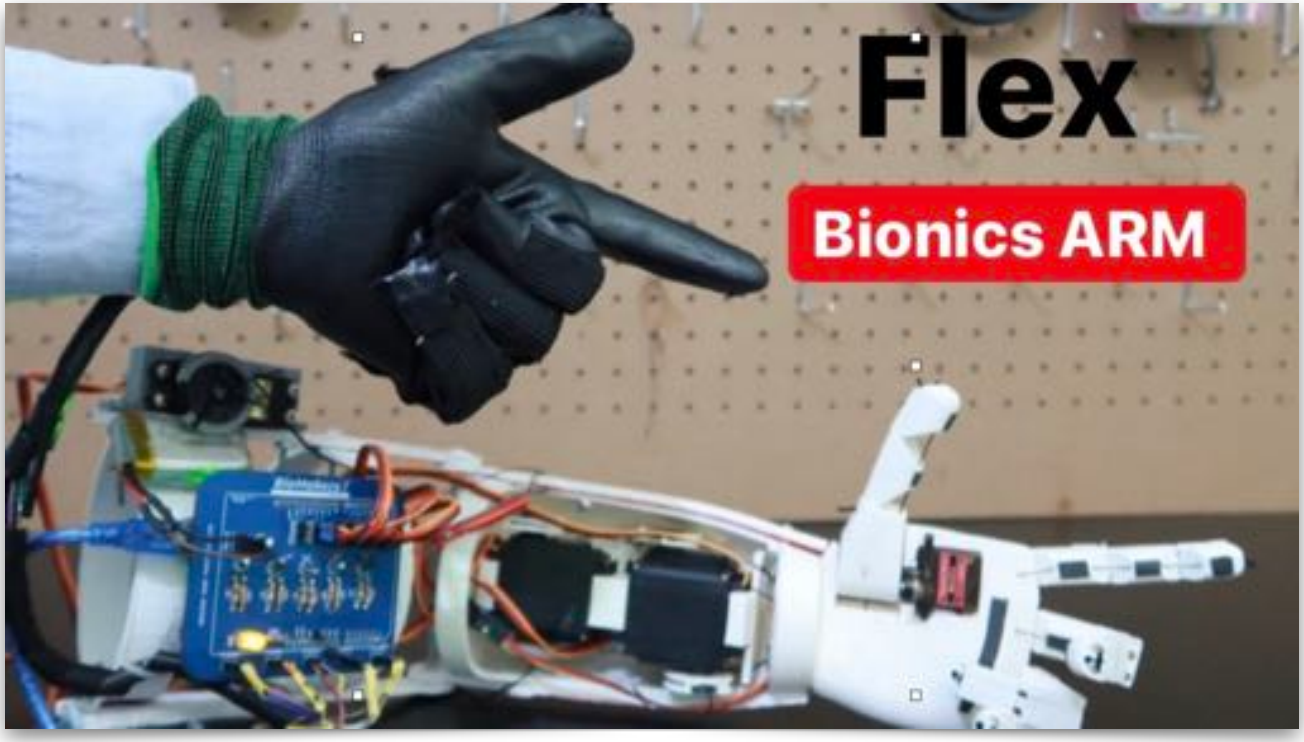
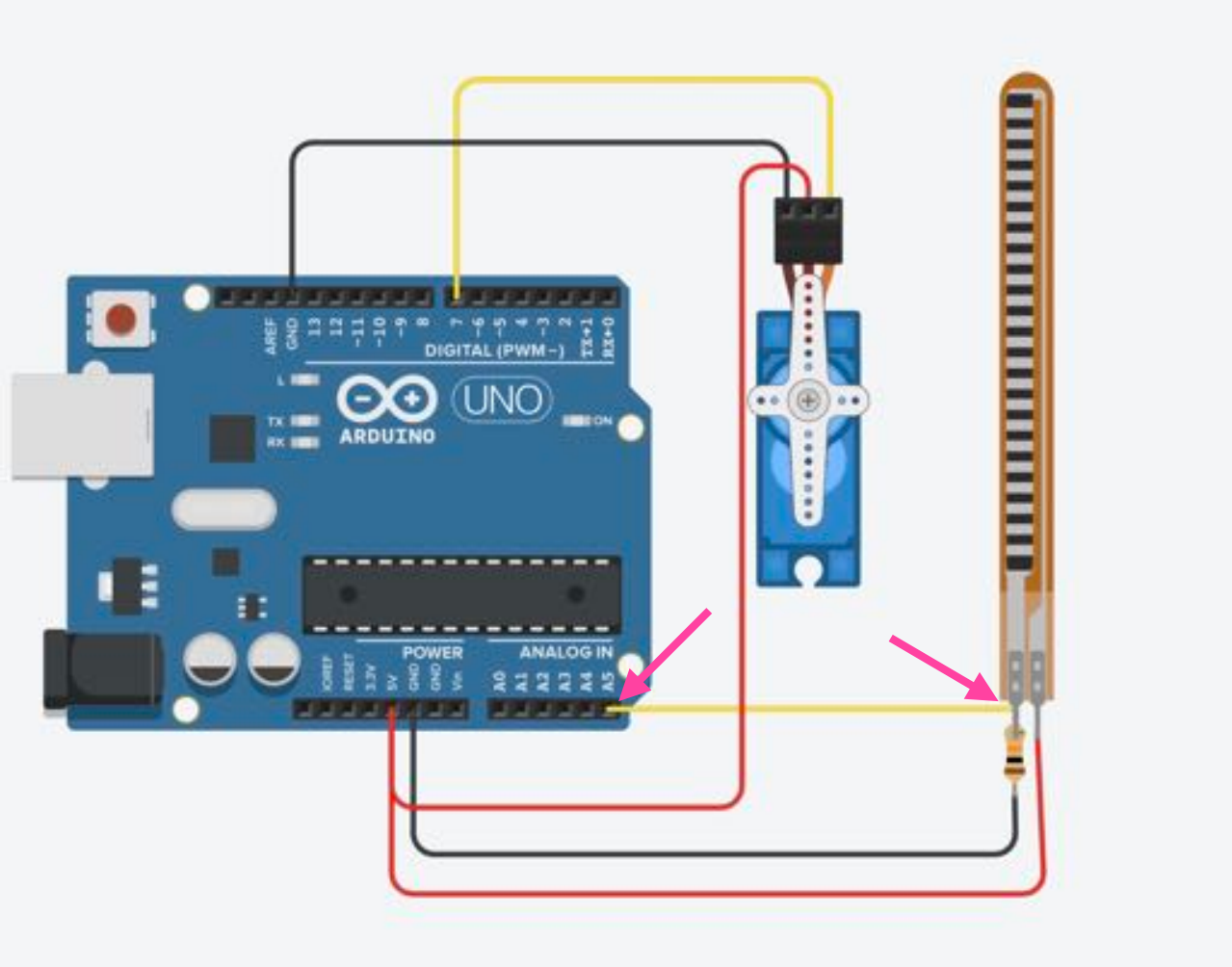


Image source: https://youtu.be/SIxfbKCXKbY

Flex sensors can be used to create Robotic Arms. Flex sensors on a glove determine which finger you have bent and by how much and this triggers a set of servo motors in the robotic arm to move.

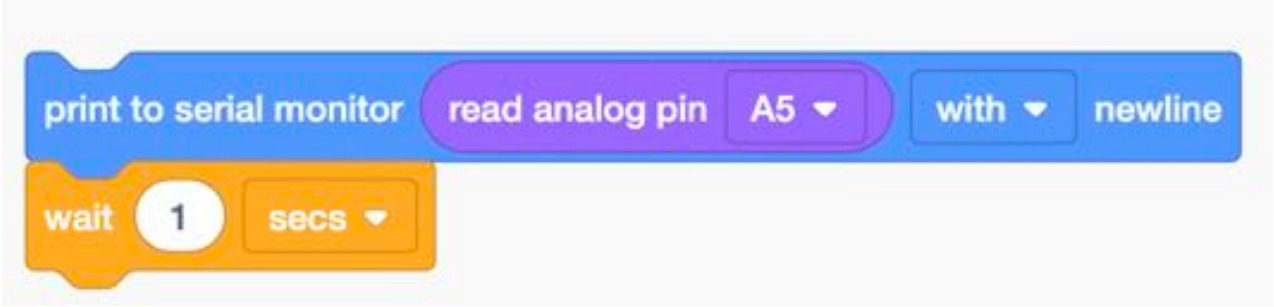
Robotic Arms are used in surgery, to perform dangerous operations like defusing a bomb, or to lift heavy objects.

Timeless Lifeskills FOUNDATION

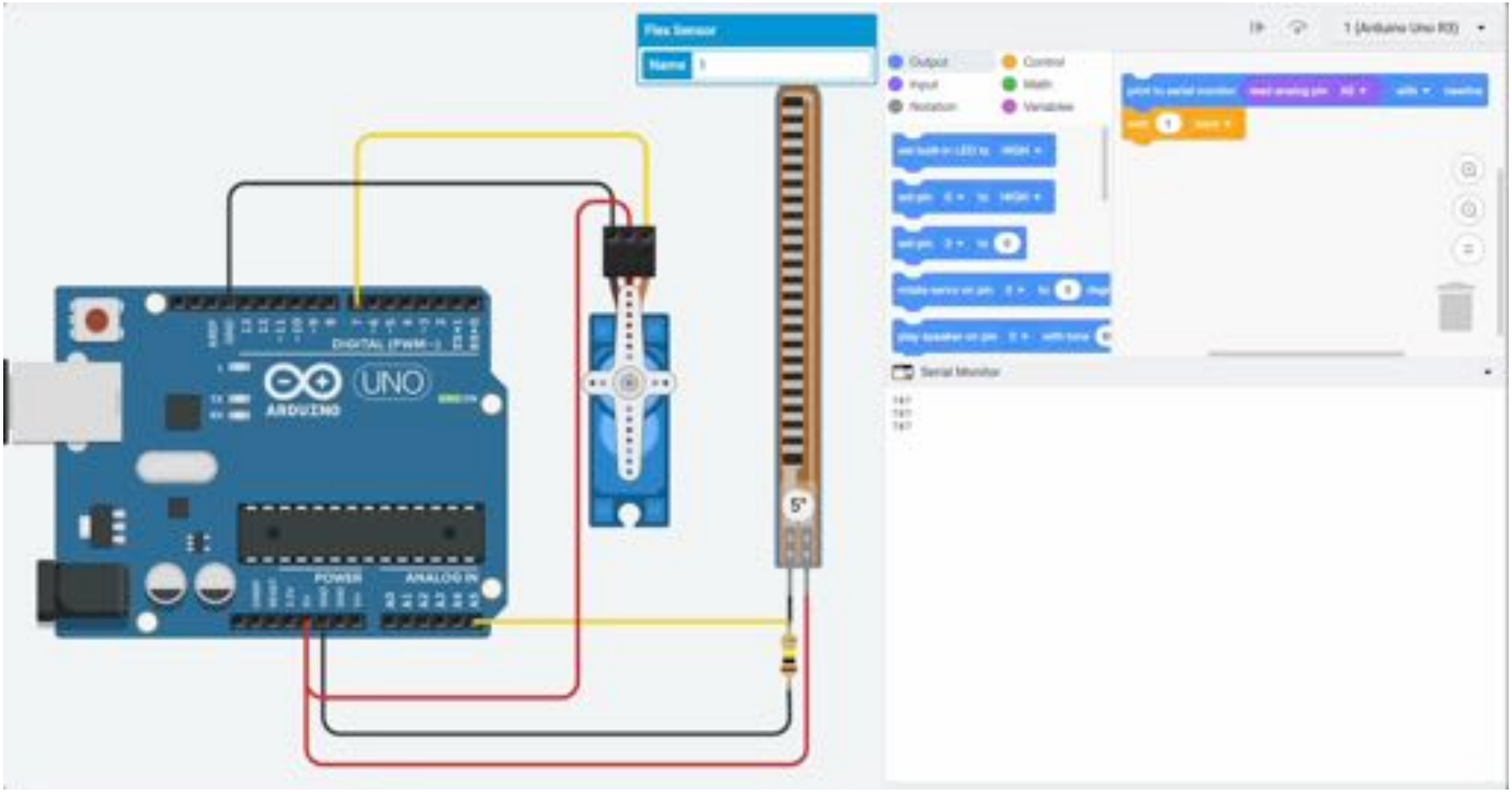


Controlling a Servo with Flex Sensor

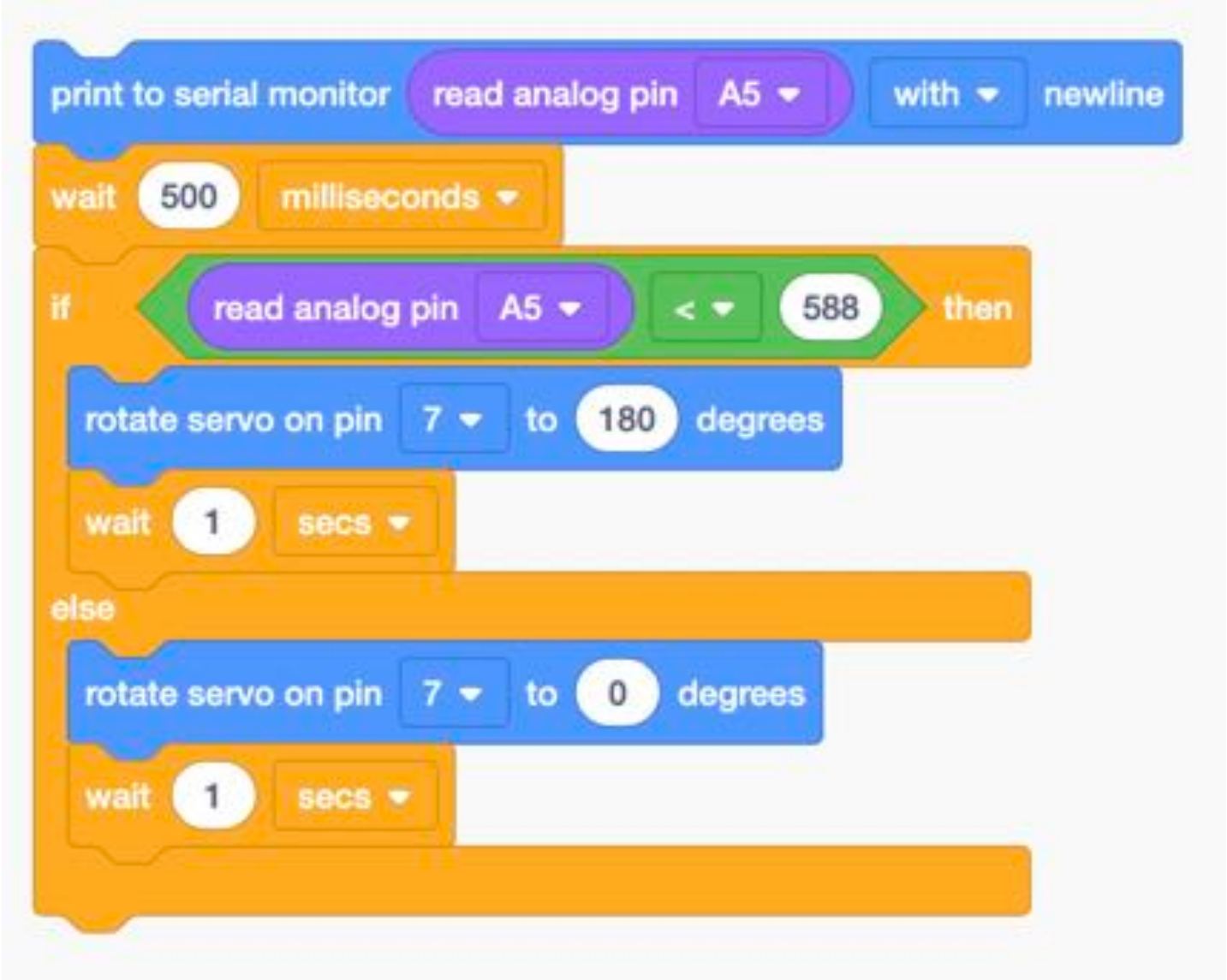
1. Connect one terminal of the flex sensor to Arduino 5V pin.
2. To the other terminal of the flex sensor, attach a resistor (between 10-100 kΩ). You can experiment and see how many kΩ give you maximum range of output voltage.
3. Connect the other end of the resistor to Arduino GND pin.
4. From middle of the flex sensor and resistor, connect a wire that goes to one of the Analog pins of the Arduino (A5 in this case).
5. Connect the Servo as shown.



To find out the range of values generated by the ADC, as the flex sensor is bent, you use the Print to Serial Monitor command and Read the value of the Analog pin to which the flex sensor is connected (A5).



In this case, when the flex sensor is straight, the value generated is 787 and when it is bent the value generated is 389. We want to trigger some action, say turn a servo, when the flex sensor is around half-bent, i.e. the value is around 588. Let's write the code for this.


print to serial monitor
read analog pin A5
with
newline
wait 500 milliseconds
if
read analog pin A5
<
588
then
rotate servo on pin 7 to 180 degrees
wait 1 secs
else
rotate servo on pin 7 to 0 degrees
wait 1 secs

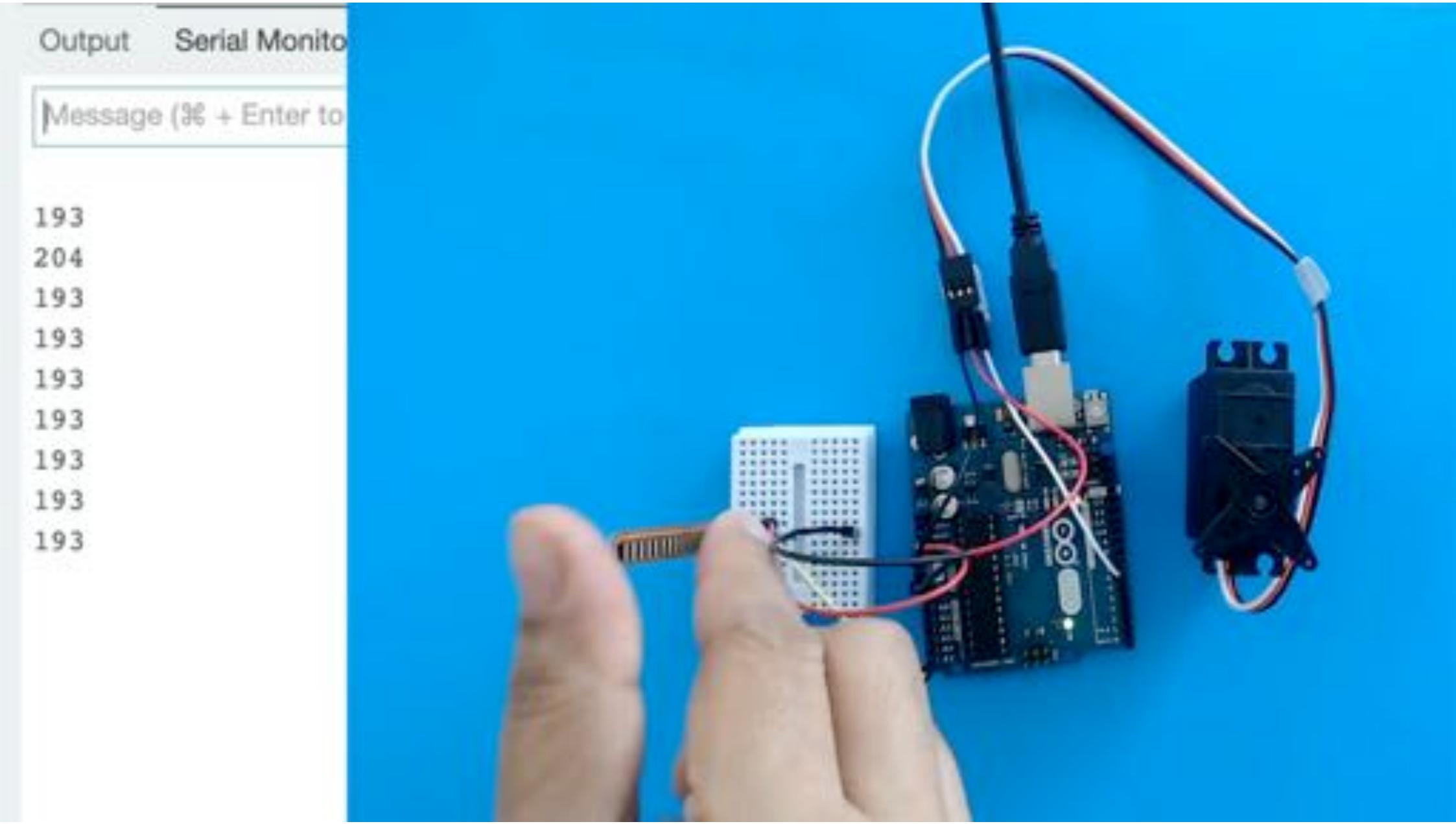
Arduino Code

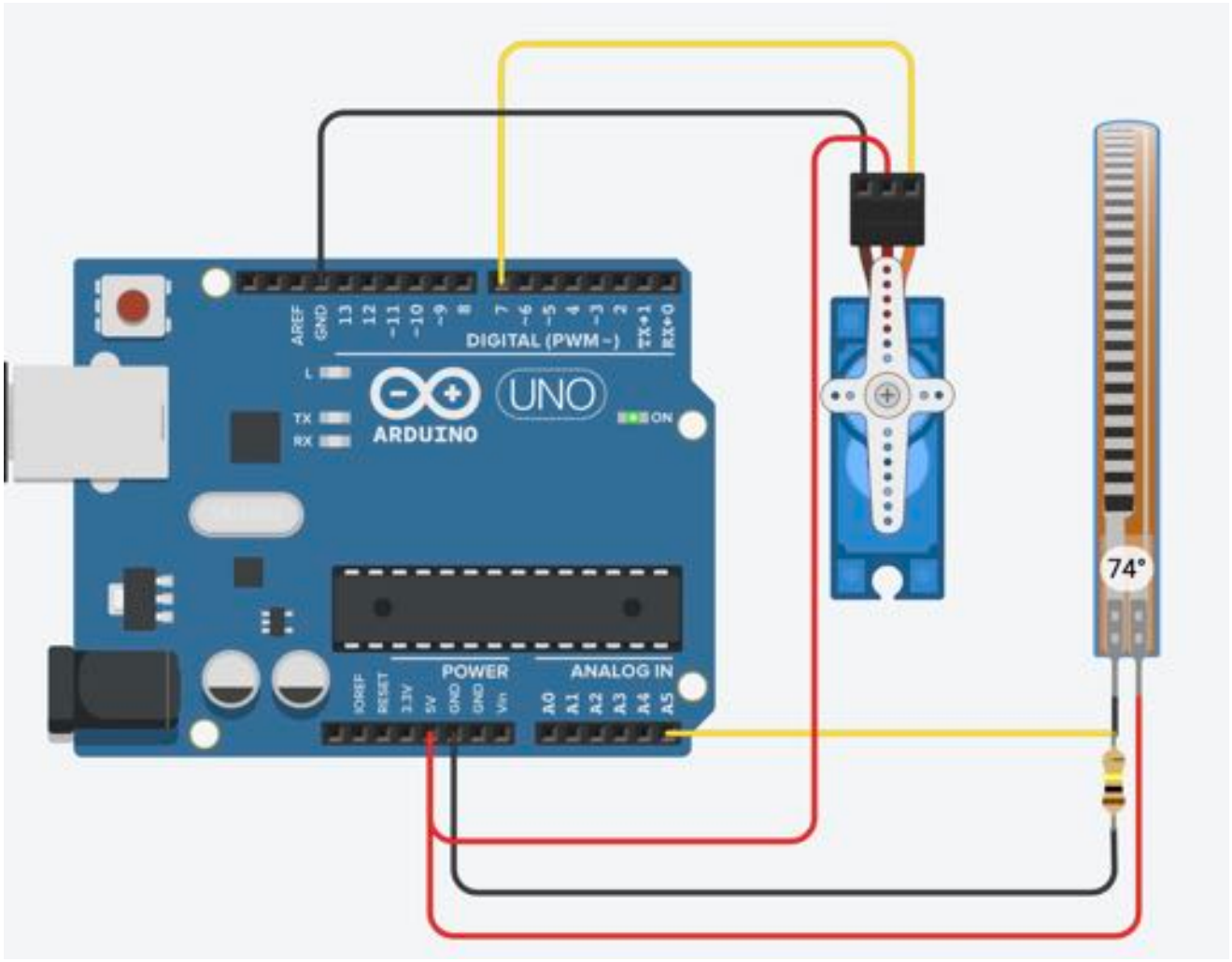
```
// C++ code
//
#include <Servo.h>

Servo servo_7;

void setup()
{
  pinMode(A5, INPUT);
  Serial.begin(9600);
  servo_7.attach(7, 500, 2500);
}
```

```
void loop()
{
  Serial.println(analogRead(A5));
  delay(500);
  if (analogRead(A5) < 588) {
    servo_7.write(180);
    delay(1000);
  } else {
    servo_7.write(0);
    delay(1000);
  }
}
```

Timeless Lifeskills FOUNDATION

Note: in the video example on the next slide, the resistor used is 10 kΩ and the value to trigger the servo has been changed to 100.


Output
Serial Monito
Message (⌘ + Enter to
193
204
193
193
193
193
193
193
193
193

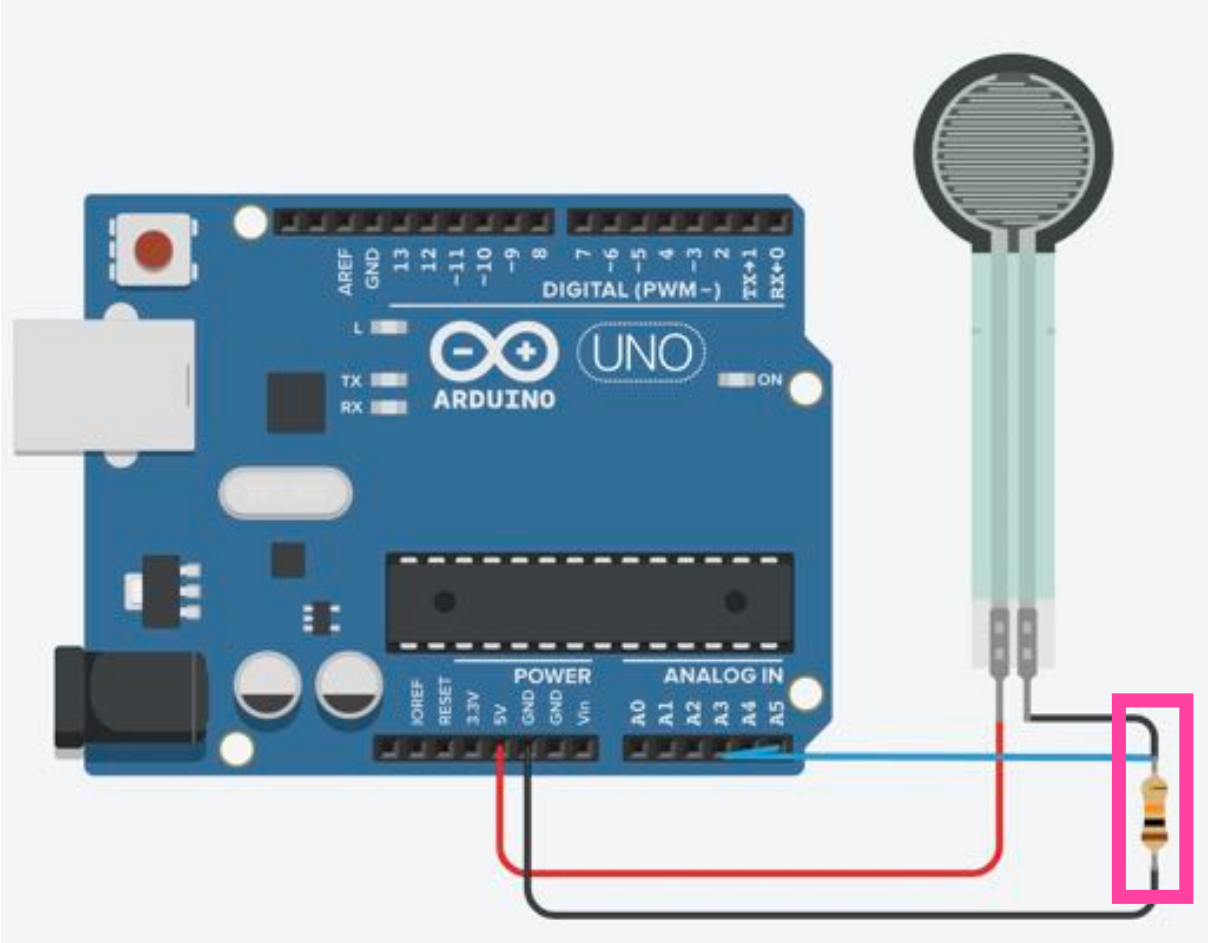


Tinkercad Project Link:

https://www.tinkercad.com/things/5fgBHGEZxbp

Force Sensitive Sensor

Timeless Lifeskills
FOUNDATION



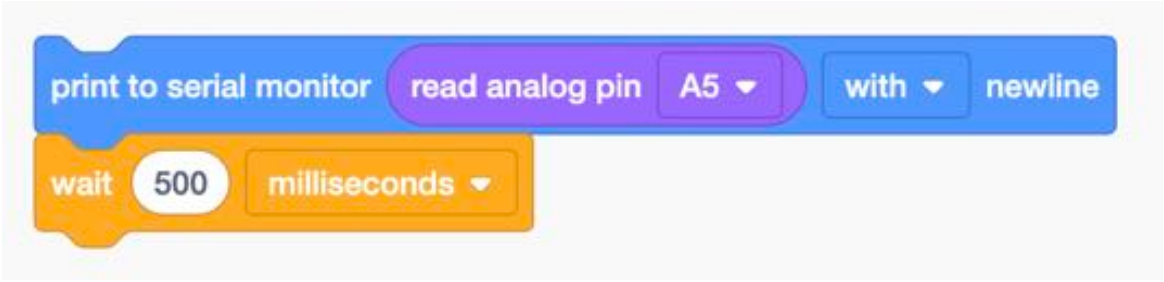
Voltage Divider

The force sensitive sensor (FSR), also called pressure sensitive sensor, is essentially a variable resistor. As more pressure is applied on the sensor its resistance decreases.

Devices like Arduino, that have an ADC, are good at detecting changes in voltage but not that good at detecting changes in resistance.

However, by adding another resistor to the FSR, in series, we can create a Voltage Divider. Then, by reading the change in the output of the voltage divider, we can write a programme that can do decision-making based on how much the pressure is being applied on the sensor.

See the Flex Sensor slides for complete description of Voltage Dividers




```
// C++ code

#include <Servo.h>

void setup()
{
  pinMode(A5, INPUT);
  Serial.begin(9600);
}
void loop()
{
  Serial.println(analogRead(A5));
  delay(500);
}
```

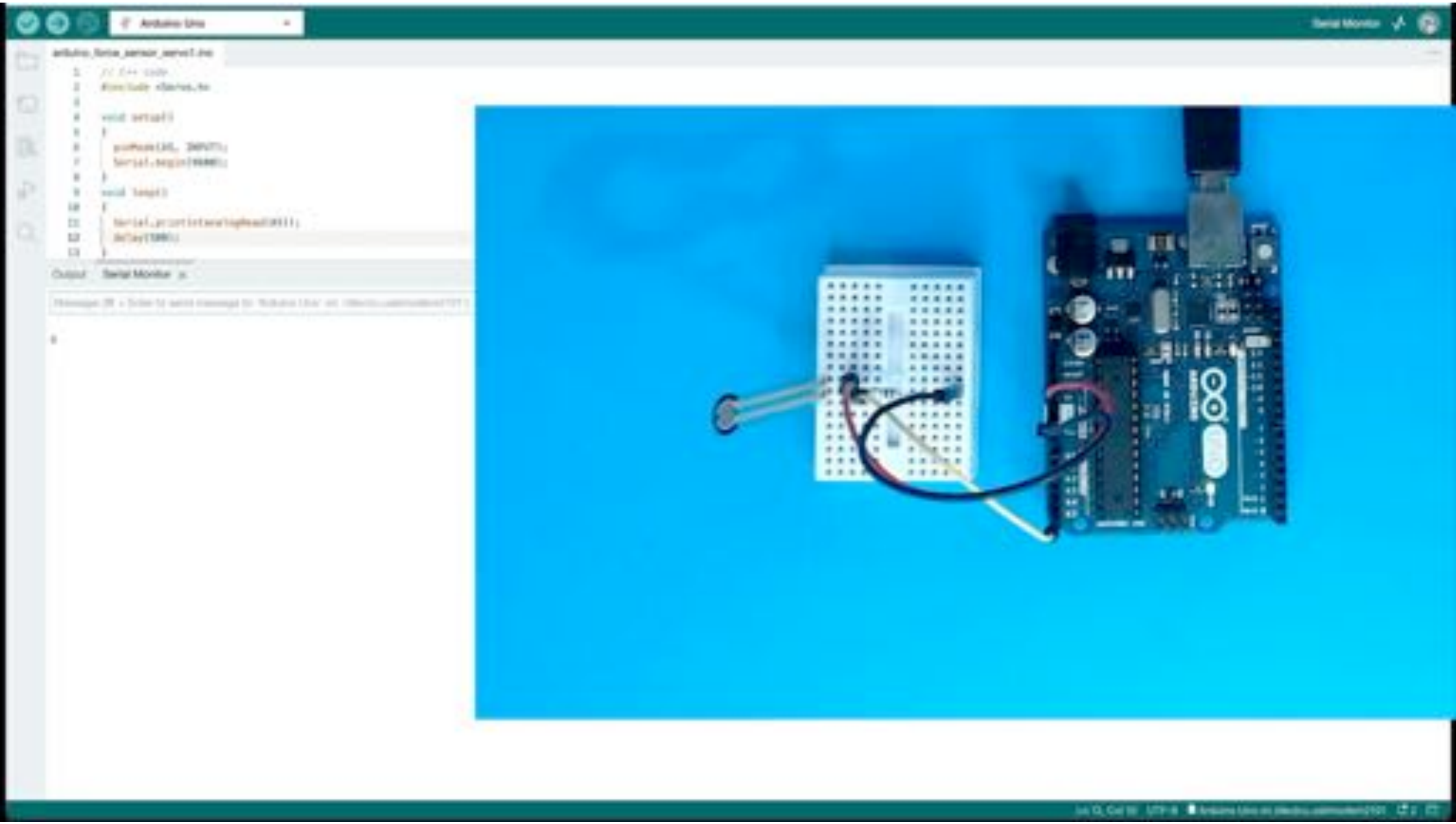
Measuring Arduino Reading as pressure applied to the FSR is changed

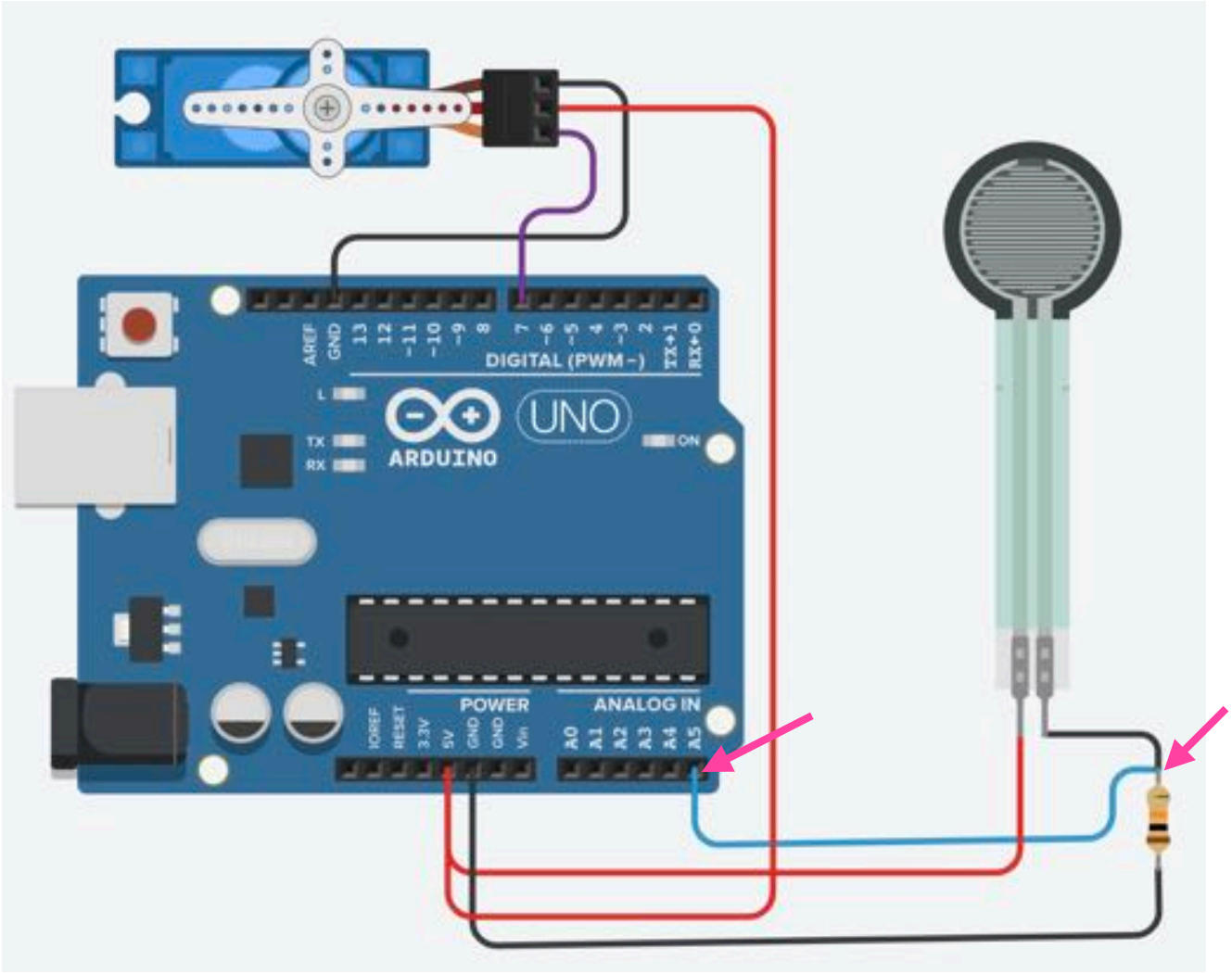
Write this simple code to find out the Analog to Digital Converter (ADC) readings as you change the pressure on the FSR.

See the video on the next slide.

In this case, a reading of around 300 means a little pressure is being applied on the FSR, a reading of around 400 implies more pressure is being applied, and a reading of more than 500 means a lot of pressure is being applied.

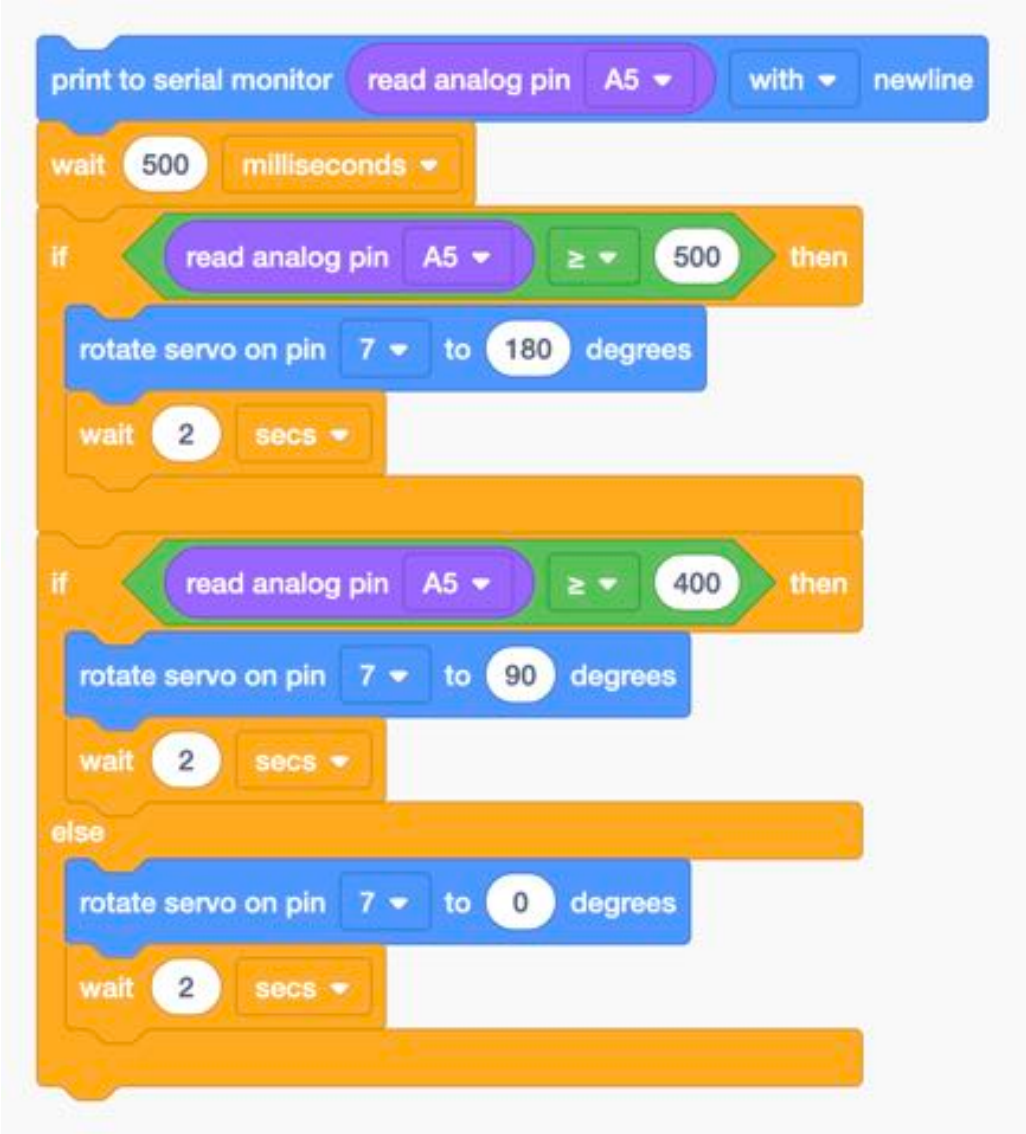
We can use these readings to write a Nested Conditional statement where as more pressure is applied the servo rotates more degrees.





Controlling a Servo with a Force Sensitive Sensor

1. Connect one terminal of the FSR to Arduino 5V pin.
2. To the other terminal of the FSR, attach a 10 kΩ resistor.
3. Connect the other end of the resistor to Arduino GND pin.
4. From middle of the FSR and resistor, connect a wire that goes to one of the Analog pins of the Arduino (A5 in this case).
5. Connect the Servo as shown.



Nested Conditional Statement to control the Servo

Since we know that based on the pressure applied to the FSR, the numbers generated by the Analog to Digital Converter (ADC) of the Arduino range from 0 (no pressure) to 500 (a lot of pressure), we can make the servo turning to varying degrees by using a nested conditional command.

Here, we are saying that if the Reading on analog pin 5 is more than 400 (i.e. moderate pressure is being applied on the FSR), turn the servo to 90°

If the reading is more than 500 (i.e. a lot of pressure is being applied on the FSR), turn the servo to 180°

If reading is less than 400, turn the servo to 0° (default position).

Arduino Code

```
// C++ code
//
#include <Servo.h>

Servo servo_7;

void setup()
{
  pinMode(A5, INPUT);
  Serial.begin(9600);
  servo_7.attach(7, 500, 2500);
}
```

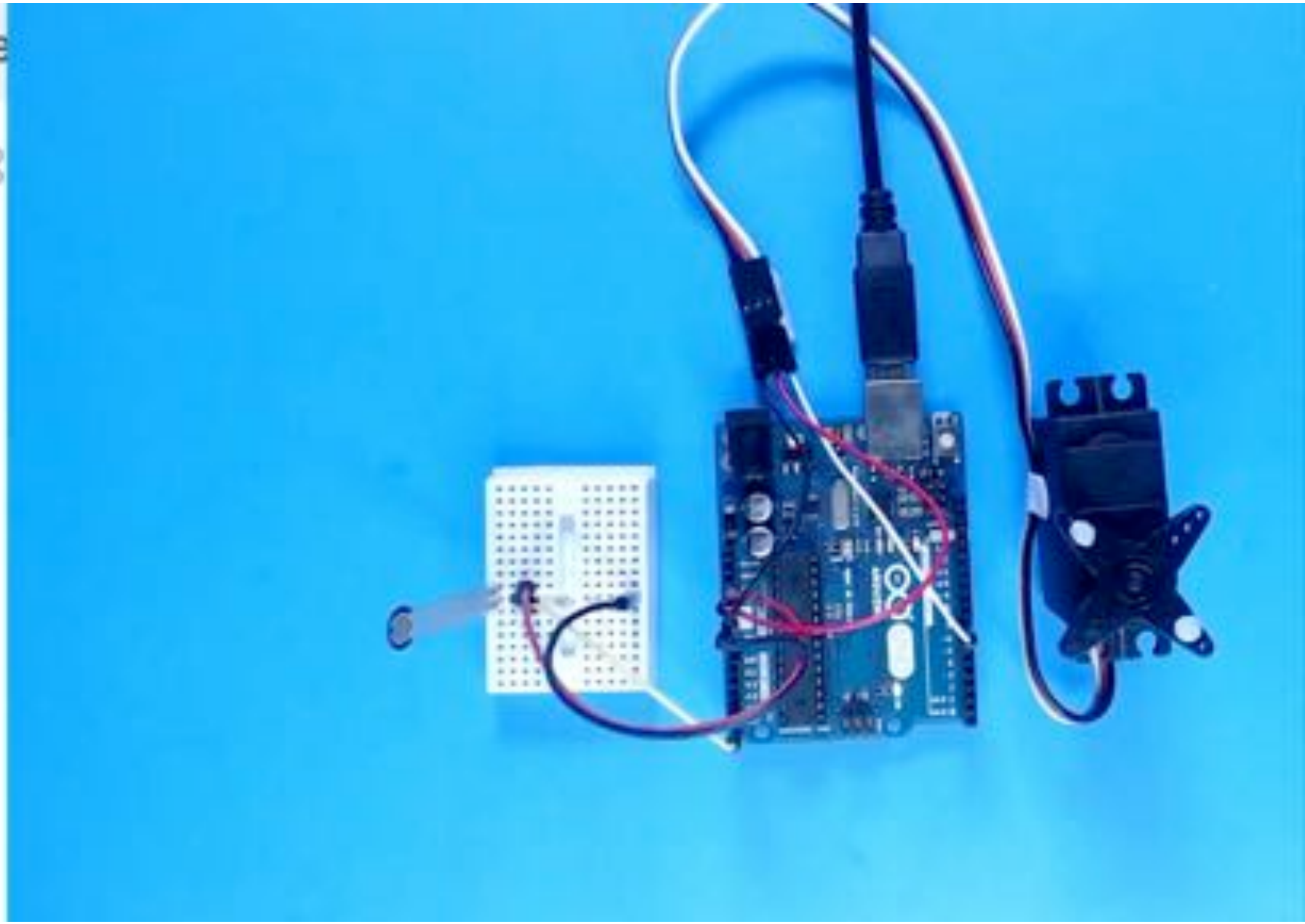
```
void loop()
{
  Serial.println(analogRead(A5));
  delay(500);
  if (analogRead(A5) >= 500) {
    servo_7.write(180);
    delay(2000);
  }
  if (analogRead(A5) >= 400) {
    servo_7.write(90);
    delay(2000);
  } else {
    servo_7.write(0);
    delay(2000);
  }
}
```

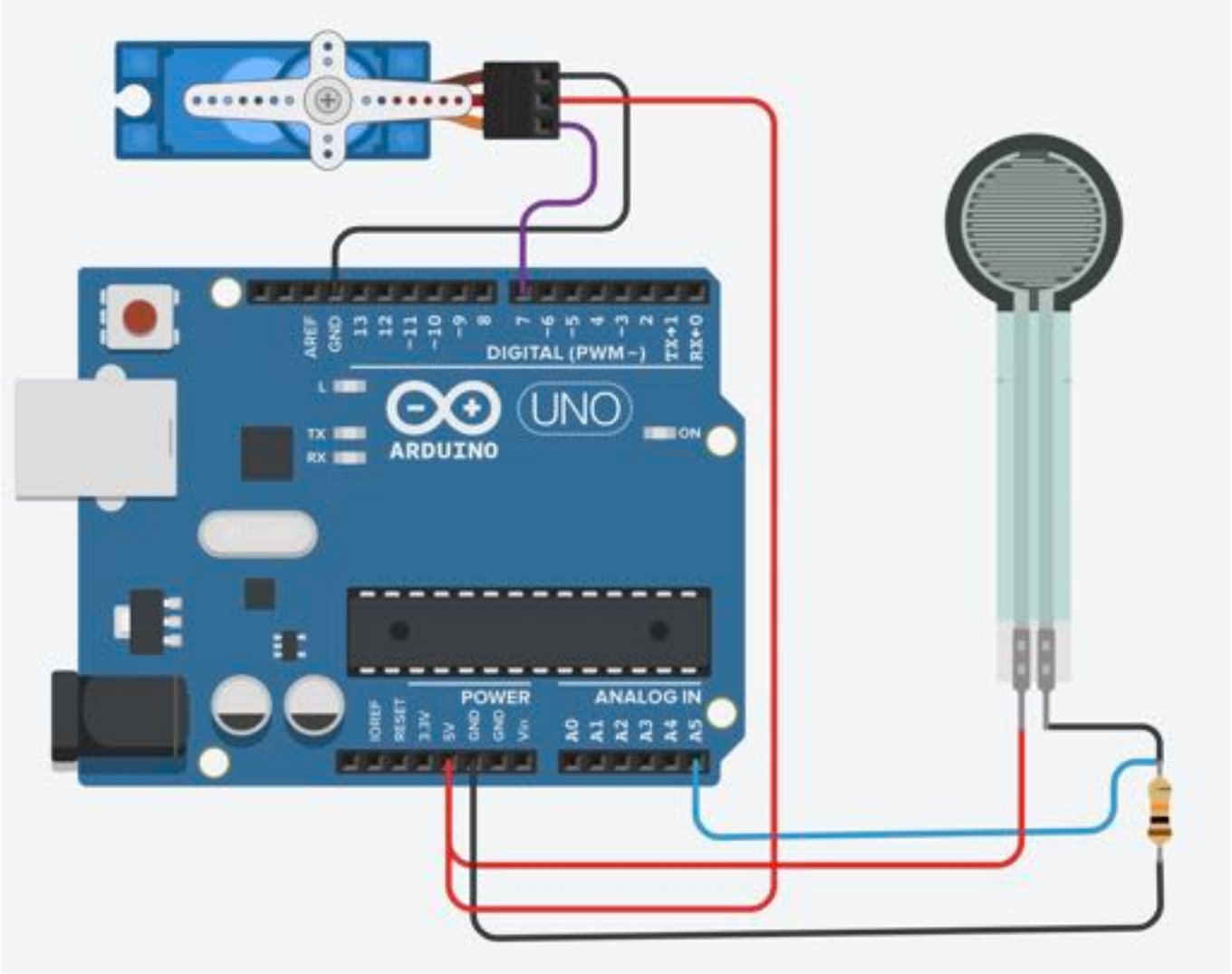
Timeless Lifeskills FOUNDATION

Output Se

Message (⌘

00



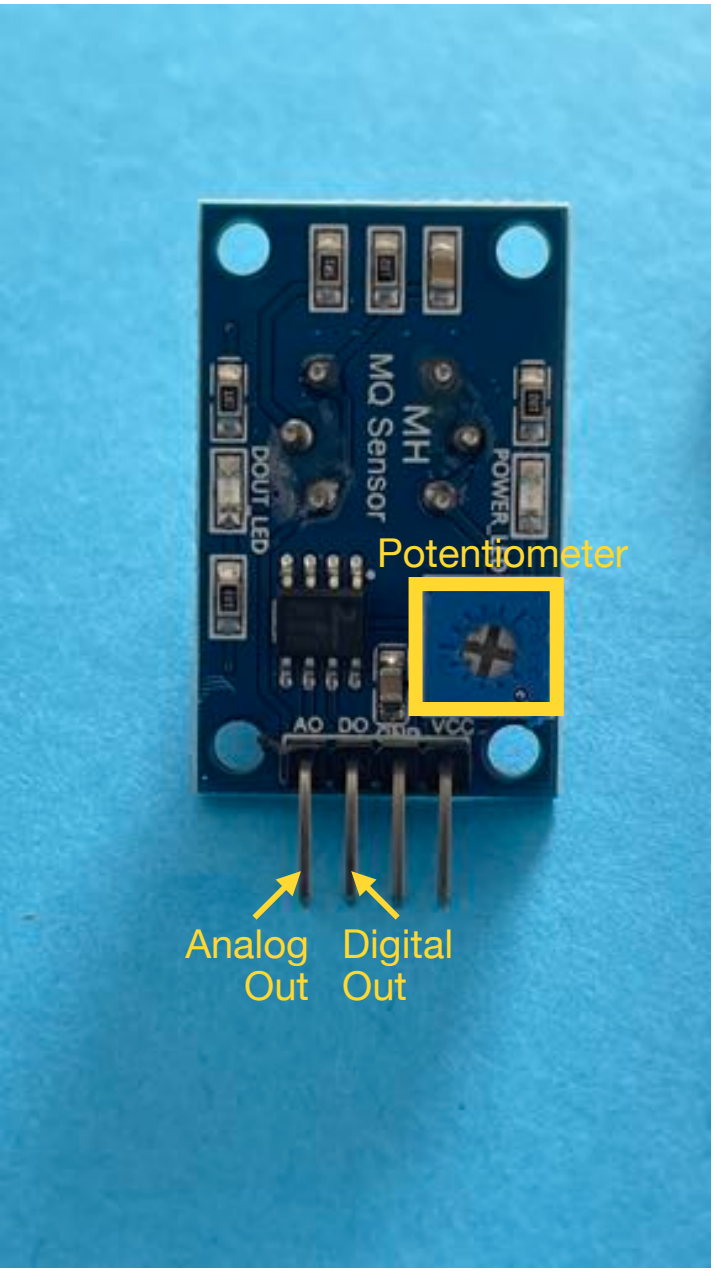


Tinkercad Project Link:

https://www.tinkercad.com/things/9MSurNn1fI9

MQ2 Smoke Sensor

Timeless Lifeskills
FOUNDATION

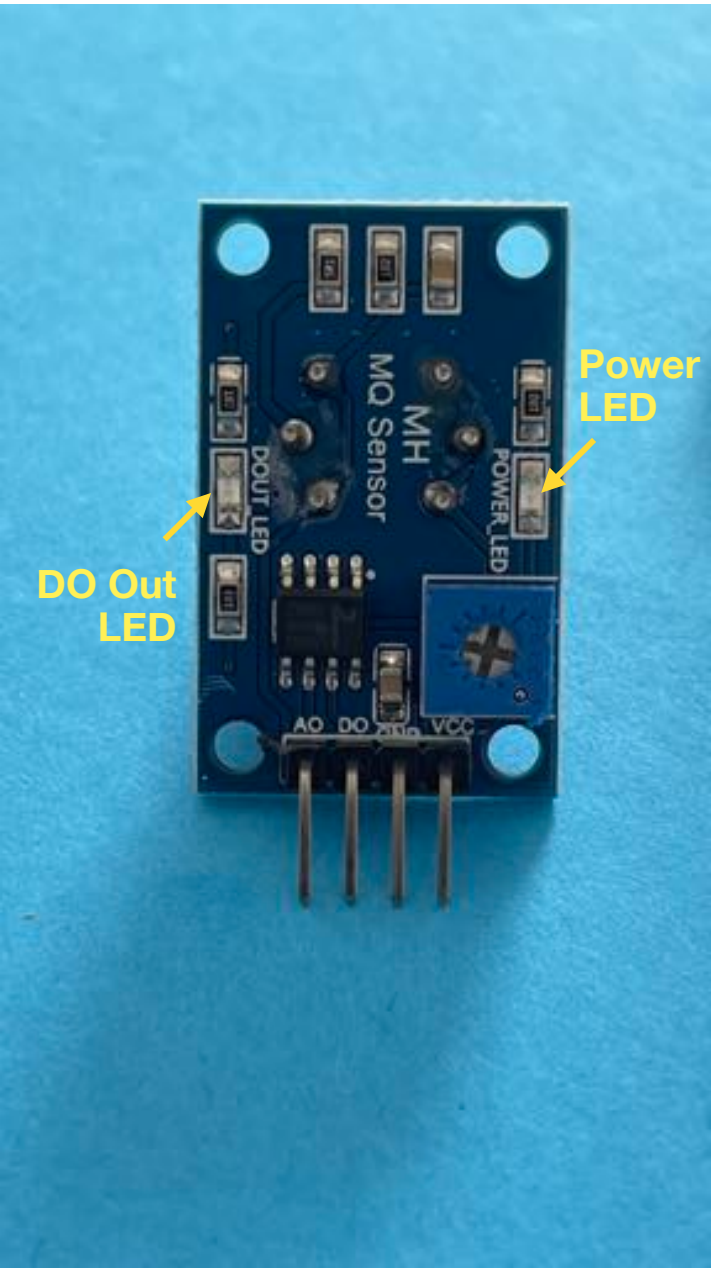


MQ2 sensor can detect smoke and inflammable gases like LPG, propane, hydrogen, methane, alcohol, and carbon monoxide concentration in the air. You can, for example, use this sensor to create a Fire Alarm.

The sensor is a metal oxide semi-conductor and its resistance changes when exposed to gases.

The sensor has an analog out (AO) and a digital out (DO). The analog output voltage varies in relation to the concentration of smoke or inflammable gases - higher the concentration, higher the output voltage and vice-versa.

This analog signal is digitised and it output on the DO pin. DO pin is low when gas concentration is above the threshold value and high otherwise. If you use the DO pin, you can adjust the sensitivity of the sensor by using the onboard potentiometer.

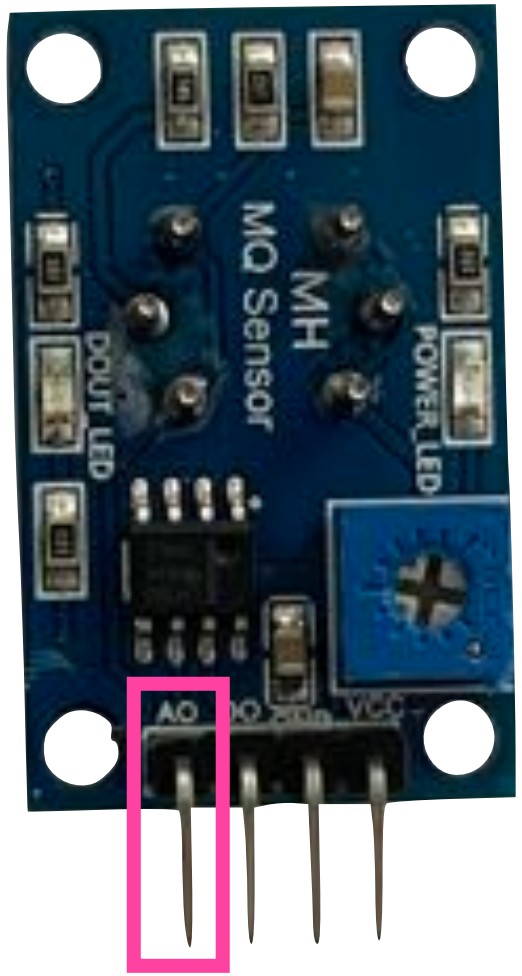


The Power LED lights up when the sensor is turned on and the DO Out or status LED lights up when the gas concentration exceeds the threshold value (while using the digital pin).

To get accurate reading from the sensor, you have to preheat the sensor for 24 hours (since it is a heat-based sensor).

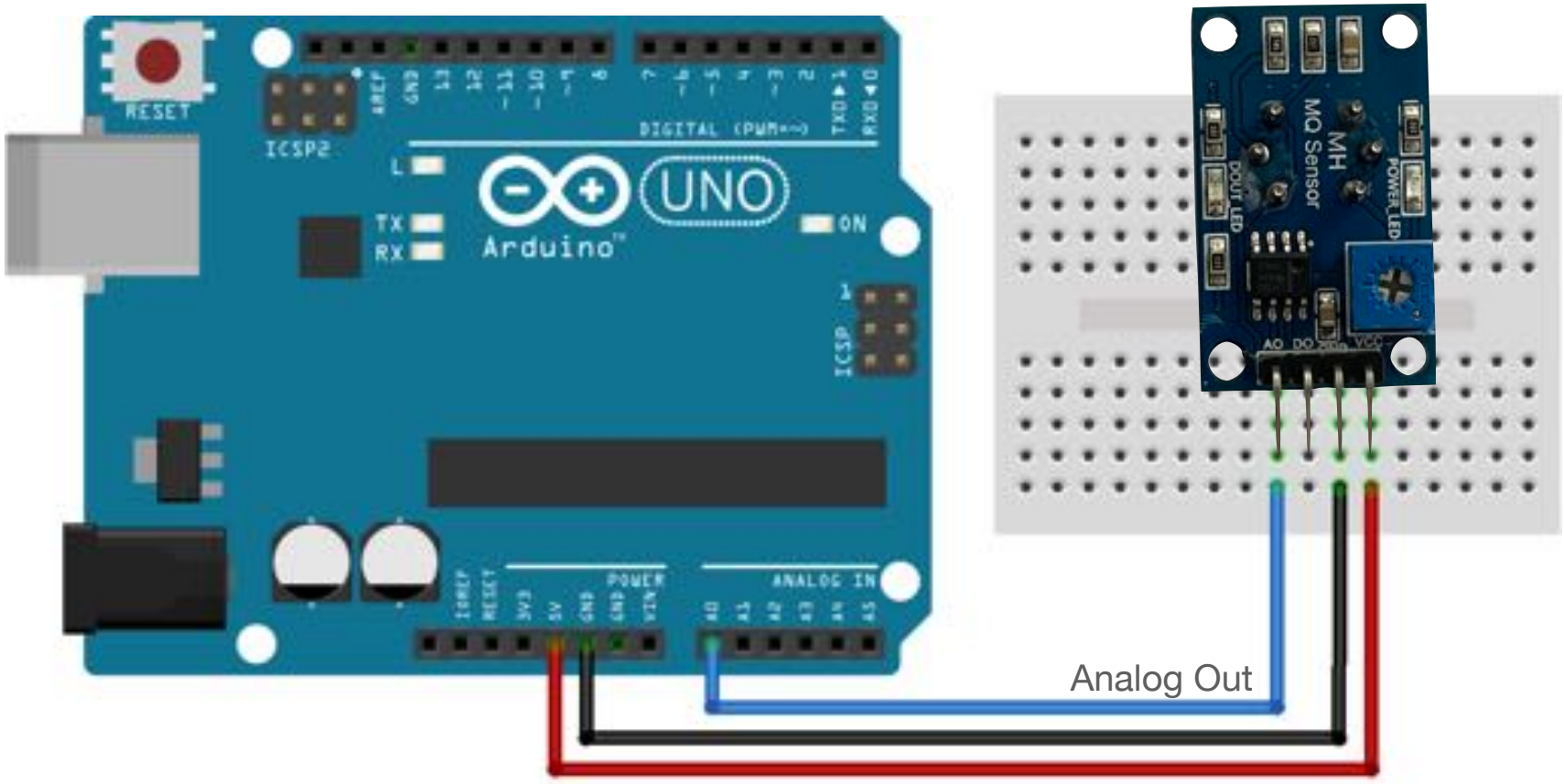
However, if you are just testing the sensor to learn how it works, you can use it without pre-heating (in this case, reading value will not be accurate but the logic will work).

If want to make a real smoke or gas alarm, you must pre-heat the sensor and calibrate it.

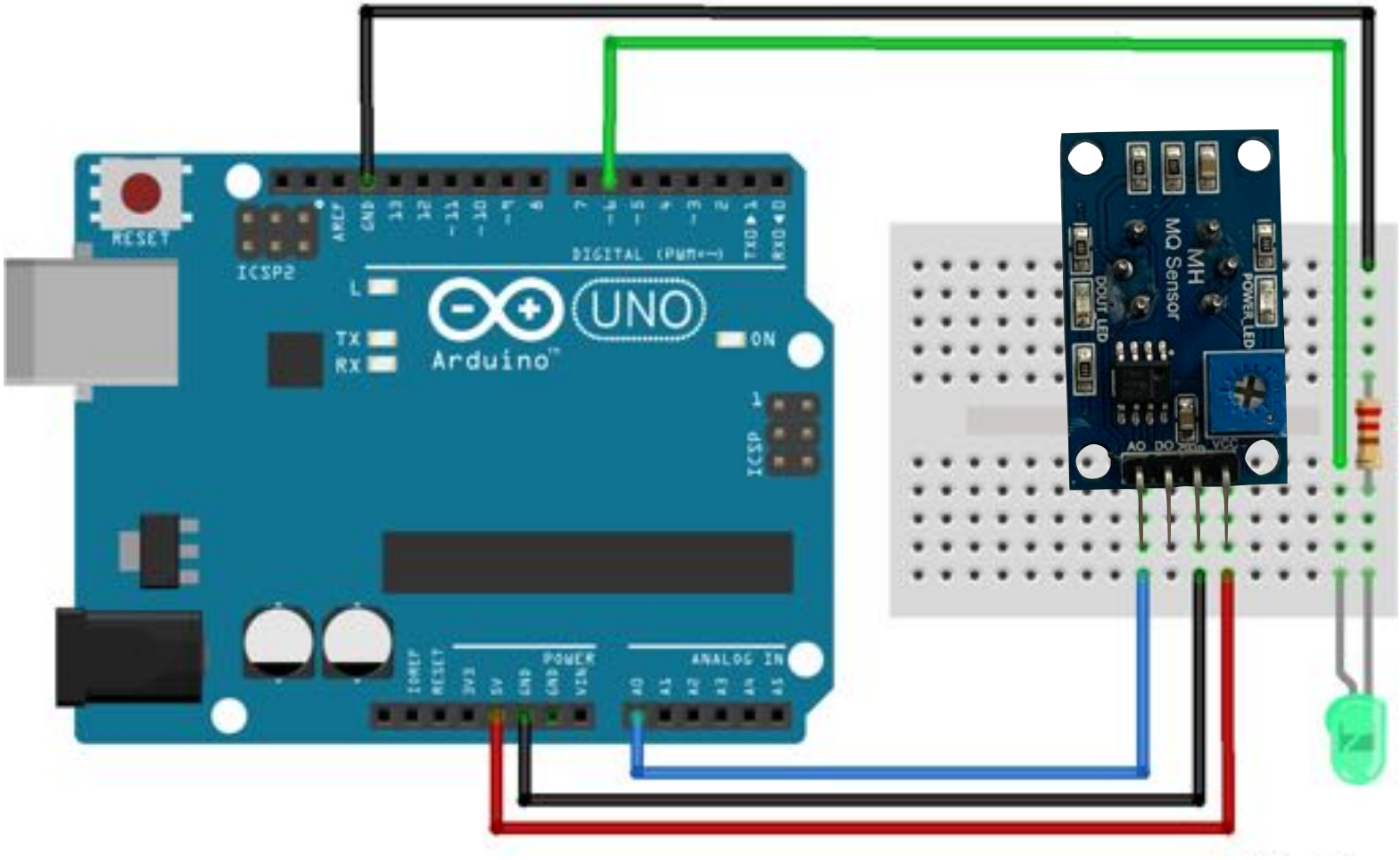


MQ2
Smoke Sensor
Analog Reading

Analog Pin (AO) - Circuit Diagram



MQ Sensor with LED

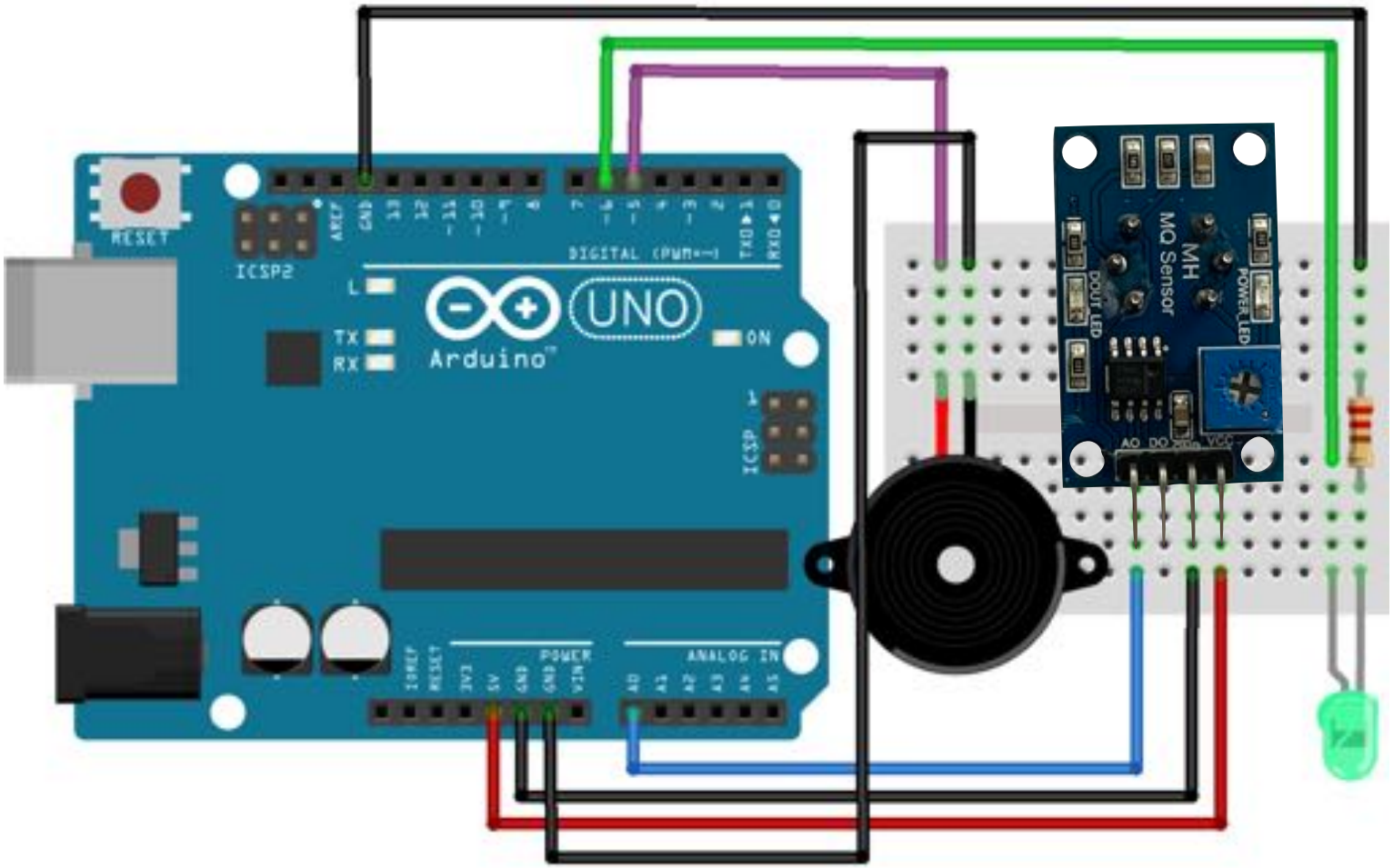


Add a LED

- Connect LED anode (+) to pin 6
- Connect LED cathode (-) to a 200 Ω resistor and to Arduino ground
- When no smoke is detected, green LED will light up
- When smoke is detected, LED will be off and alarm will sound

Timeless Lifeskills FOUNDATION

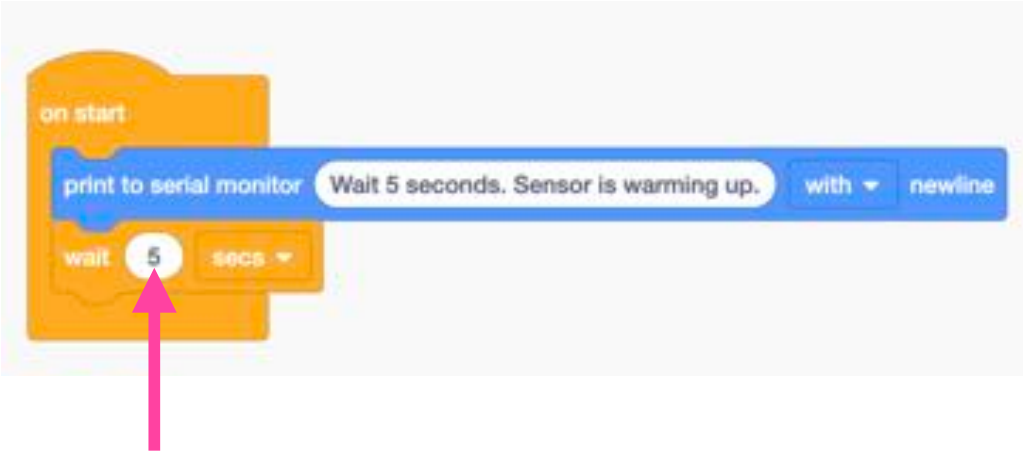
MQ Sensor with LED and Buzzer



Add a Buzzer

- Connect buzzer anode (+) to pin 5
- Connect buzzer cathode (-) to Arduino ground
- When no smoke is detected, buzzer will be off (LED will light up)
- When smoke is detected, buzzer will be on (LED will be off)

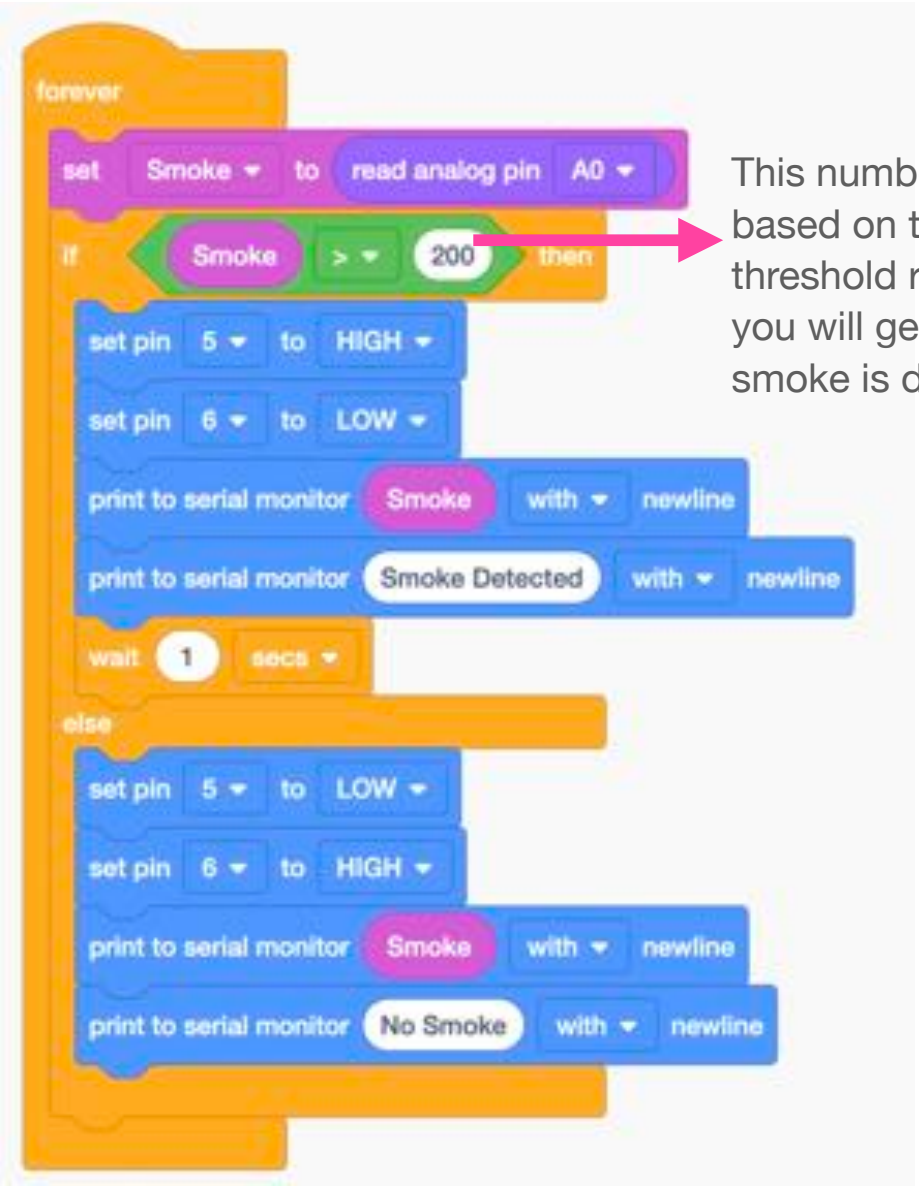
Timeless Lifeskills FOUNDATION



To get accurate reading from the sensor, you should preheat the sensor for 24 hours.

Here, we are just testing the sensor to learn how it works, without properly heating and calibrating it.

Even so, we have to give some time to the sensor to stabilise, hence we have added a 5 second wait.



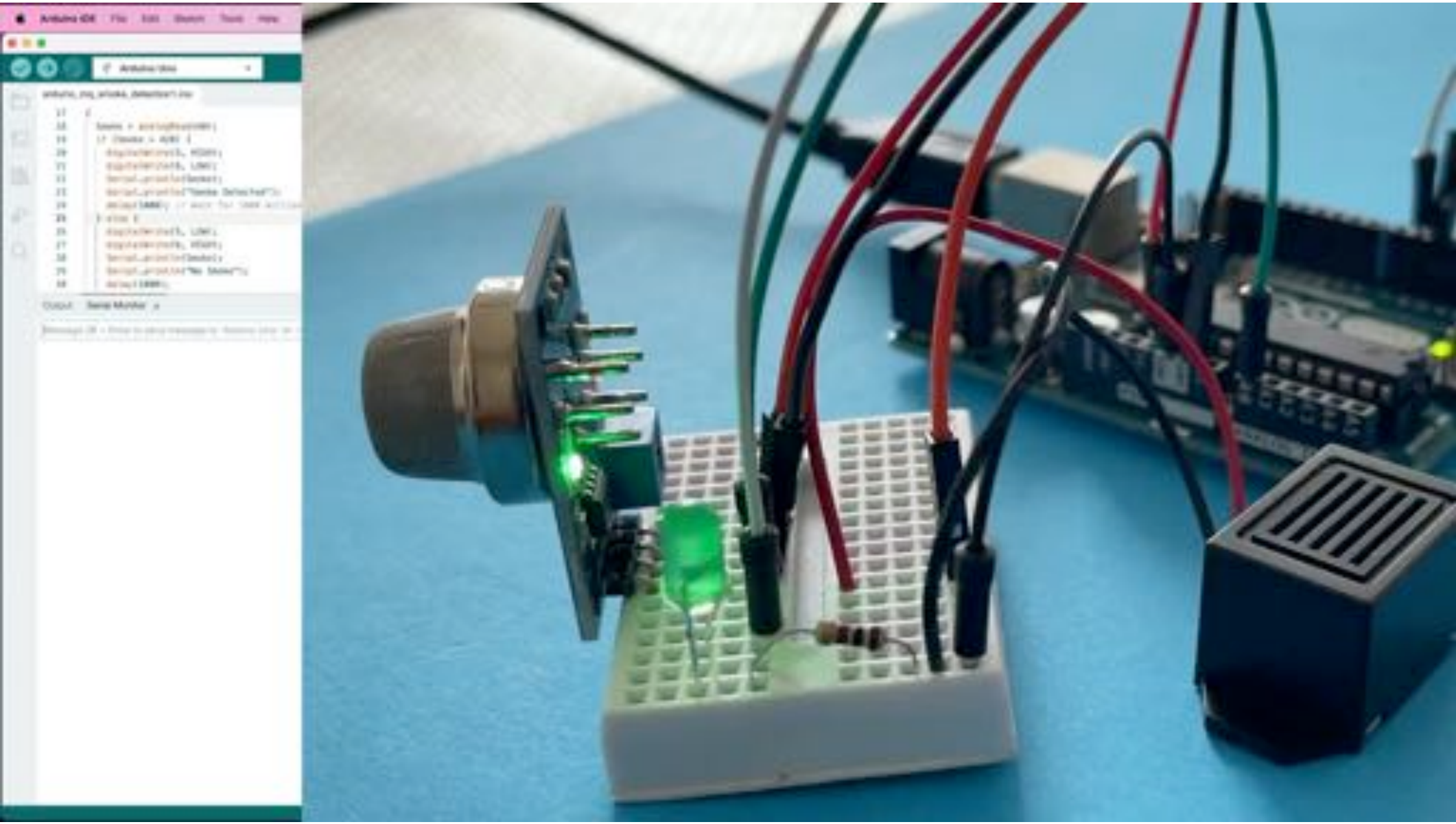
This number is based on the threshold reading you will get when smoke is detected

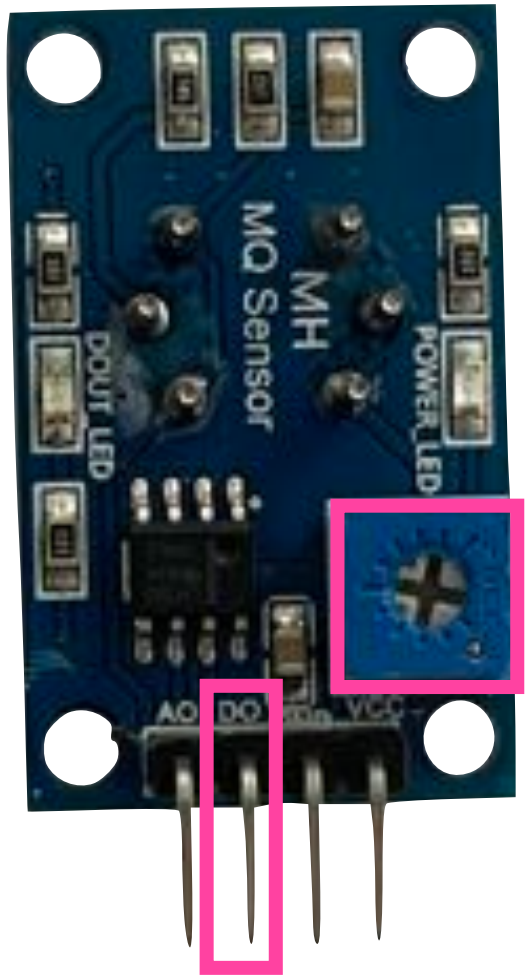
Arduino Code

```
// C++ code
//
int Smoke = 0;
void setup()
{
  Serial.begin(9600);
  pinMode(A0, INPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  Serial.println("Wait 5
seconds. Sensor is warming
up.");
  delay(5000);
}
```

```
void loop()
{
  Smoke = analogRead(A0);
  if (Smoke > 420) {
    digitalWrite(5, HIGH);
    digitalWrite(6, LOW);
    Serial.println(Smoke);
    Serial.println("Smoke Detected");
    delay(1000);
 } else {
    digitalWrite(5, LOW);
    digitalWrite(6, HIGH);
    Serial.println(Smoke);
    Serial.println("No Smoke");
    delay(1000);
  }
}
```

Timeless Lifeskills FOUNDATION

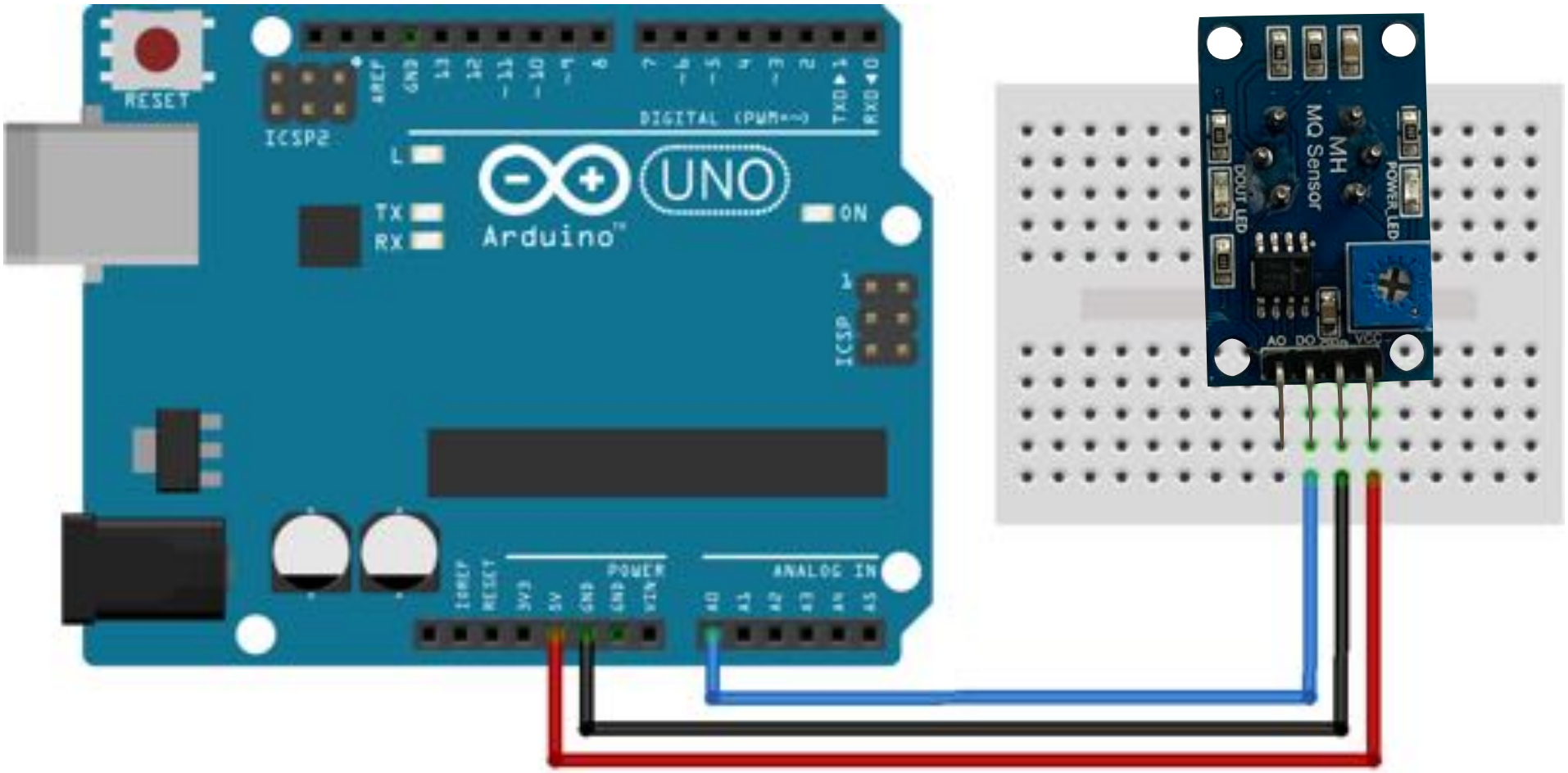




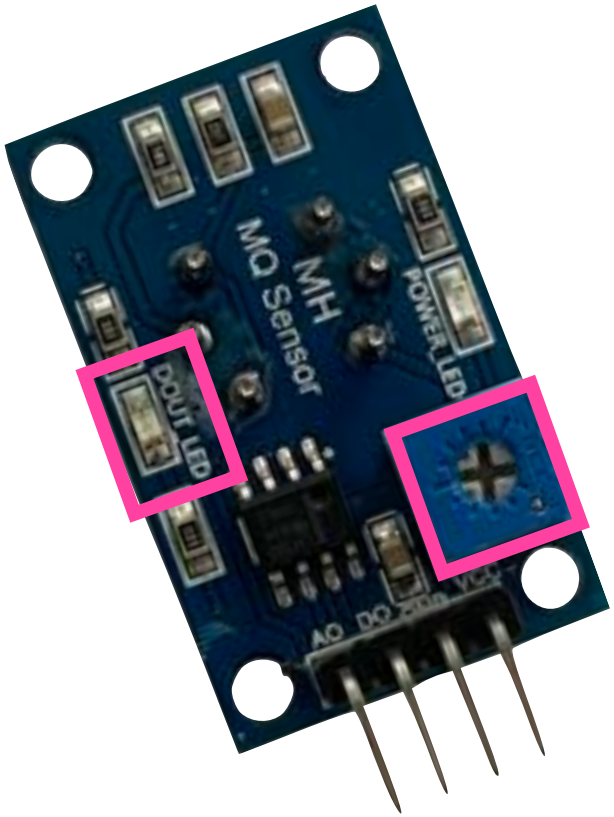
MQ2
Smoke Sensor
Digital Reading

Timeless Lifeskills
FOUNDATION

Digital Pin (DO) - Circuit Diagram

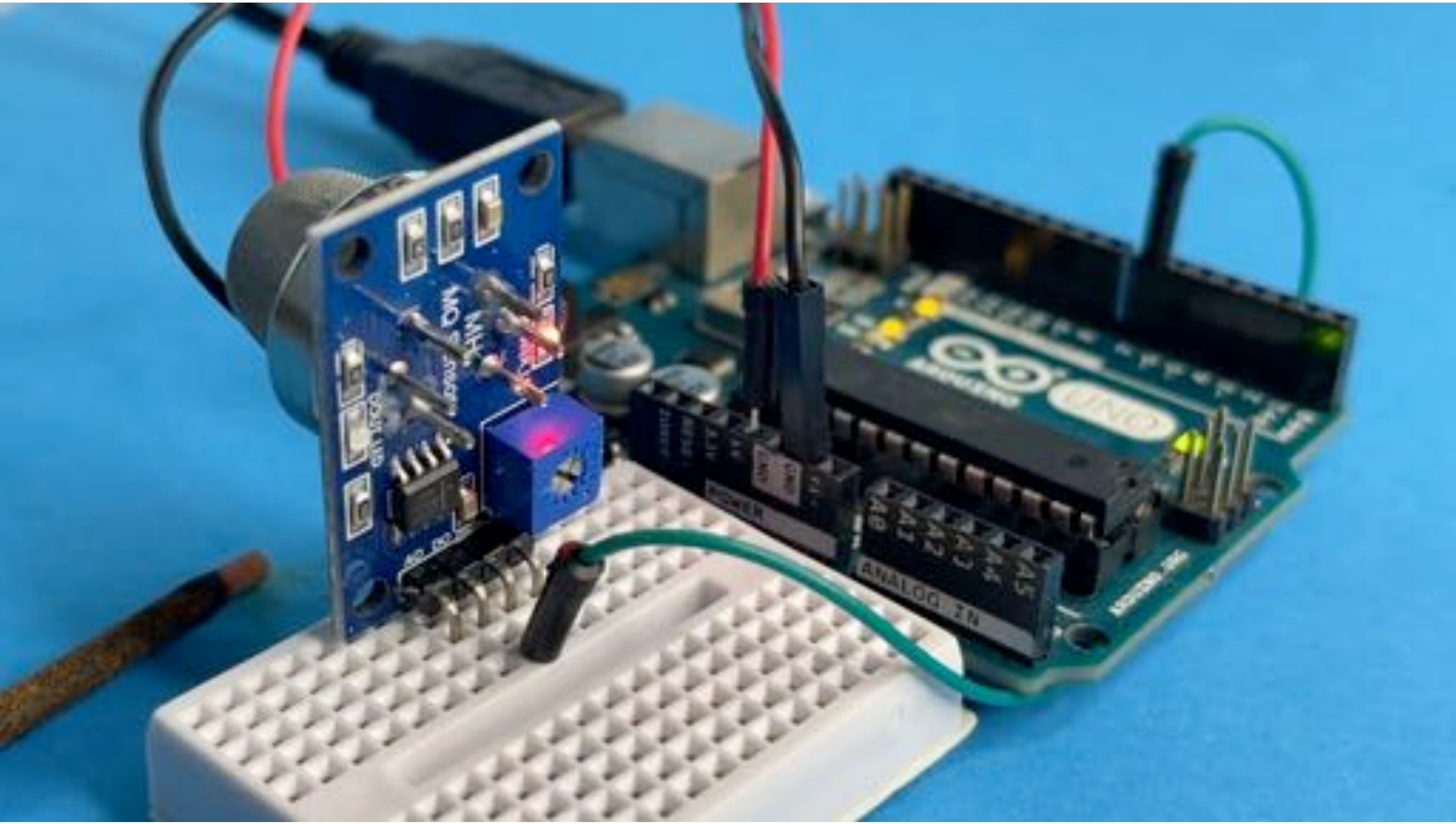


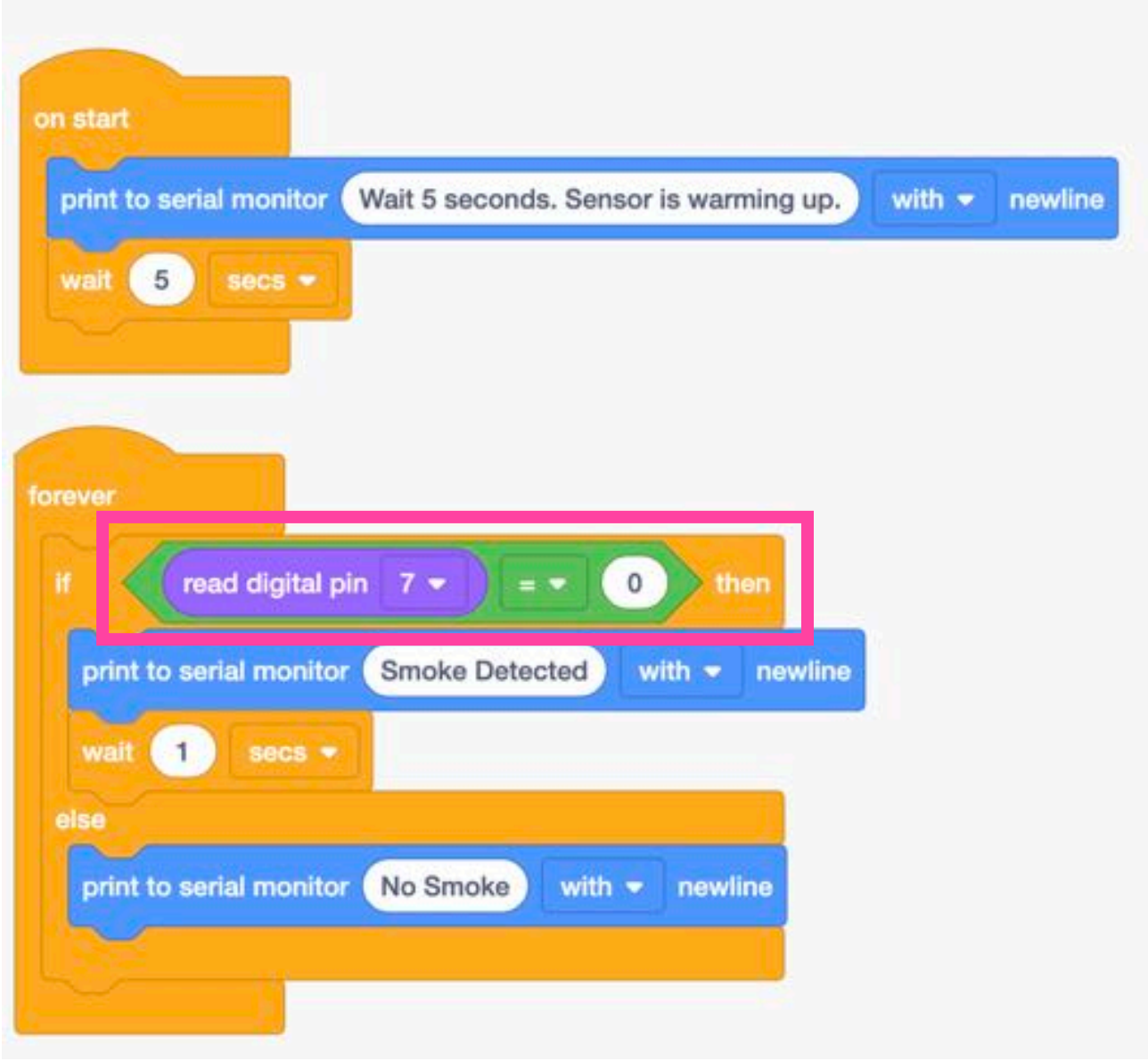
How to calibrate the MQ2 sensor for Digital Reading



1. Turn the potentiometer clockwise a few turns
2. Put the sensor in a smoky environment (like light an incense stick to generate smoke)
3. Turn the potentiometer anticlockwise till the DO OUT LED (status LED) lights up
4. Then turn the potentiometer clockwise just a little till the status LED is off
5. Now the sensor is calibrated to detect smoke
6. If you use the sensor after a long break, you will need to calibrate the sensor again
7. Video in the next slide demonstrates this process

Timeless Lifeskills
FOUNDATION


UNO
ANALOG IN



1. We are reading digital pin 7, to which the DO pin of the MQ2 sensor is connected
2. DO pin stays HIGH when there is no smoke and becomes LOW when smoke is detected
3. Hence, in our code the conditional statement is “If Read Digital Pin 7 = 0” (i.e. DO pin is LOW), message displayed is “Smoke Detected”
4. Else (which would imply DO Pin 7 is HIGH), message displayed is “No Smoke”

Arduino Code

```
// C++ code

void setup()
{
  Serial.begin(9600);
  pinMode(7, INPUT);

  Serial.println("Wait 5
seconds. Sensor is warming
up.");
  delay(5000);
}
```

```
void loop()
{
  if (digitalRead(7) == 0) {
    Serial.println("Smoke Detected");
    delay(1000);
  } else {
    Serial.println("No Smoke");
    delay(1000);
  }
}
```

Timeless Lifeskills FOUNDATION