

STREAMING

LIVE

ESL

Embedded Training

Basic Hardware



Embedded System Laboratory

Lesson Outline

09.10 - 10.00 am

Hour 1: Introduction to Arduino and Tinkercad

- Basic C Programming in Arduino
- ข้อมูลเบื้องต้นเกี่ยวกับบอร์ด Arduino และส่วนประกอบต่างๆ (ไมโครคอนโทรลเลอร์ พิน แหล่งจ่ายไฟ ฯลฯ)
- ข้อมูลเบื้องต้นเกี่ยวกับ Tinkercad และวิธีใช้เพื่อจำลองวงจร
- สาธิตการใช้งานวงจร LED เบื้องต้นและการเขียนโปรแกรมใน Arduino
- กิจกรรมภาคปฏิบัติ: สร้างวงจร LED พื้นฐานและตั้งโปรแกรมให้กระพริบในอัตราที่แตกต่างกัน

Lesson Outline

10.10 - 11.00 am

Hour 2: Sensors

- ความรู้เบื้องต้นเกี่ยวกับเซ็นเซอร์และบทบาทในวงจร
- สาริตวิจารณาเซ็นเซอร์อุณหภูมิและการอ่านค่าใน Arduino
- กิจกรรมภาคปฏิบัติ: สร้างวงจรเซ็นเซอร์อุณหภูมิและตั้งโปรแกรมให้แสดงอุณหภูมิบนจอภาพ Serial Monitor

Lesson Outline

11.10 - 12.00 am

Hour 3: Advanced Circuit

- สาริตวงจรเซอร์วัตแสงและวิธีใช้ตัวแปรควบคุมความสว่างของหลอด LED
- กิจกรรมภาคปฏิบัติ : สร้างวงจรเซ็นเซอร์แสงและตั้งโปรแกรมให้ปรับความสว่าง LED ตามระดับแสงโดยรอบ

ถ้าเวลาเหลือ จะมีเนื้อหาแถมให้

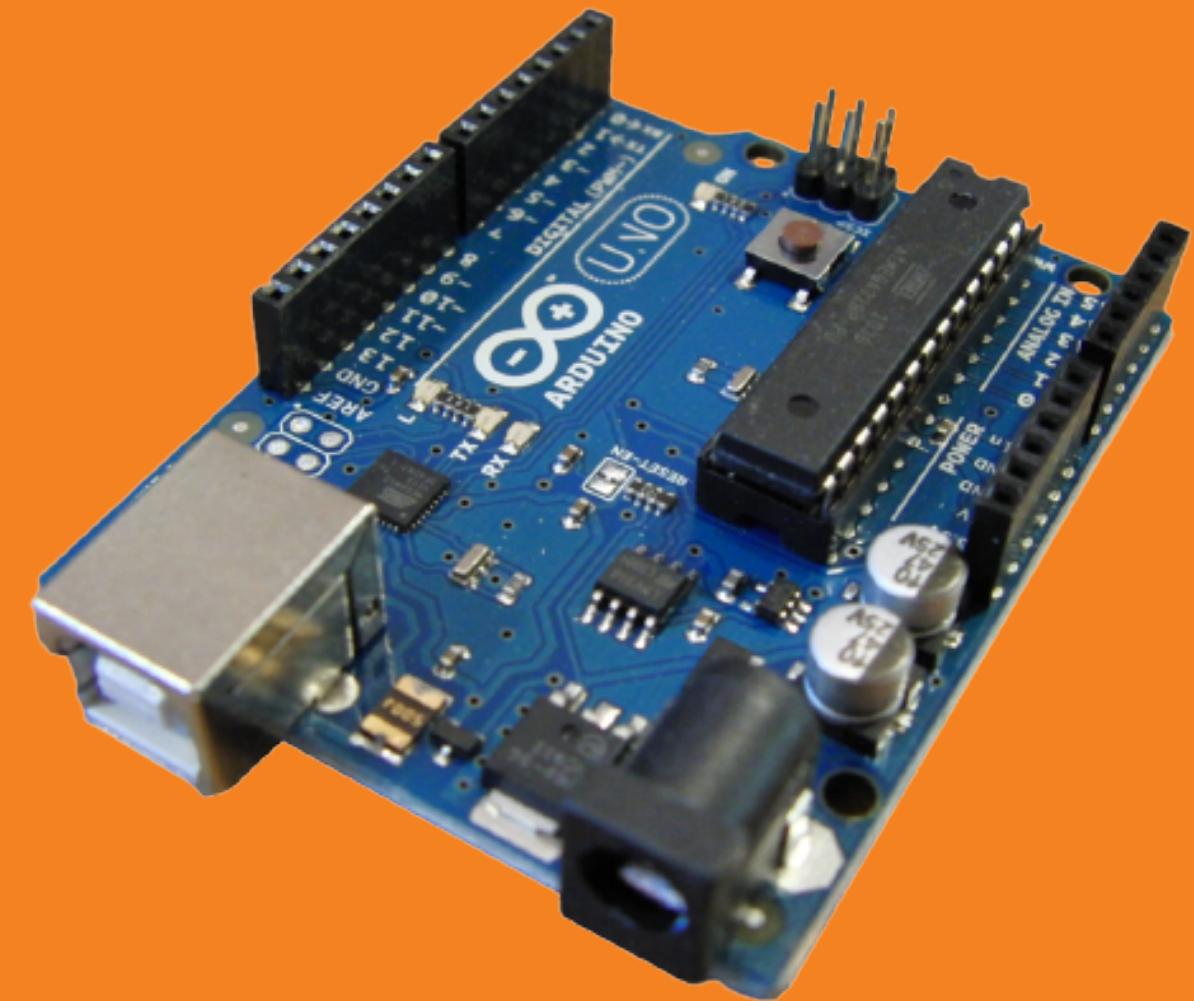
Break

12.00 - 13.00 pm

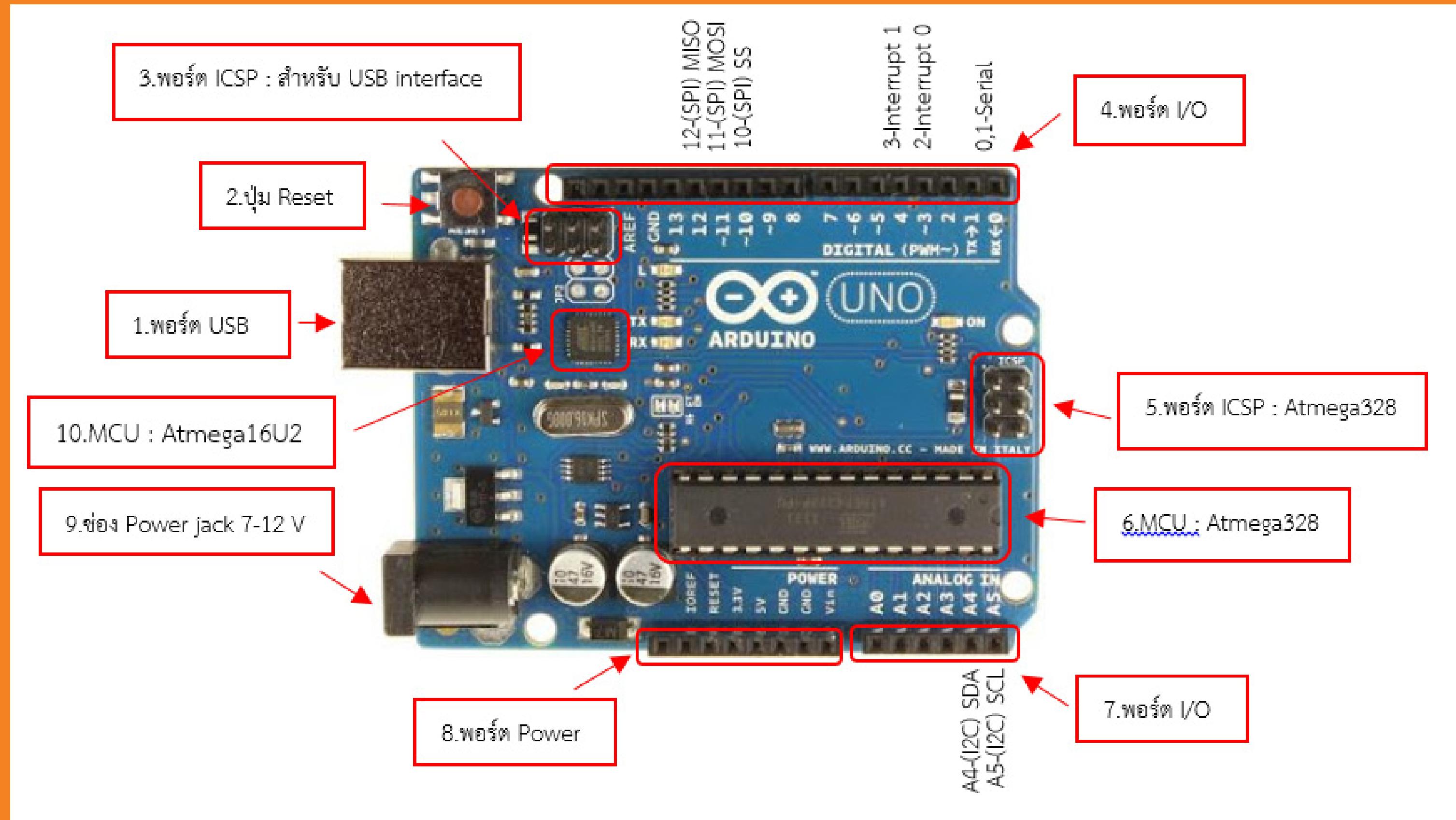
Introduction to Arduino and Tinkercad

บอร์ด Arduino

- อ่านว่า (อา-ดู-อ-โน่ หรือ อาดูยโน่)
- เป็นบอร์ด ไมโครคอนโทรลเลอร์
- มีการใช้งานที่ง่าย
- มี Library มาก
- มีอุปกรณ์ที่นำไปเชื่อมต่อมาก
- ทำให้ได้รับความนิยมอย่างมาก



องค์ประกอบของ Arduino



Spec



Arduino UNO Rev 3 Specifications

MCU: Atmega328P

Digital I/O pins: 14(6 pins are PWM)

Clock Speed: 16 MHz

Memory: 32KB of Flash, 2KB SRAM, 1KB EEPROM

Power Source: DC power jack and USB port

Analog I/O pins: 6(can be used as Digital I/O)

Operating Voltage: 5 Volts

Input Voltage limit: 7-20 Volts

Dimensions: 68.6 mm X 53.4 mm

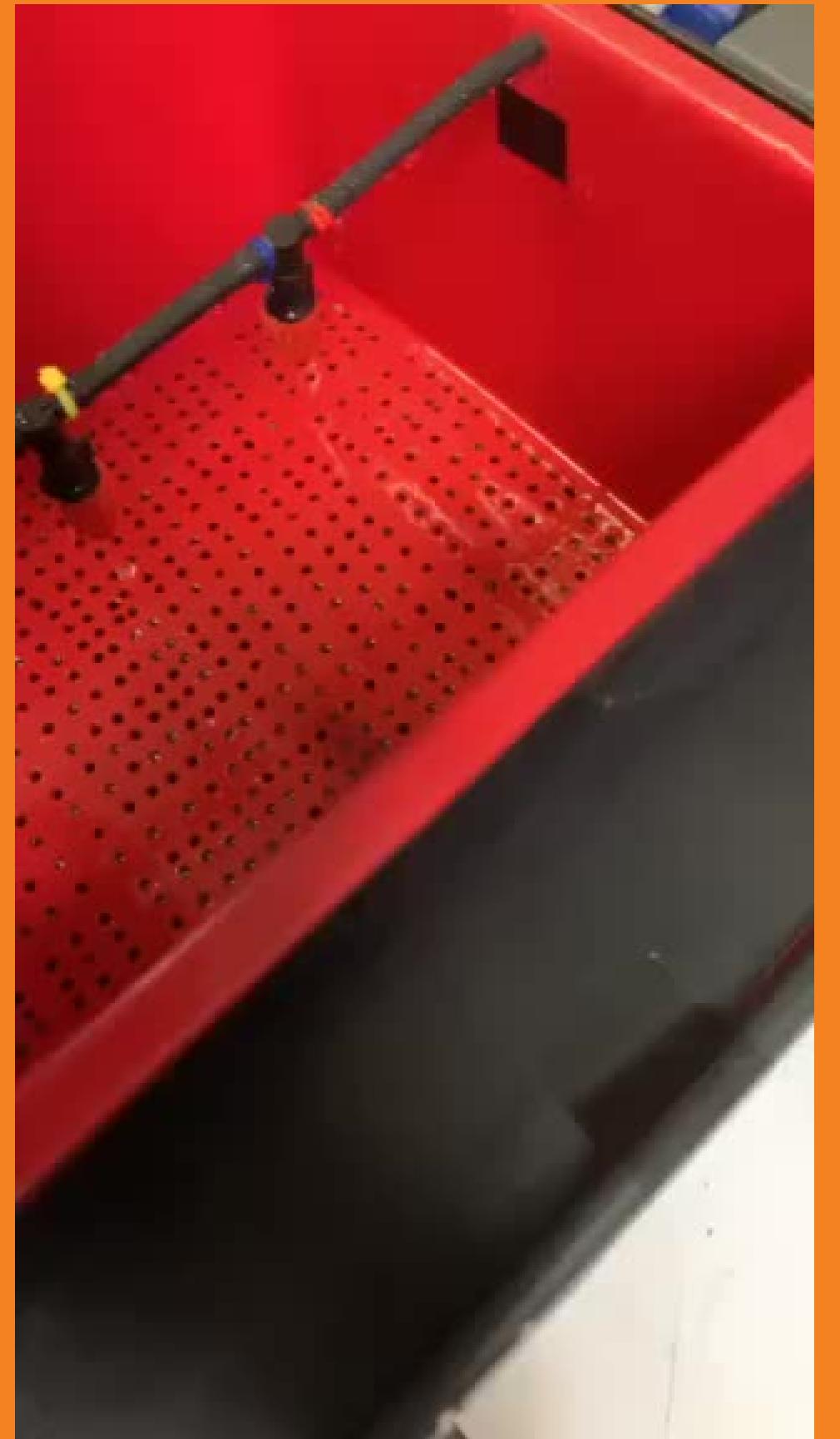
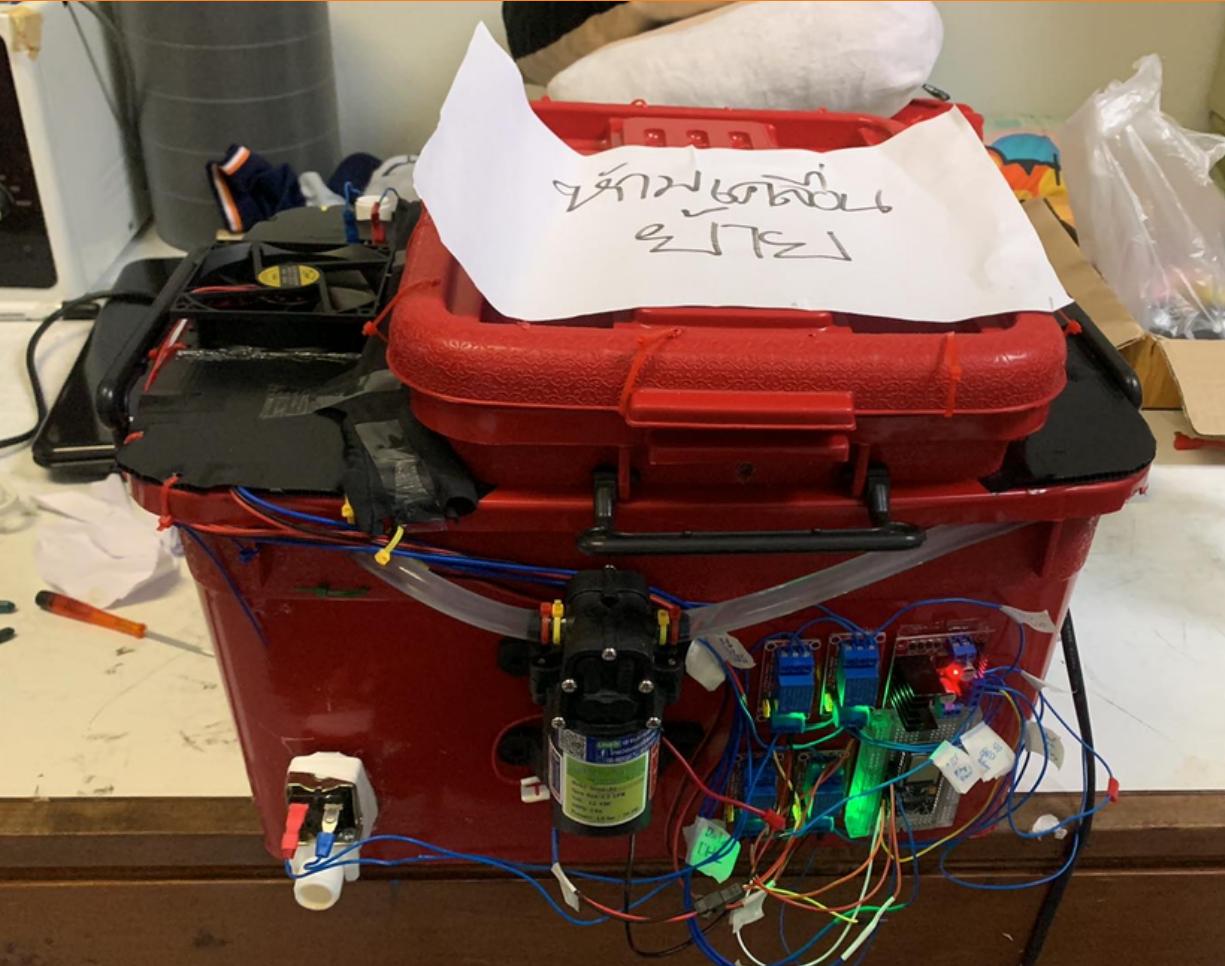
www.eTechnophiles.com

Example Arduino Project

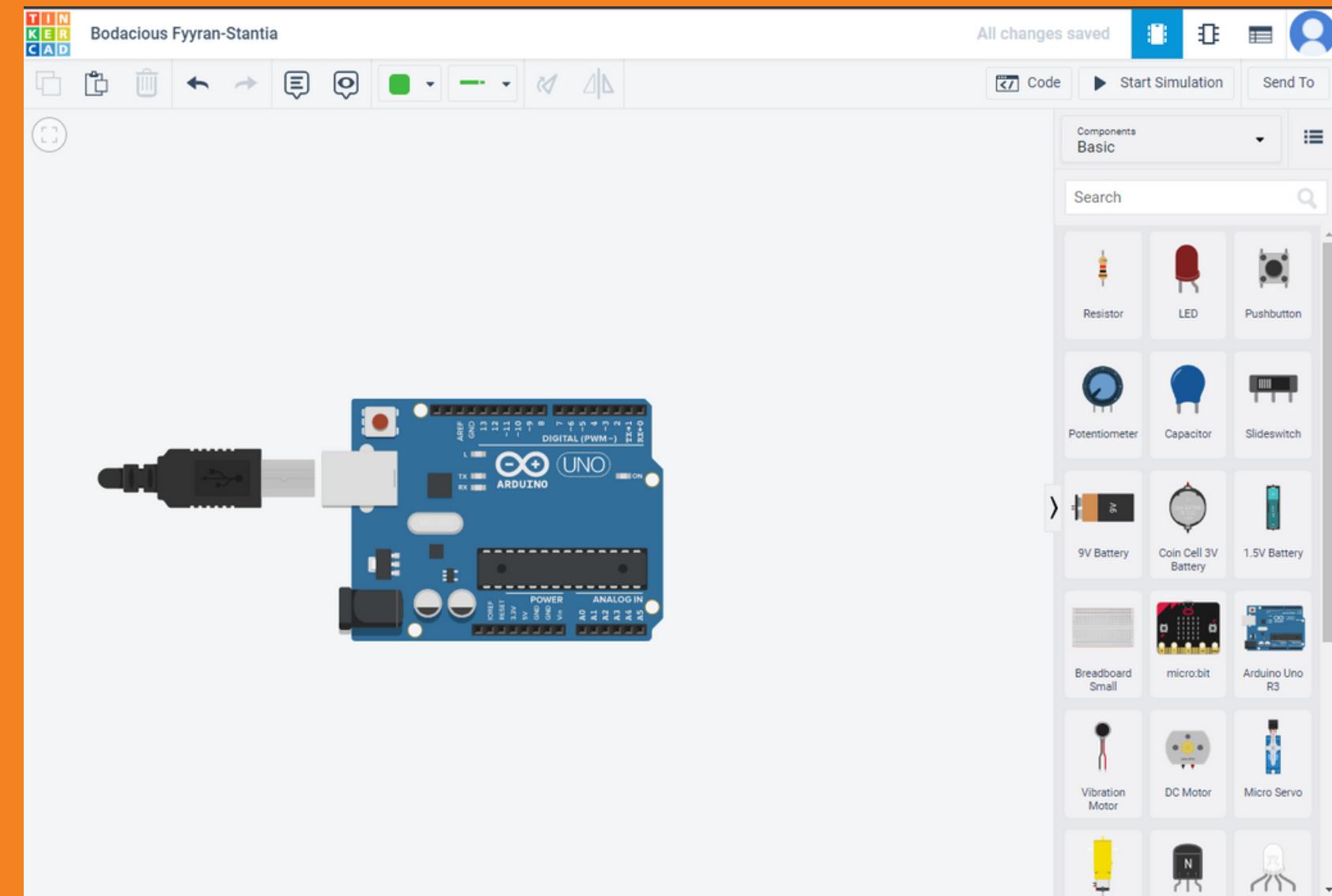
วิทยาศาสตร์ + เทคโนโลยีและวิศวกรรม



Example Arduino Project



- ไปที่เว็บ <https://www.tinkercad.com/>
- สร้าง account โดยเลือก personal account ในช่อง starters เลือก Arduino แล้วลากตามรูปมาวาง จากนั้นกดที่ Code และเลือก Text และกด Start Simulation



Arduino Sketch

```
1 void setup()
2 {
3
4 }
5
6 void loop()
7 {
8
9 }
```

การประกาศตัวแปร

สามารถประกาศได้ 2 รูปแบบ

1 #define

2 ประกาศตามรูปแบบภาษาซีกี้ไป

- โดยกำหนดชนิดของตัวแปรเสนอ **int**, **float**, **char**, **String**

```
1 #define num1 10 // ประกาศให้ num1 เท่ากับ 10
2 int num2 = 5; // ประกาศให้ num2 เท่ากับ 5
3 float num3 = 4.2; // ประกาศให้ num3 เท่ากับ 4.2
4 char c1 = 'a'; // ประกาศให้ c1 เท่ากับ อักษร a 1 ตัว
5 String str1 = "Hello World!"; // ประกาศให้ str1 เท่ากับ Hello World!
6
7 void setup()
8 {
9     // คือการคอมเม้นโค้ด จะไม่อ่านอะไรมีความที่หลังจาก //
10 }
11
12 void loop()
13 {
14 }
```

Serial.begin | Serial.println

```
void setup() {  
    // put your setup code here, to run once:  
    Serial.begin(9600);          <- ถ้าต้องการใช้ Serial Monitor  
    Serial.println("Start!");    จะเป็นต้องมีคำสั่งนี้เสมอ  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
    Serial.println("Hello World!");  
}
```

ลองแสดง ข้อมูลด้านล่างผ่าน Serial Monitor

เขียนใน setup() ไม่ขึ้นและ ลองดูได้ครับ

- ชื่อ
- อายุ
- โรงเรียน

Serial.println | Serial.print

```
#define x 10 //  
int y = 5;  
  
void setup() {  
    // put your setup code here, to run once:  
    Serial.begin(9600);  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
    Serial.print (x+y);  
}
```

Operator	Meaning
+	บวก
-	ลบ
*	คูณ
/	หาร

ลองเปลี่ยน + เป็นอย่างอื่นดูสิ

if | else if | else

```
#define x 10 //  
int y = 5;  
  
void setup() {  
    // put your setup code here, to run once:  
    Serial.begin(9600);  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
    if(x > y){  
        Serial.println("x is greater than y");  
    }else if(x == y){  
        Serial.println("x is equal y");  
    }else{  
        Serial.println("y is greater than x");  
    }  
}
```

Operator	Meaning
==	เท่ากับ
!=	ไม่เท่ากับ
>	มากกว่า
>=	มากกว่าหรือเท่ากับ
<	น้อยกว่า
<=	น้อยกว่าหรือเท่ากับ

ให้น้องๆ ลองเปลี่ยนตัวเลขในตัวแปร
และสังเกตุผลลัพธ์

Serial.available | Serial.parseInt

```
int incomingdata = 0; // for incoming serial data

void setup() {
    Serial.begin(9600); // opens serial port, sets data rate to 9600 bps
}

void loop() {
    // check if data is available
    if (Serial.available() > 0) {
        // read the incoming int:
        incomingdata = Serial.parseInt();

        // prints the received data
        Serial.print("I received: ");
        Serial.println(incomingdata);
    }
}
```

ตัวอย่างการส่งและรับค่าตัวเลขจาก Serial Monitor

Serial.available | Serial.readString

```
String incomingdata = " "; // for incoming serial data

void setup() {
    Serial.begin(9600); // opens serial port, sets data rate to 9600 bps
}

void loop() {
    // check if data is available
    if (Serial.available() > 0) {
        // read the incoming byte:
        incomingdata = Serial.readString();

        // prints the received data
        Serial.print("I received: ");
        Serial.println(incomingdata);
    }
}
```

ตัวอย่างการส่งและรับค่าตัวอักษรหรือข้อความจาก Serial Monitor

ตัวอย่างการเอาค่าที่รับเข้ามาจาก Serial Monitor มาใช้ต่อ

```
int incomingdata = 0; // for incoming serial data

void setup() {
    Serial.begin(9600); // opens serial port, sets data rate to 9600 bps
}

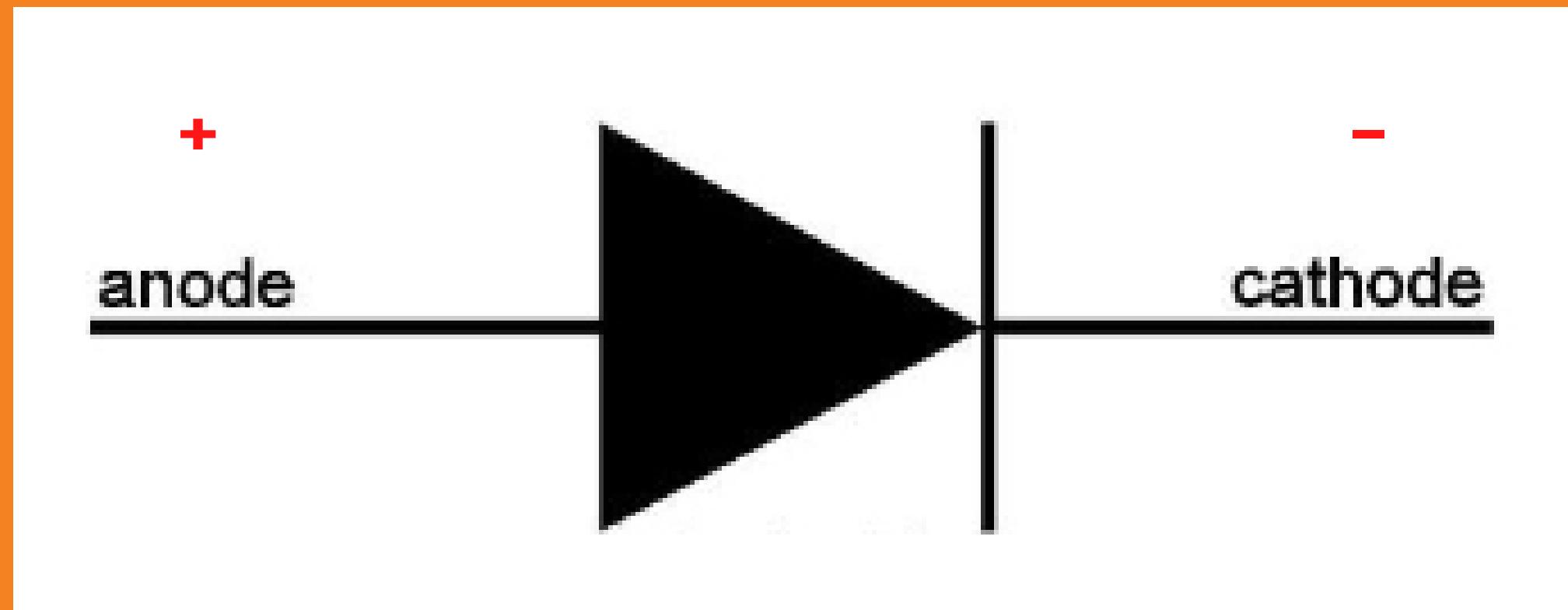
void loop() {
    // check if data is available
    if (Serial.available() > 0) {
        // read the incoming byte:
        incomingdata = Serial.parseInt();

        // prints the received data
        Serial.print("I received: ");
        Serial.println(incomingdata);

        if(incomingdata > 10){
            Serial.println("This number is greater than 10");
        }else{
            Serial.println("This number is equal or lower than 10");
        }
    }
}
```

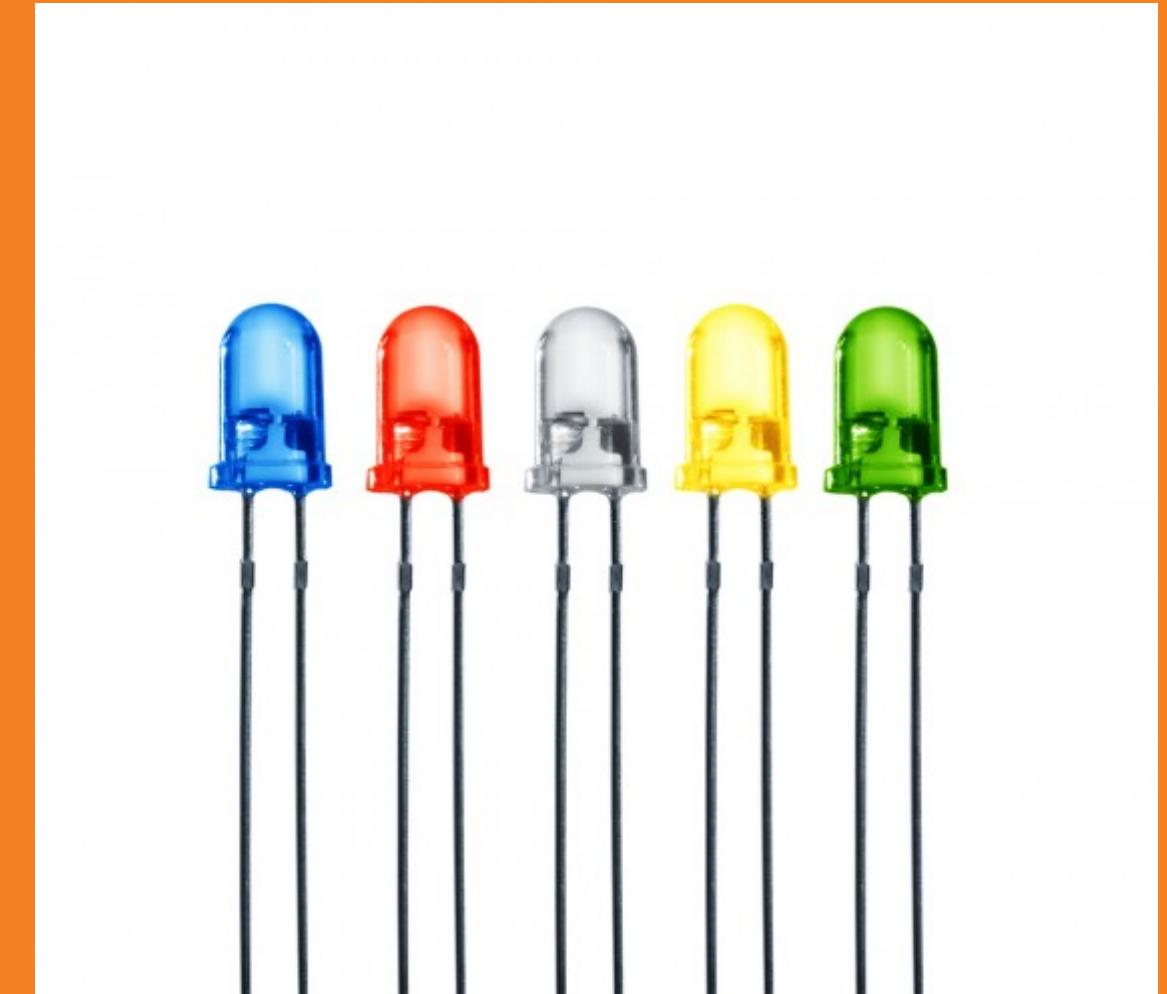
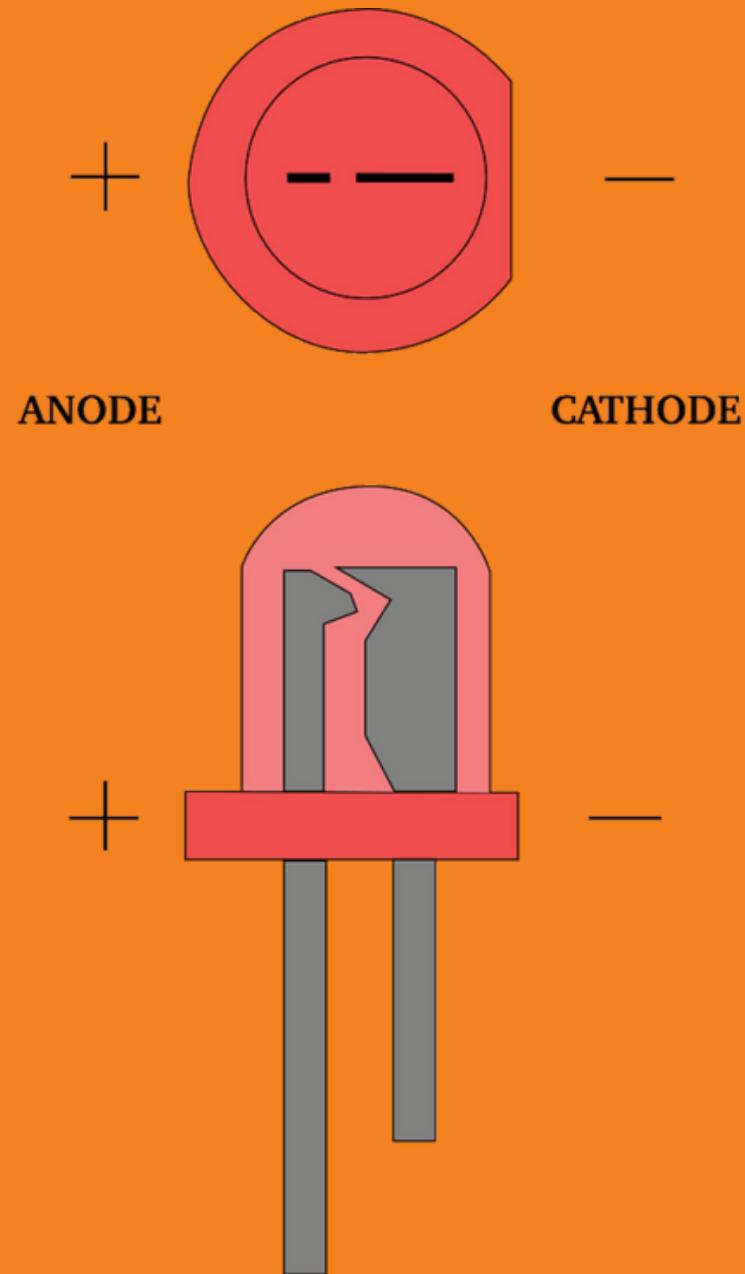
Basic circuit for LED DIODS

อุปกรณ์ที่ใช้ในการควบคุมการทำงานไฟฟ้าคือไดโอดจะยอมให้กระแสไฟฟ้าไหลผ่านในทิศทางเดียว (ใบอัสตรง) และกันการไหลในทิศทางตรงกันข้าม (ใบอัสย้อนกลับ)



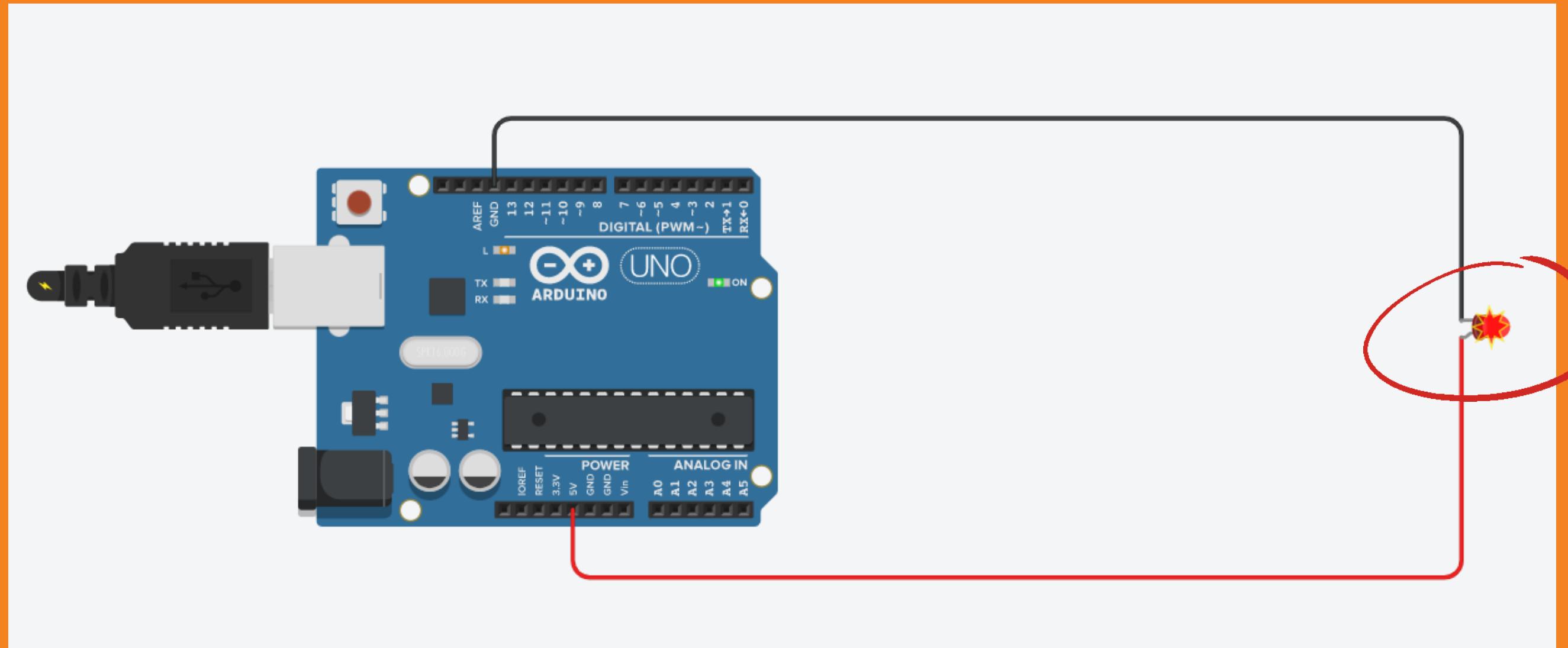
LIGHT EMITTING DIODS

ໄດໂອດເປົ່ງແສງ (Light Emitting Diode) ມັກ ເຮັດວຽກວ່າ ແອລວິດ (LED) ເປັນໄດໂອດທີ່
ອອກແບບມາເປັນພິເສດໂດຍເນື້ອໄດ້ຮັບແຮງດັນໄບອັສຕຽງ ຈະເປົ່ງແສງອອກມາໄດ້ ທີ່ມີກົ້ງ ສີແດງ ເນື່ວ
ຝ່າ ສ້າມ ເຫຼືອງ ວລຢ



LIGHT EMITTING DIODS

LED โดยทั่วไปกระแสสูงสุดที่ใช้จะไม่เกิน 20 mA หากเกินหลอดอาจชำรุดได้

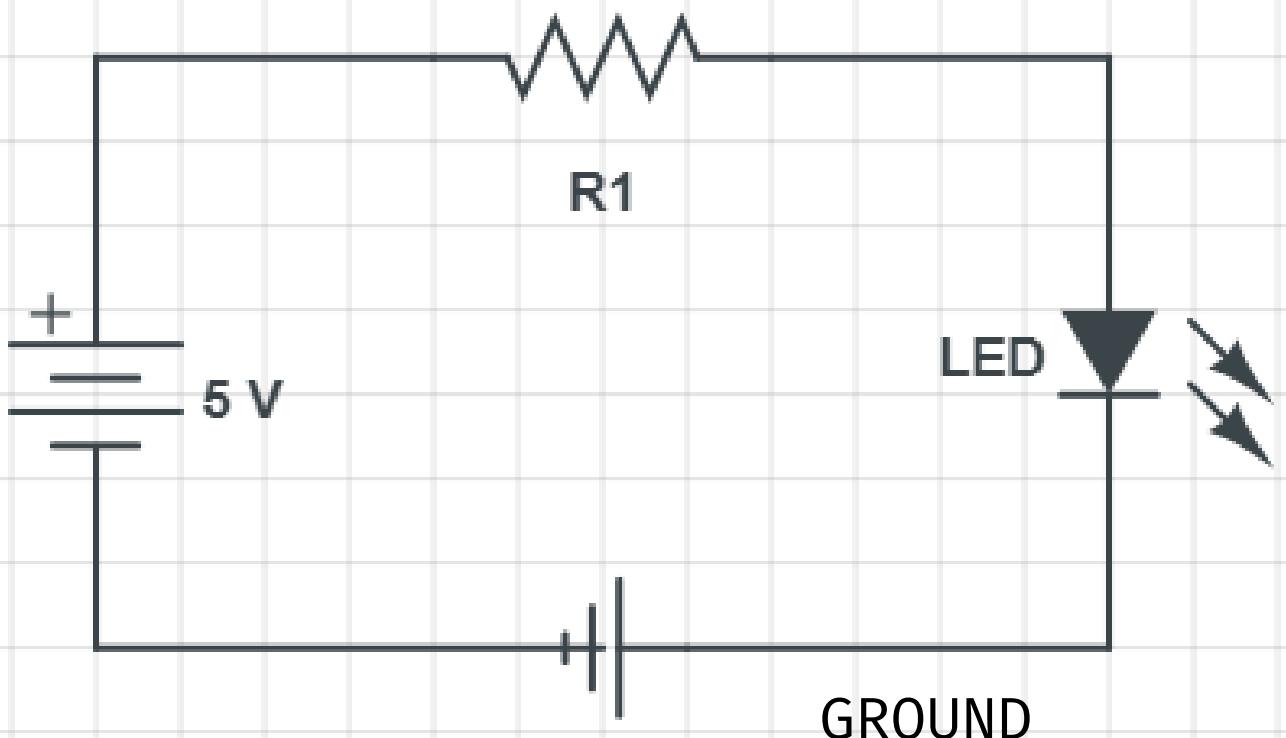


เพราะนั้น เราจึงต้องนำ ตัวต้านทาน หรือ R เข้ามาช่วย



LIGHT EMITTING DIODS

ตัวอย่างการต่อวงจร LED แบบ Diagram

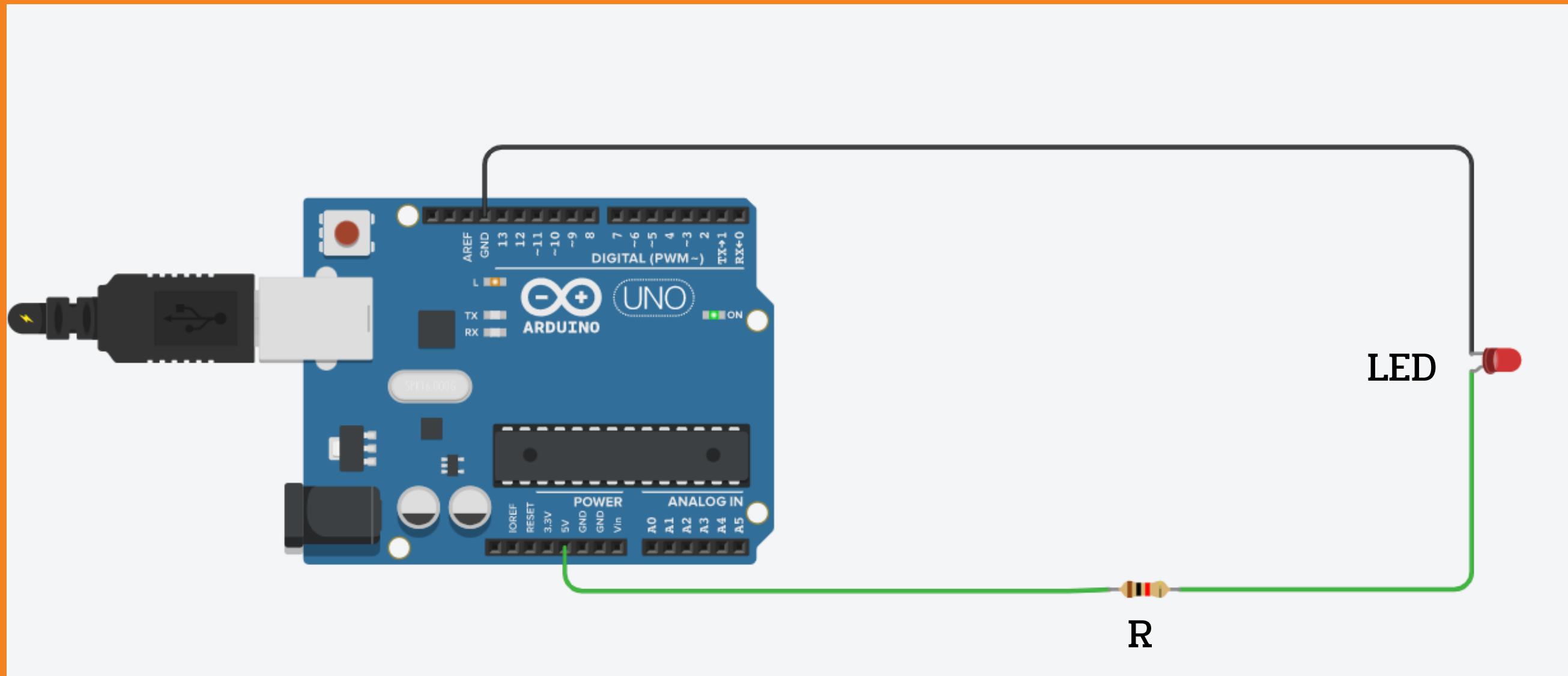


Trick การจำ

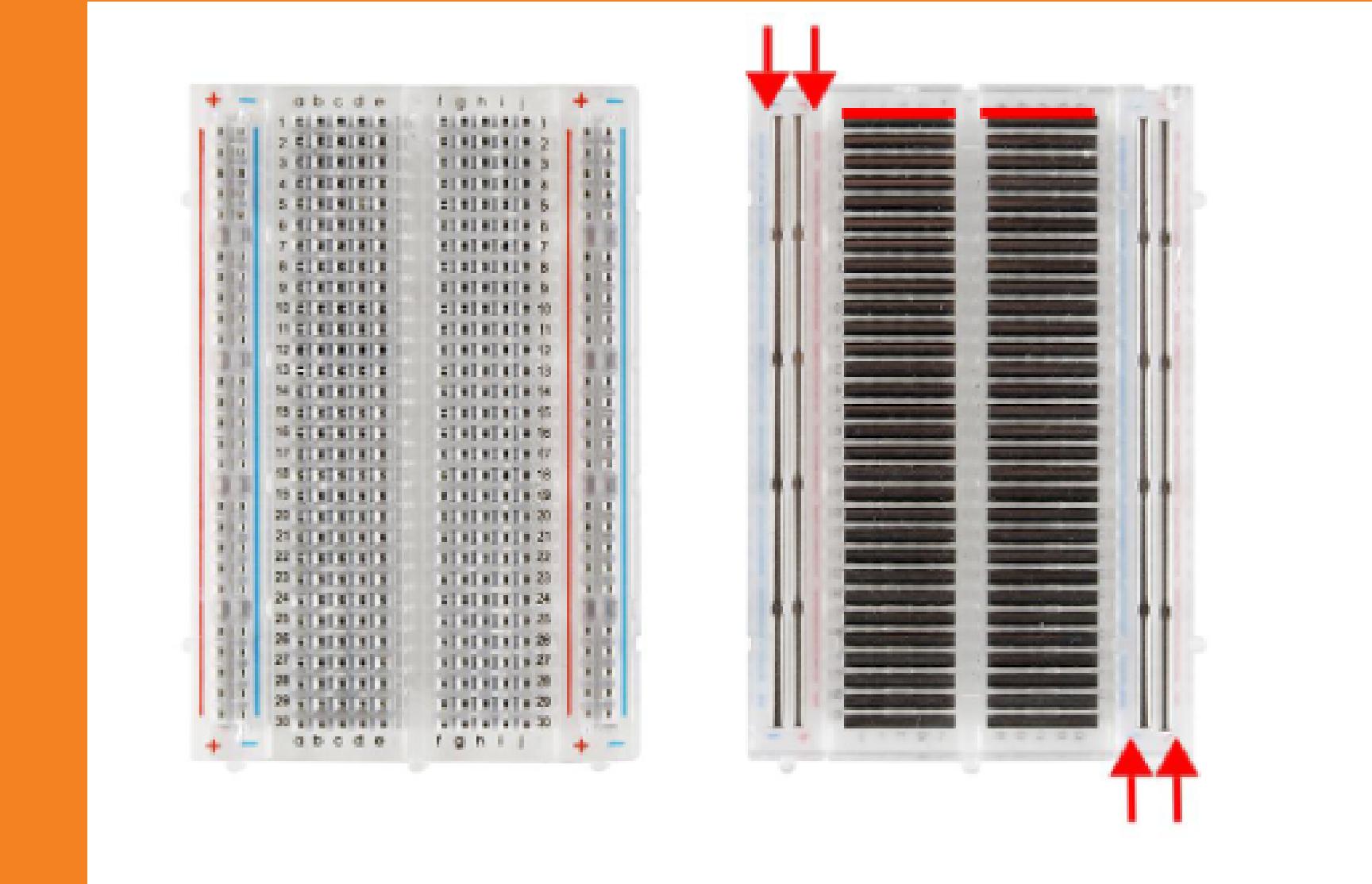
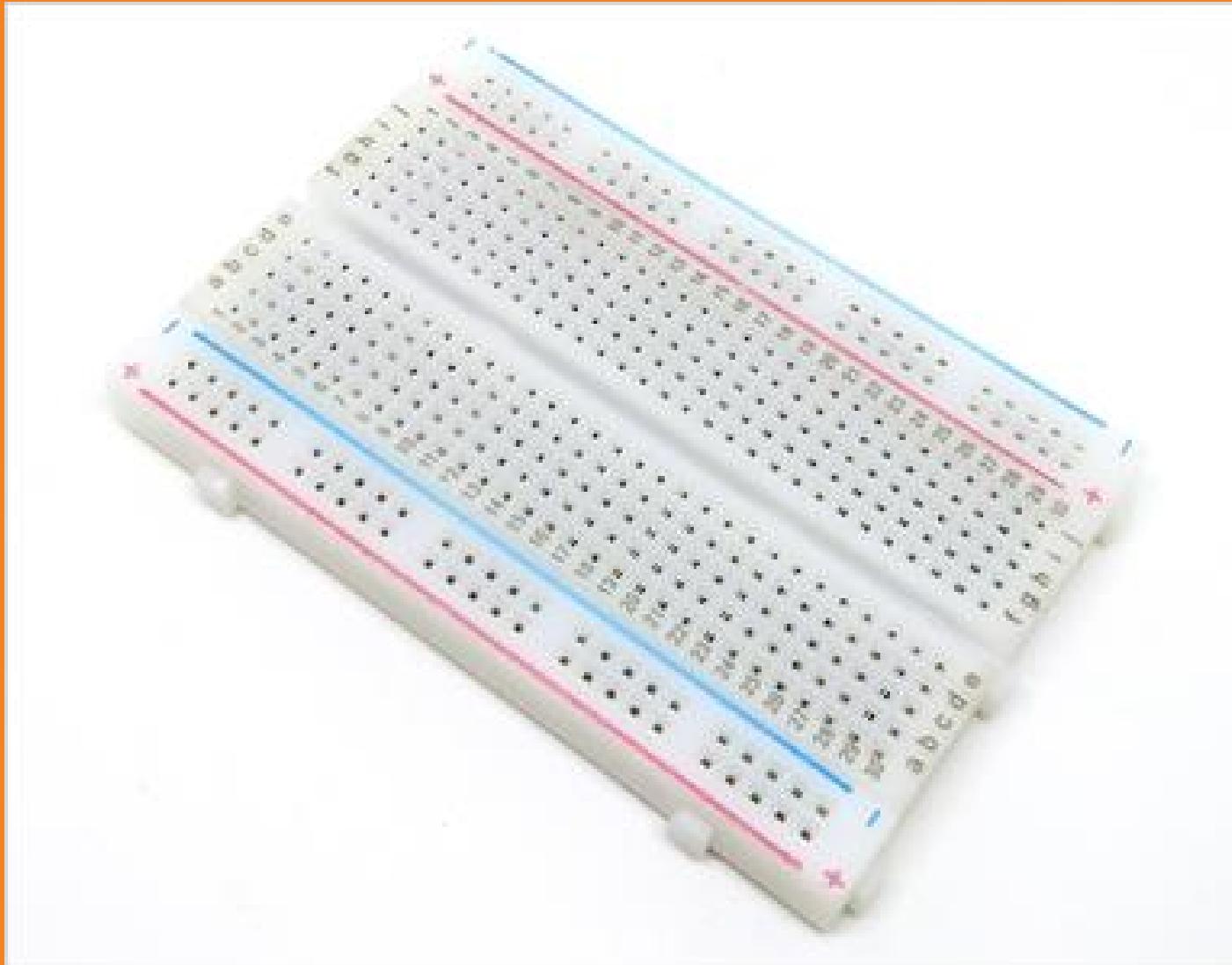
1. ข้อบวก ต้องต่อ กับ กระแสที่เข้ามาเสมอ
2. อย่าลืมต่อตัวต้านทานด้วยเดียวใหม่หรือจะเปิดได้
3. ข้อลบ ต้องต่อ กับ ภานุยเสมอ

LIGHT EMITTING DIODS

ຕັວອຢາງການຕ່ວງຈຽດຕາມ Diagram ດັ່ງກ່າວ

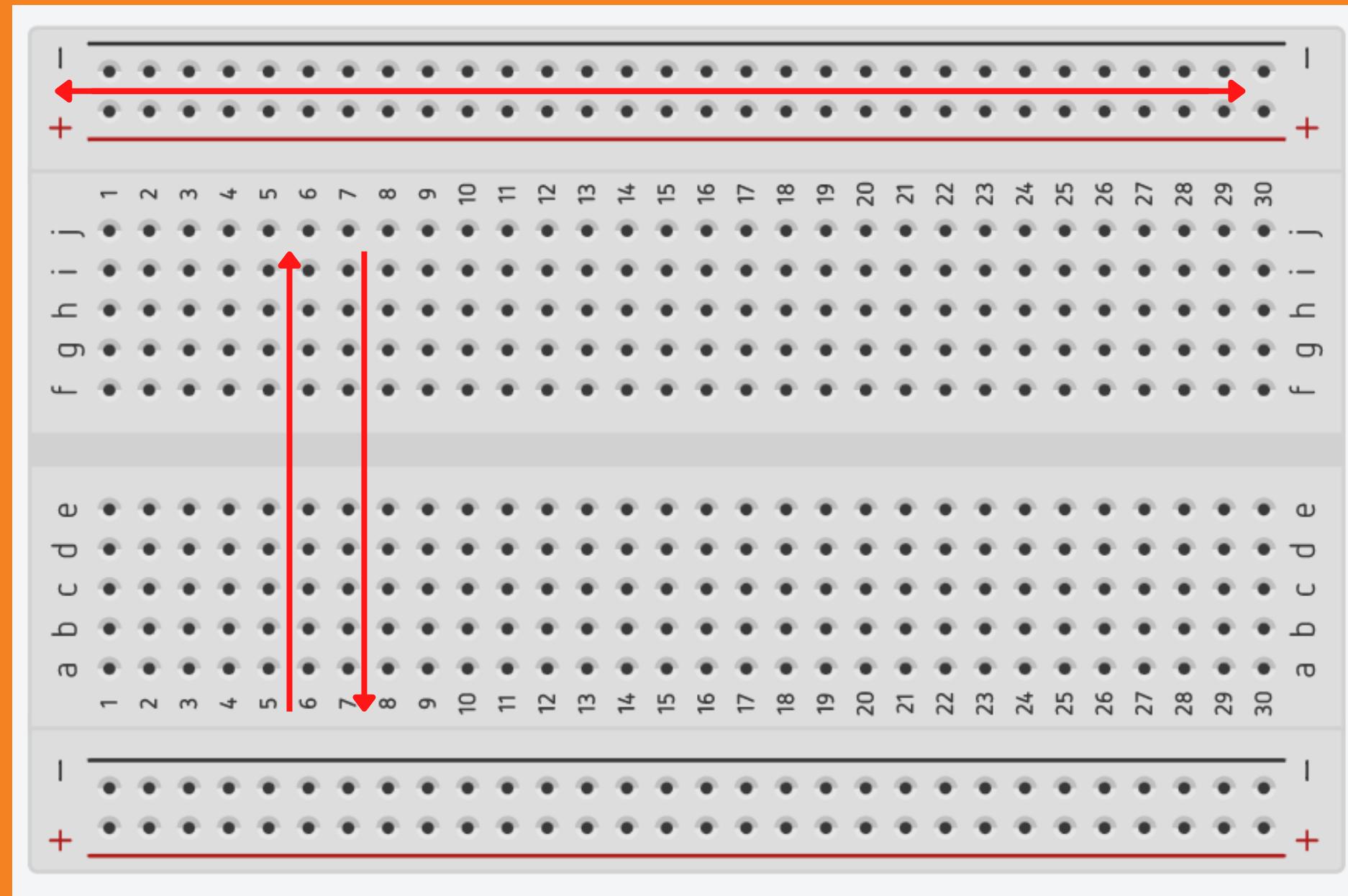


Protoboard ກົດ Breadboard



Protoboard ឬ Breadboard

ការកំងាល់



} ក្រសេវកីតែខ្សោយចាយ
នៃបណ្តុះបណ្តុះ X ឬបណ្តុះបណ្តុះលូ

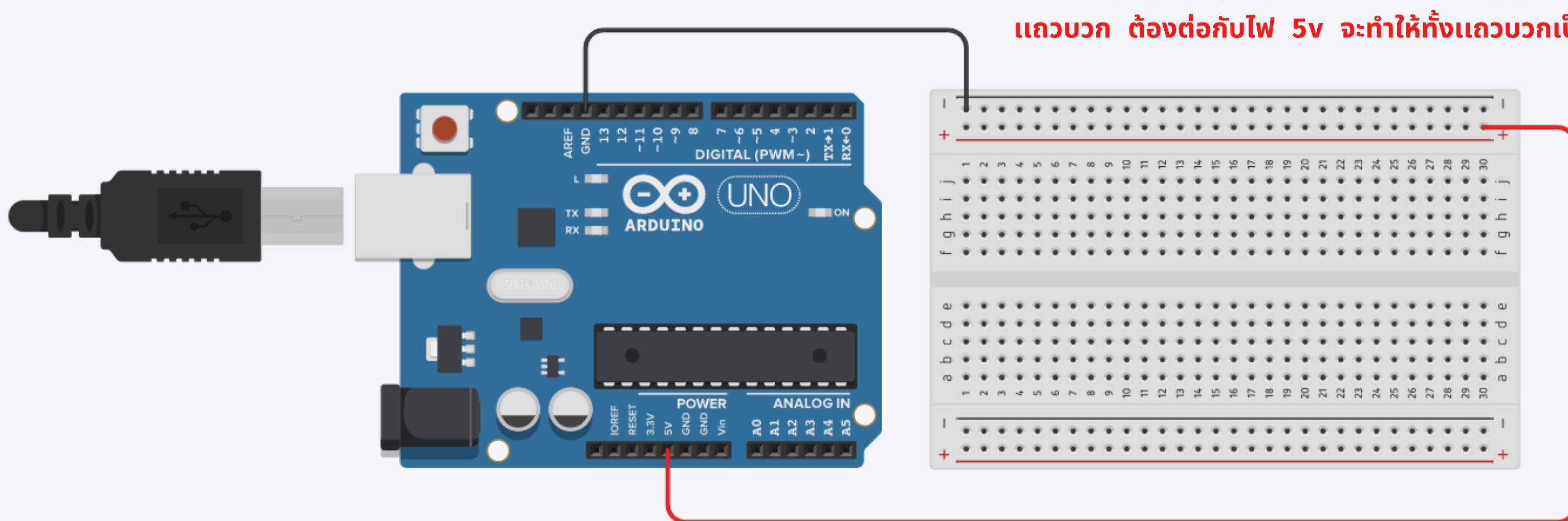
} ក្រសេវកីតែខ្សោយចាយ
នៃបណ្តុះបណ្តុះ Y ឬបណ្តុះបណ្តុះពិនិត្យ

Protoboard หรือ Breadboard

เทคนิคการรวมกระแสไฟ หรือ รวมภายน์ ถ้าหากกรณีช่องจ่ายไฟหรือช่องภายน์ไม่พอ

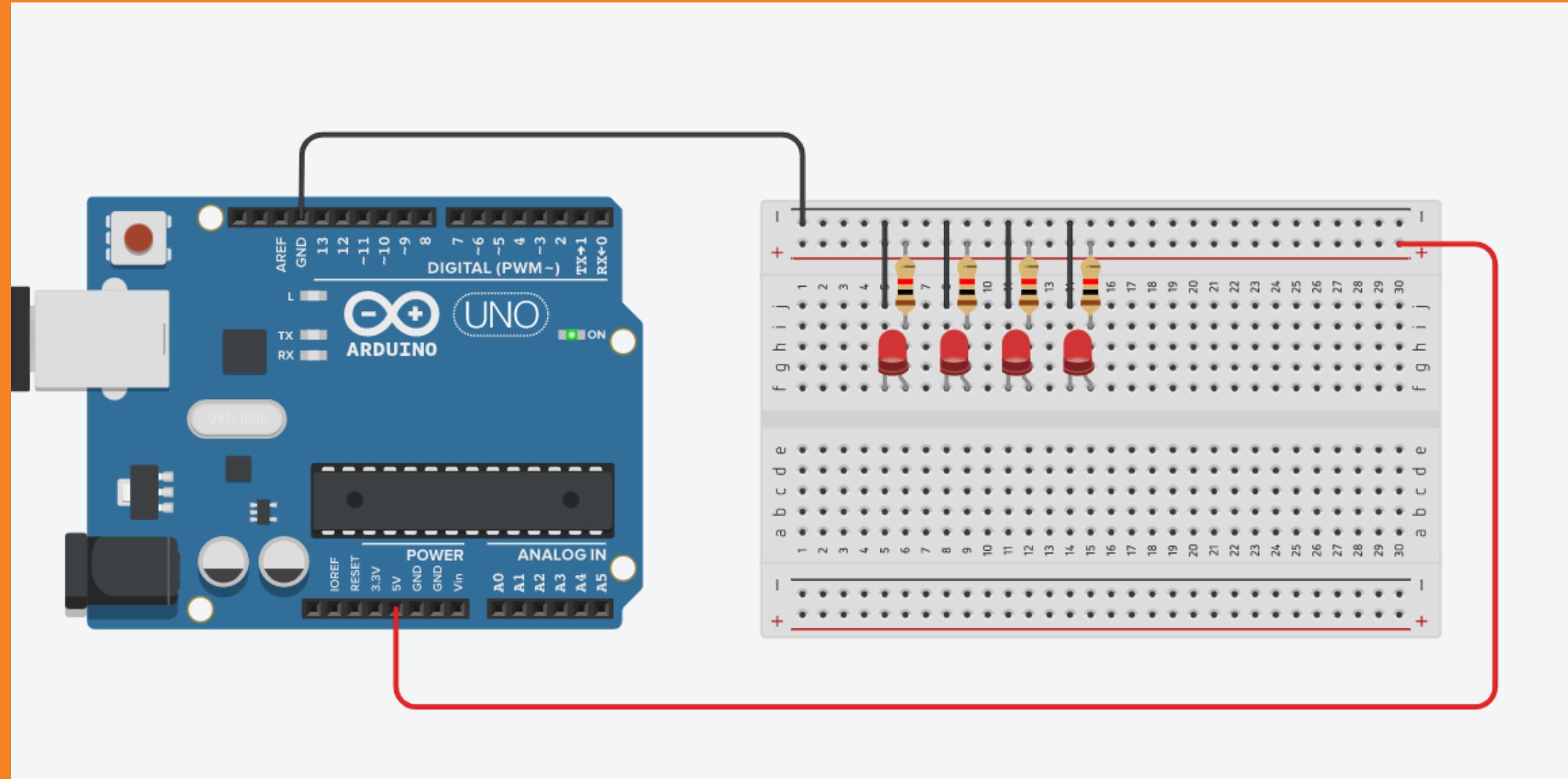
แควลับต่อภายน์ จะทำให้กั้งแควลับเป็นภายน์

แควบวก ต้องต่อภายน์ 5v จะทำให้กั้งแควบวกเป็นไฟ 5v



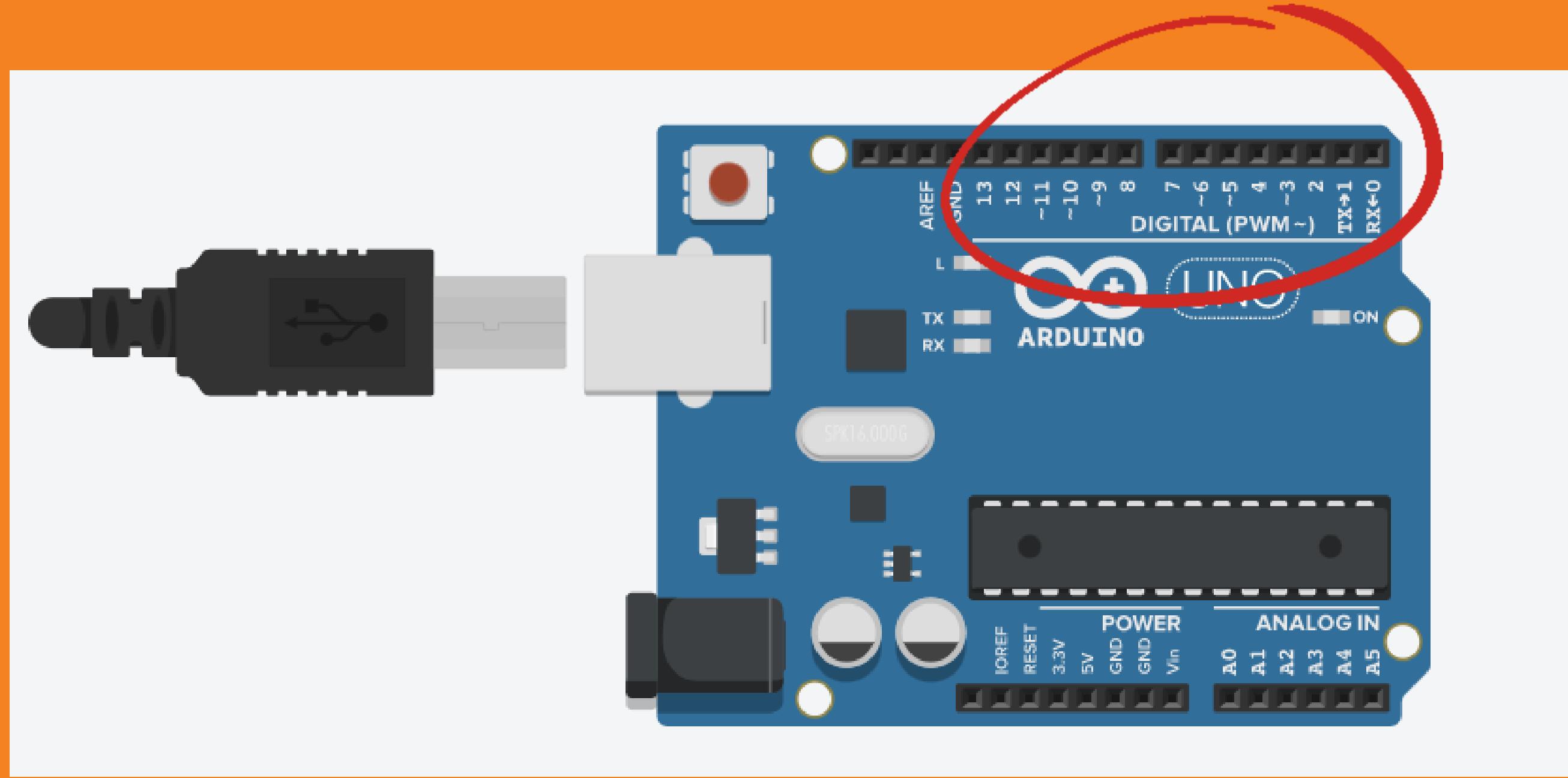
Protoboard នូវ Breadboard

ត្រូវយោងការពេលខែដី 1 ខែទៅ

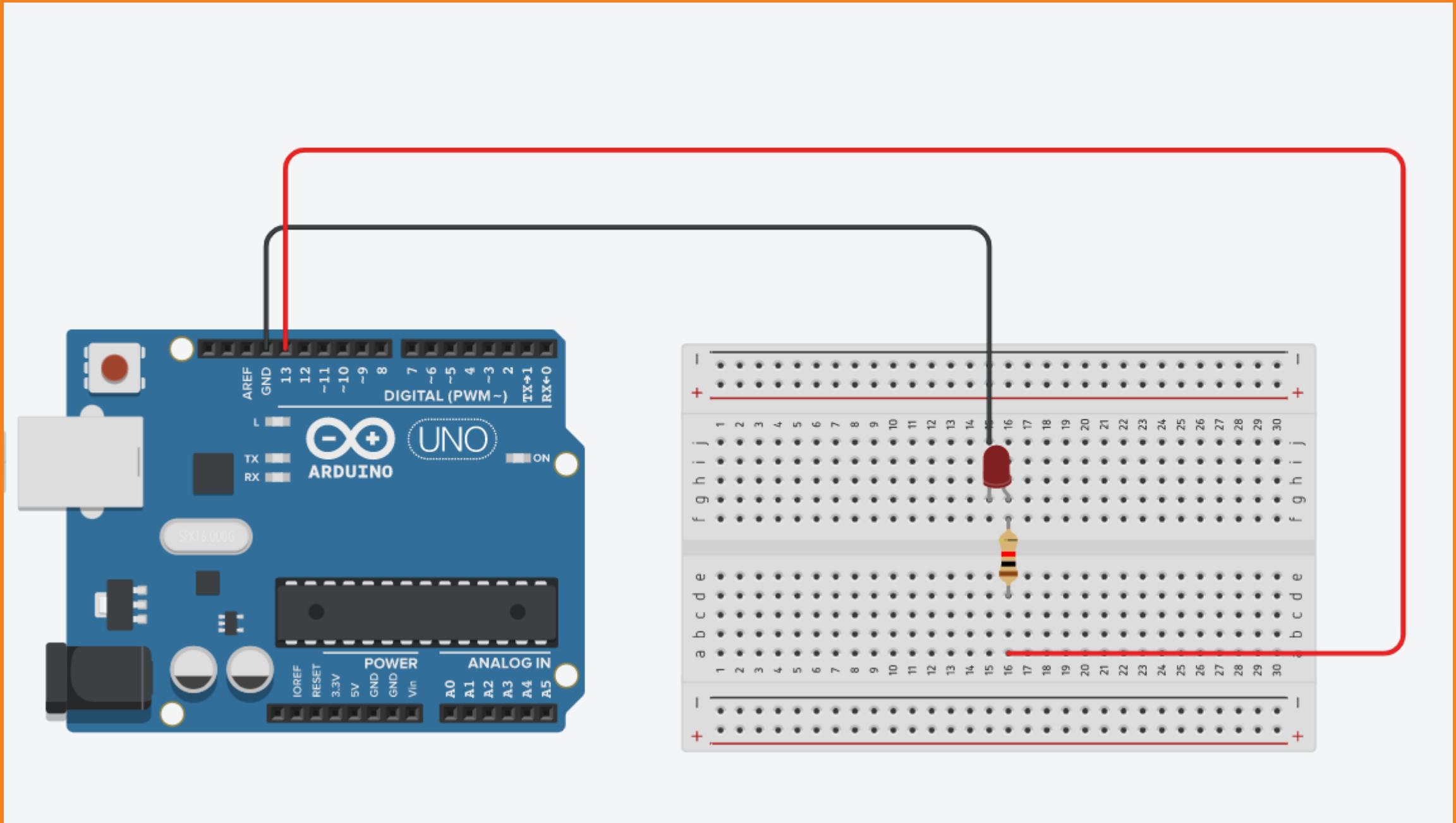
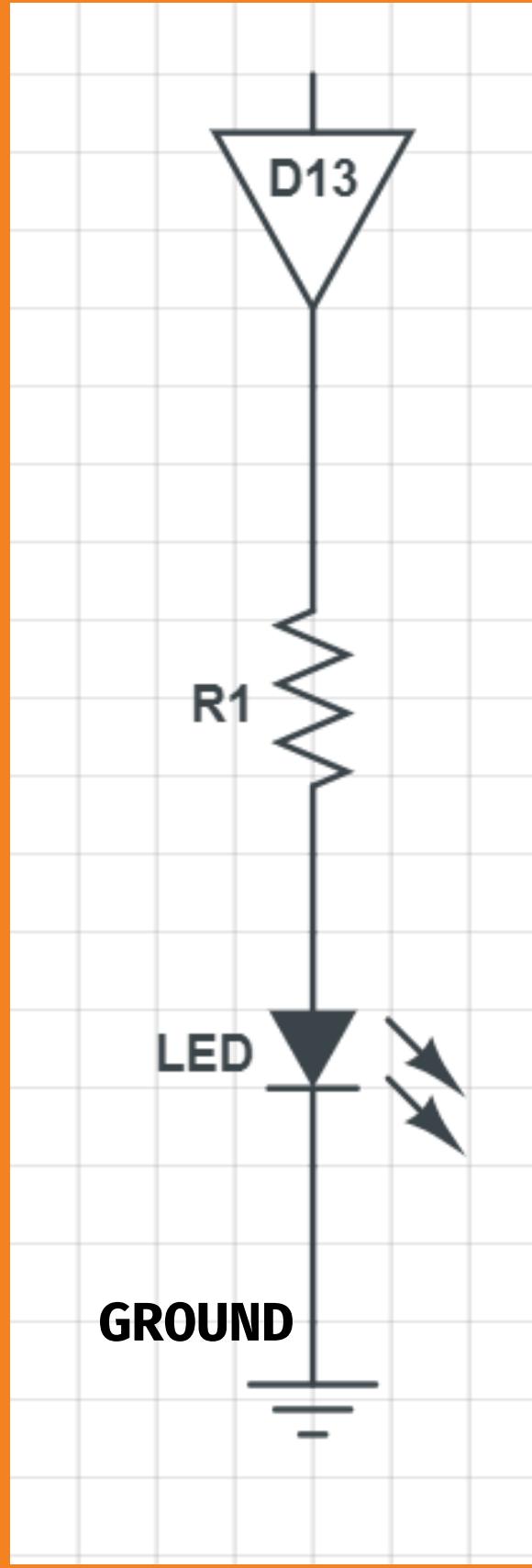


การต่อวงจรโดยใช้ขา I/O

การต่อแบบใช้ไฟ 5v ในการจ่ายไฟนั้น จะไม่สามารถเขียนโปรแกรมควบคุมได้
 เพราะนั้น ถ้าหากต้องการที่จะควบคุมกระแสไฟ ต้องใช้ขา I/O กั้ง 13 ขา จ่ายไฟแทนช่อง 5v



การต่อวงจรโดยใช้ขา I/O



Arduino Sketch

```
void setup()
{
    // put your setup code here, to run once:

}

void loop()
{
    // put your main code here, to run repeatedly:

}
```

DIGITAL OUTPUT

SETUP PINMODE

Syntax:

`pinMode(pin, mode)`

Parameter:

pin: the number of the pin whose mode you wish to set

mode: INPUT, OUTPUT or INPUT_PULLUP.

DIGITAL OUTPUT

DIGITAL OUTPUT PROGRAMMING (ON-OFF)

Syntax:

`digitalWrite(pin, logic)`

Parameter:

pin: the number of the pin whose mode you wish to set

logic : HIGH or LOW.

DIGITAL OUTPUT

Example:

```
#define LED_on_Arduino 13

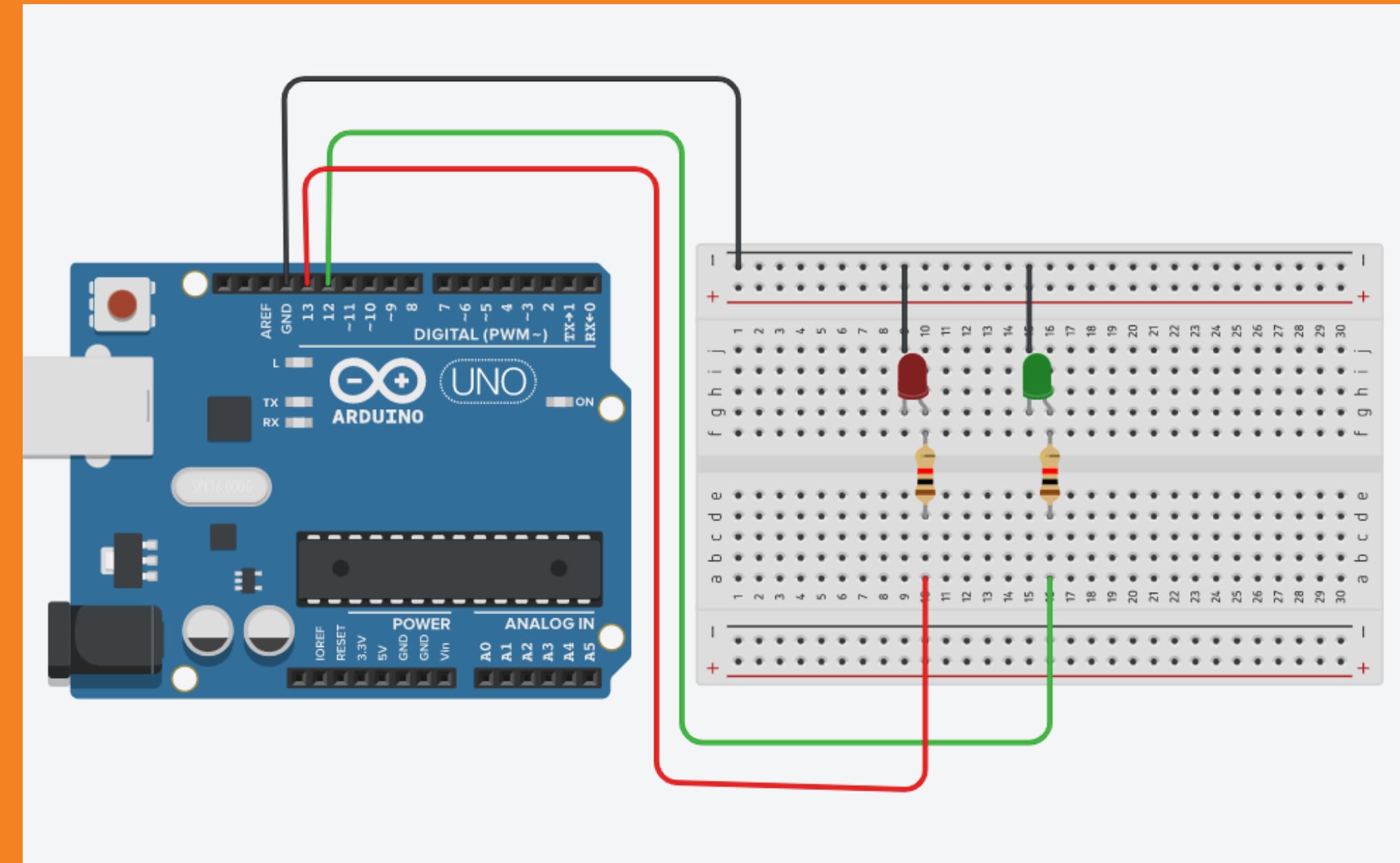
void setup()
{
    pinMode(LED_on_Arduino,OUTPUT); // setup output ←
}

void loop()
{
    digitalWrite(LED_on_Arduino,HIGH); // Pin13 is HIGH ←
    delay(250);
    digitalWrite(LED_on_Arduino,LOW); // Pin13 is LOW ←
    delay(250);
}
```

delay(x)หมายถึงให้หน่วงเวลา เป็นระยะเวลาเท่ากับ x มิลลิวินาที

Activity 1

- ให้ต่อ LED กับบอร์ด Arduino จำนวน 2 ดวง
- เขียนโปรแกรมให้ไฟติดสลับกัน โดยหน่วงเวลาอยู่ที่ 1 วินาที
- ให้วุ่นช้ำแสดงตามข้อก่อนหน้าไปเรื่อยๆ



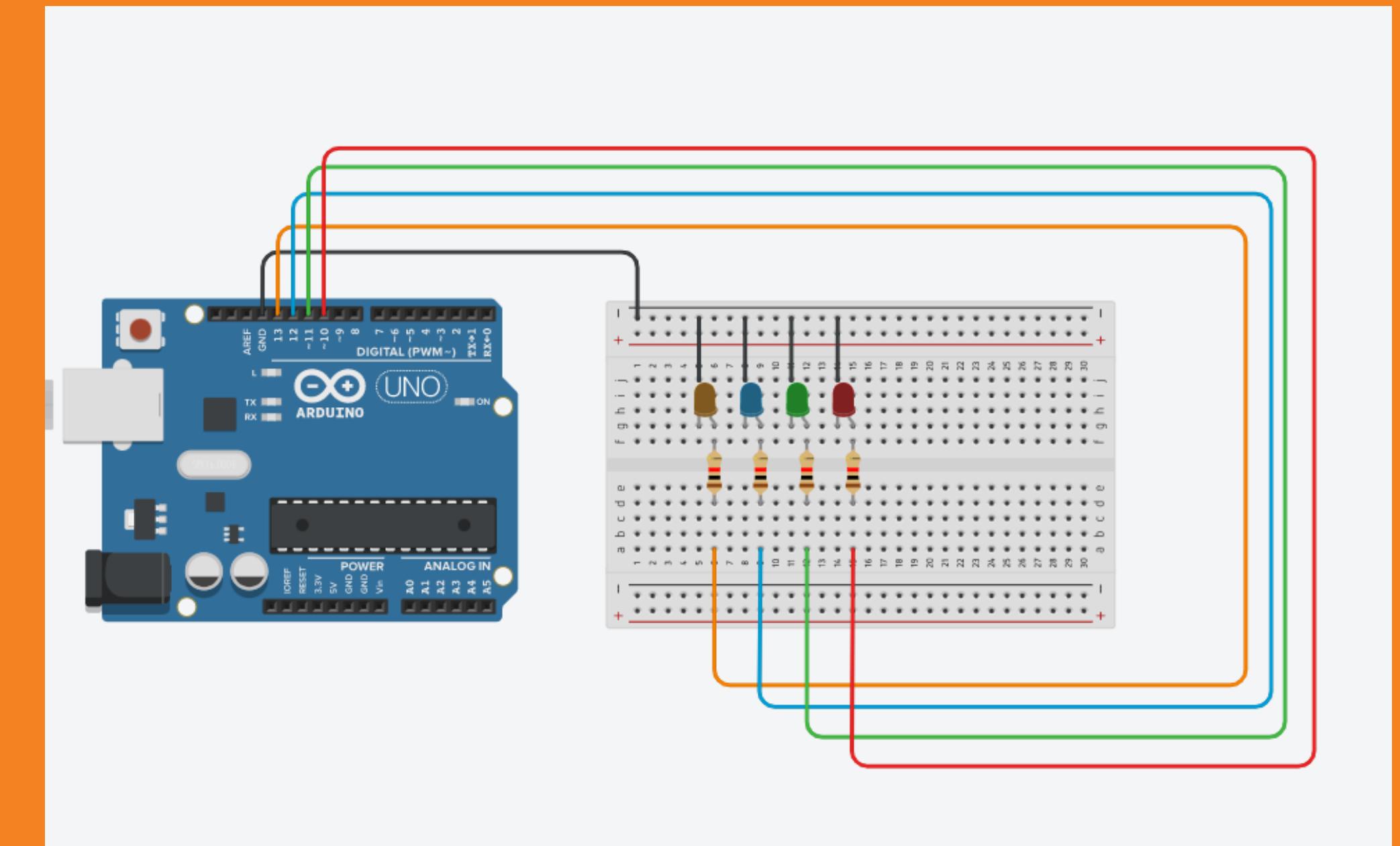
Activity 2

- ให้ต่อ LED กับบอร์ด Arduino จำนวน 4 ดวง

- เขียนโปรแกรมให้แสดง

OFF-OFF-OFF-OFF -> OFF-OFF-OFF-ON -> OFF-OFF-ON-ON -> OFF-ON-ON-ON -> ON-ON-ON-ON
โดยเว้นจังหวะละ 500 ms

- ให้วนซ้ำแสดงตามข้อก่อนหน้าไปเรื่อยๆ



End Hour 1

Sensors

Sensors



Arduino sensors.



In Arduino, sensors are electronic devices that are used to detect changes in the physical environment and convert them into electrical signals that can be read by the Arduino board.



There are many types of sensors that can be used with Arduino, including:

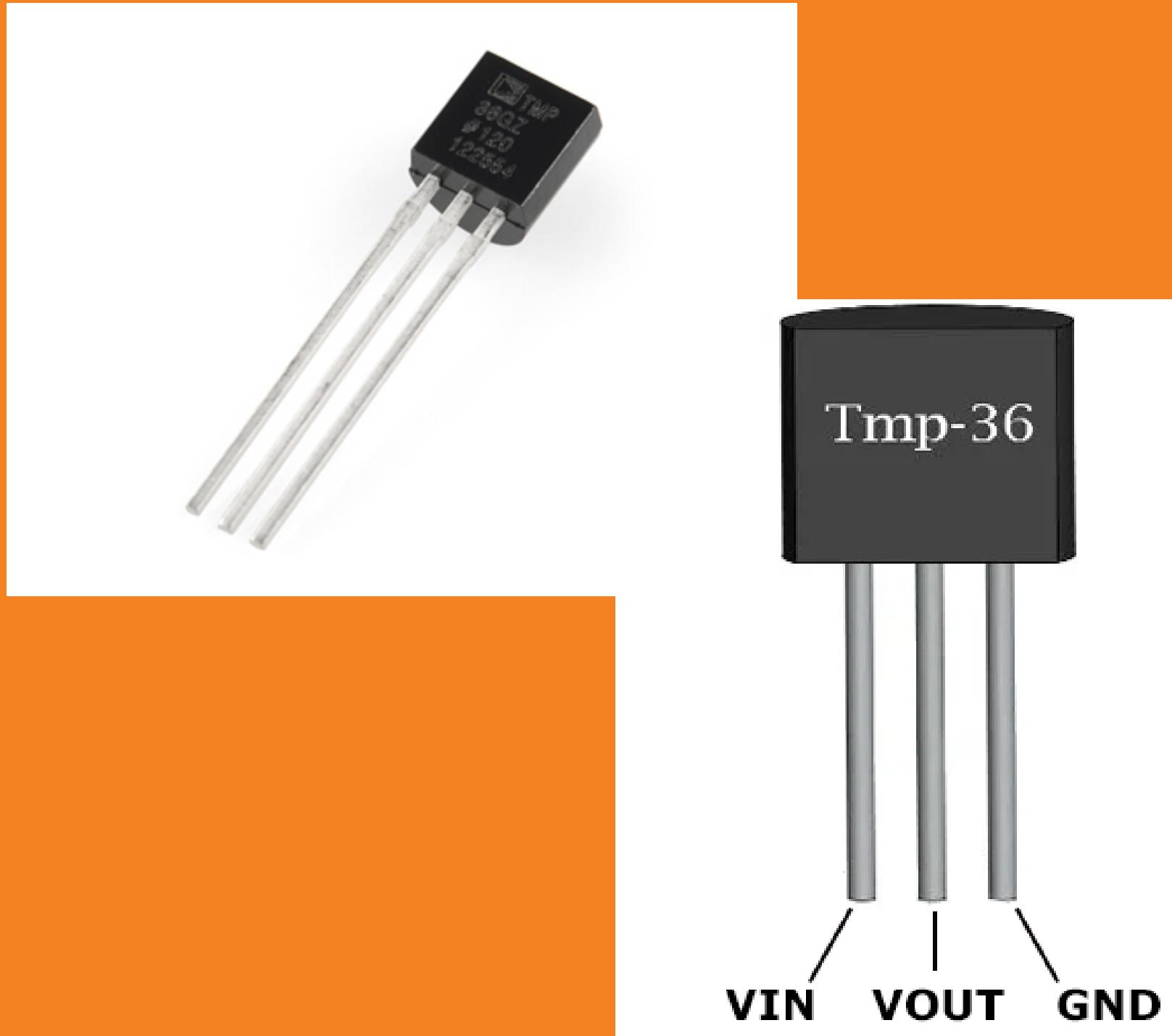
1. Temperature sensors - used to measure temperature changes.
2. Light sensors - used to measure light intensity.
3. Motion sensors - used to detect movement.
4. Sound sensors - used to detect sound levels.
5. Pressure sensors - used to measure pressure changes.
6. Touch sensors - used to detect when an object is touched.
7. Distance sensors - used to measure distance to an object.



These sensors can be connected to the Arduino board via input/output (I/O) pins, which can be programmed to read the signals from the sensors and trigger actions based on the data received. This makes Arduino a powerful tool for building projects that require interaction with the physical world, such as robotics, home automation, and environmental monitoring.

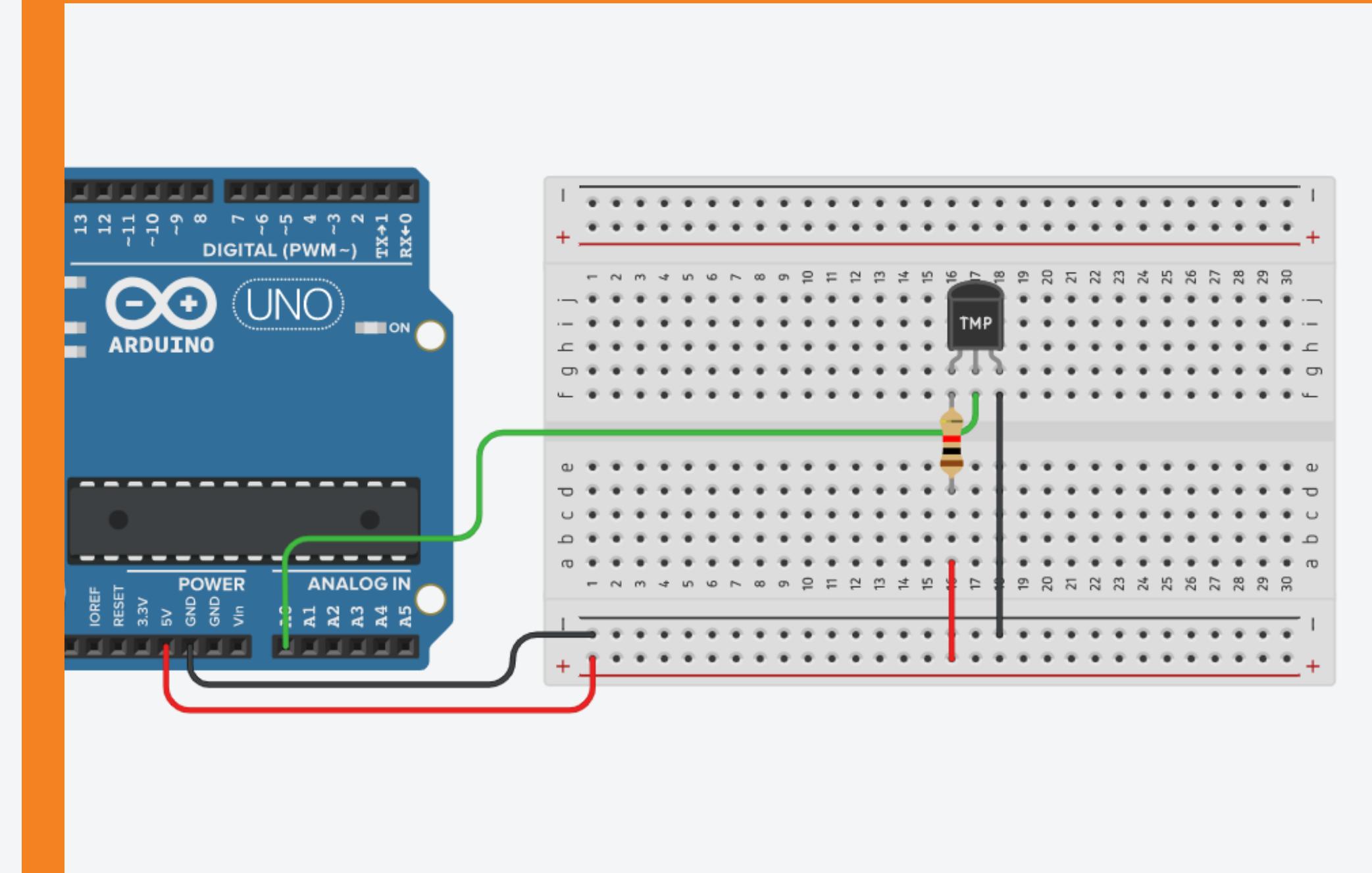
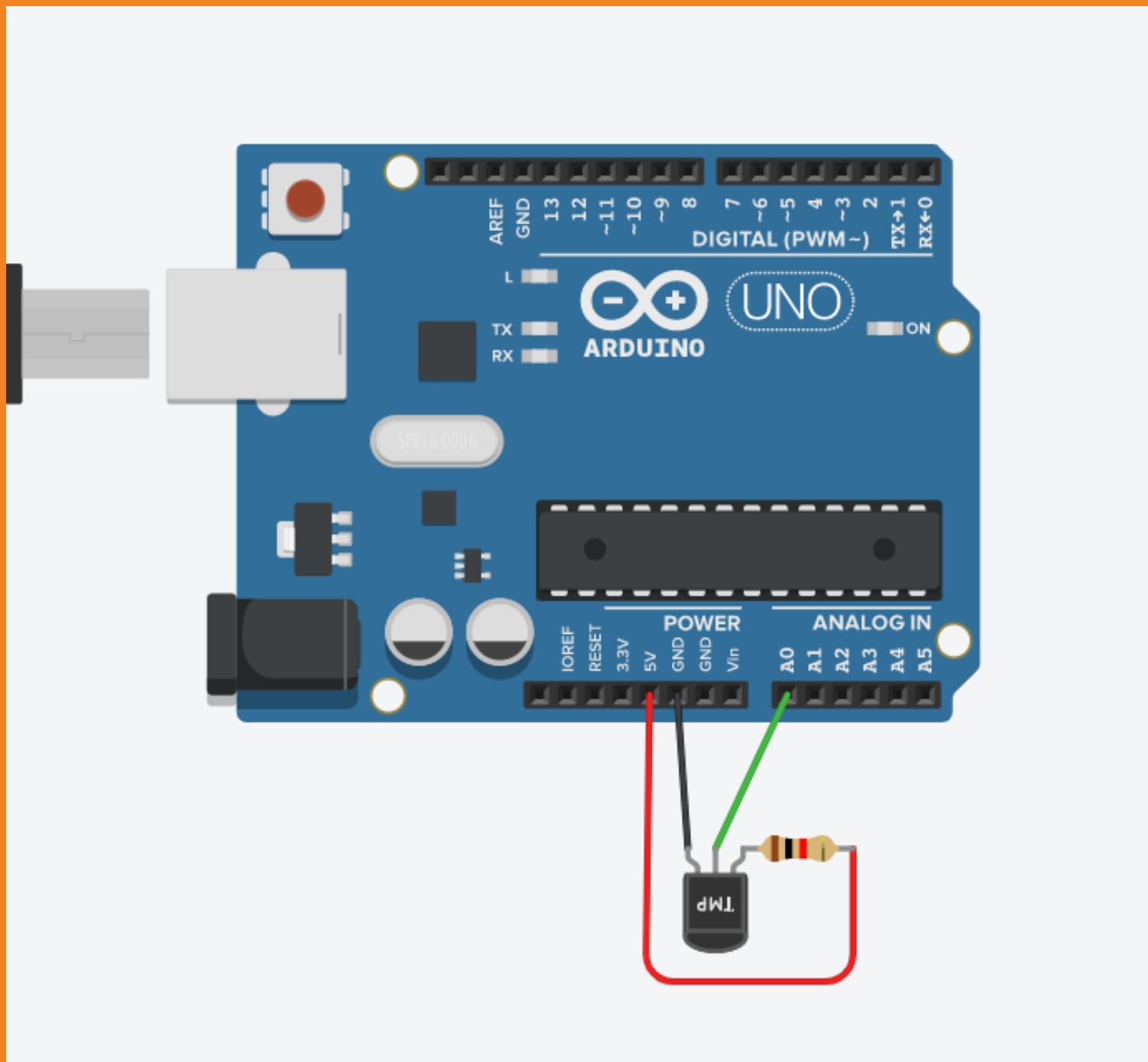


TMP36 Analog Temperature sensor TMP36



IC TMP36 เซนเซอร์วัดอุณหภูมิให้สัญญาณออกมาเป็นแบบ analog ใช้ไฟเลี้ยง 2.7 - 5.5 โวลต์ สามารถวัดได้ในช่วง -40 ถึง 125 องศาเซลเซียส

ຕົວຢ່າງການຕົວອະນຸຍາດ



ตั้งค่า ขา Sensor

```
1 #define sensorPin A0->Analog pin
2
3 int sensorInput;
4 int temp_c;
5
6 void setup() {
7     Serial.begin(9600);
8     pinMode(sensorPin, INPUT);|
9 }
10
```

รับค่าจาก Sensor

analogRead(x)



Analog pin

```
1 #define sensorPin A0
2
3 int sensorInput;
4 int temp_c;
5
6 void setup() {
7     Serial.begin(9600);
8     pinMode(sensorPin, INPUT);
9 }
10
11 void loop() {
12
13     sensorInput = analogRead(sensorPin);
14
15 }
```

รับค่าจาก Sensor

แปลงค่า Analog ให้เป็นค่า องศาเซลเซียส

```
map(((analogRead(A0) - 20) * 3.04), 0, 1023, -40, 125);
```

```
1 #define sensorPin A0
2
3 int sensorInput;
4 int temp_c;
5
6 void setup() {
7     Serial.begin(9600);
8     pinMode(sensorPin, INPUT);
9 }
10
11 void loop() {
12
13     sensorInput = analogRead(sensorPin);
14
15     temp_c = map((analogRead(A0) - 20) * 3.04, 0, 1023, -40, 125);
16
17 }
```

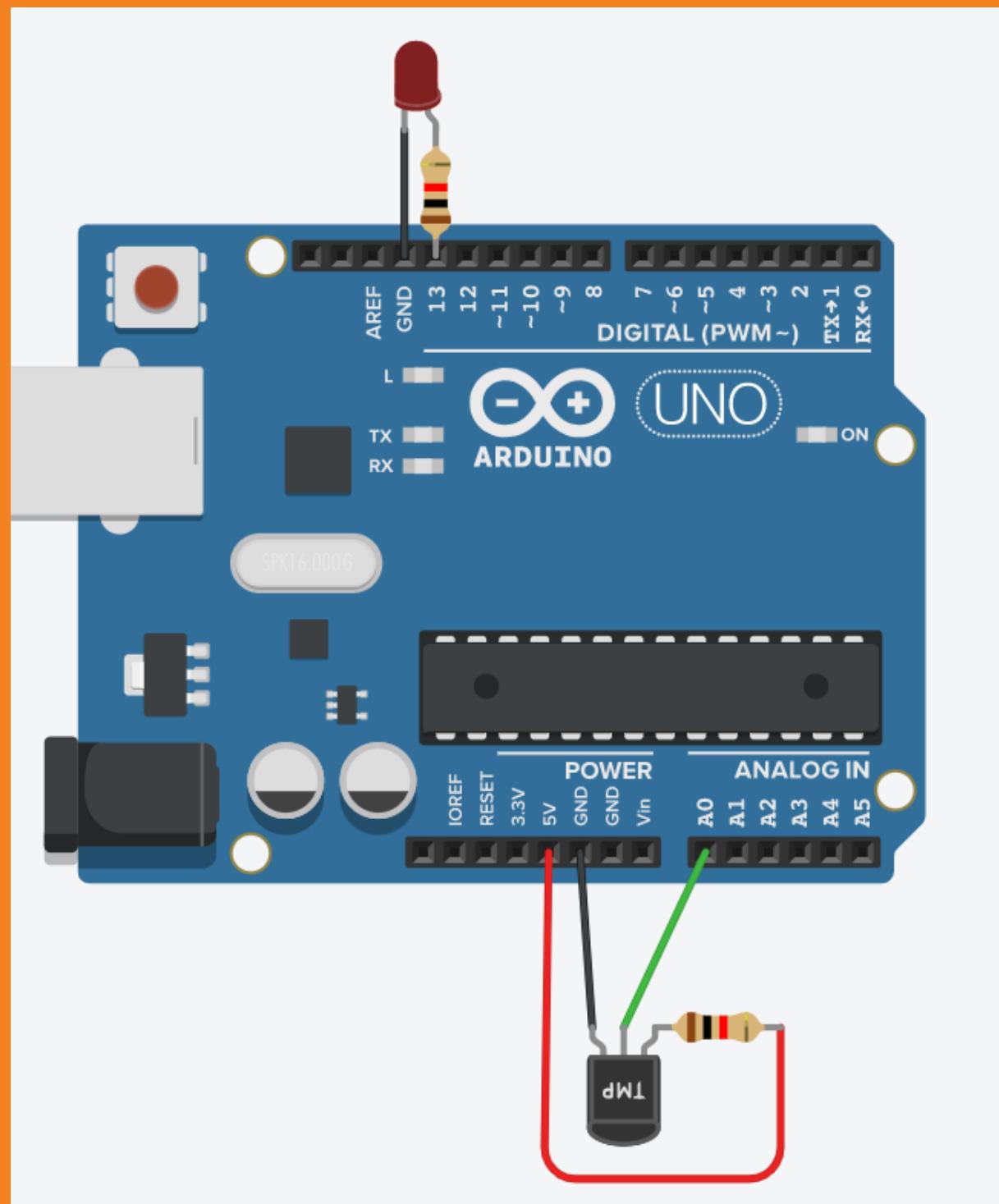
Print ค่า อุณหภูมิเช็ค ด้วยคำสั่ง Serial.println() ใน loop

```
1 #define sensorPin A0
2
3 int sensorInput;
4 int temp_c;
5
6 void setup() {
7     Serial.begin(9600);
8     pinMode(sensorPin, INPUT);
9 }
10
11 void loop() {
12
13     sensorInput = analogRead(sensorPin);
14
15     temp_c = map((sensorInput - 20) * 3.04, 0, 1023, -40, 125);
16
17     Serial.println(temp_c);
18
19 }
```

Activity 3

ระบบแจ้งเตือนไฟไหม้จำลอง

TMP36 + LED



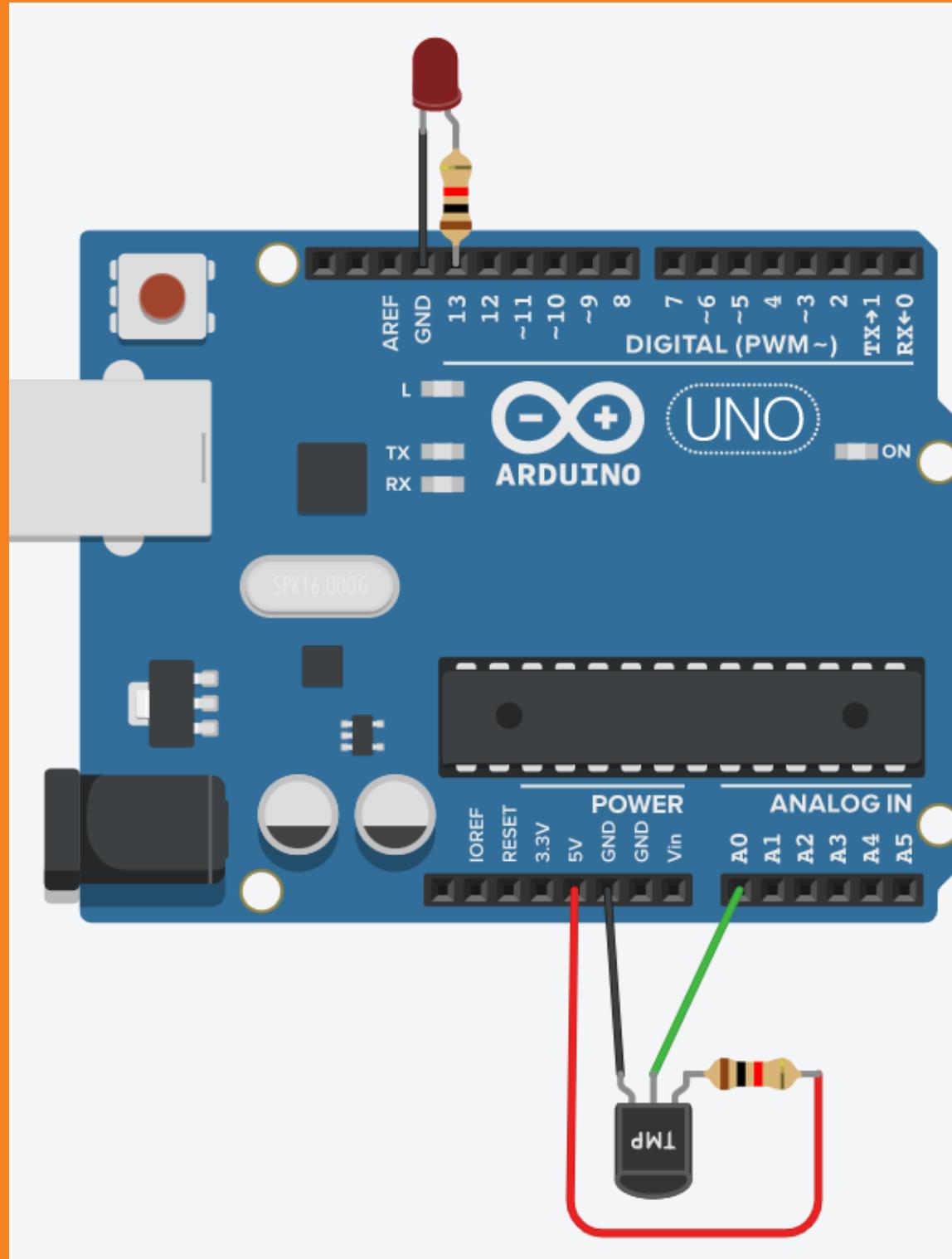
Abilities:

- แสดงค่าอุณหภูมิผ่าน Serial Monitor
- ถ้าหากอุณหภูมิมากกว่า 90 องศา LED จะทำงาน
- ถ้าหากเงื่อนไขอื่นๆ LED จะดับ

Guide:

- ต่อ LED กีชา 13 ตามรูปที่แบบให้
- เขียนกำหนดของ LED ในโปรแกรม
รวมถึงในส่วนของ setup()
- เขียน if else เพิ่มตามเงื่อนไขที่กำหนด

Example



```
1 #define sensorPin A0
2 #define ledpin 13
3
4 int sensorInput;
5 int temp_c;
6
7 void setup(){
8     Serial.begin(9600);
9
10    pinMode(sensorPin, INPUT);
11    pinMode(ledpin, OUTPUT);
12 }
13
14 void loop(){
15
16    sensorInput = analogRead(sensorPin);
17
18    temp_c = map(((sensorInput - 20) * 3.04), 0, 1023, -40, 125);
19
20    Serial.println(temp_c);
21
22    if(temp_c > 90){ //ถ้าอุณหภูมิมากกว่า 90 องศาเซลเซียส ให้ LED ทำงาน
23        digitalWrite(ledpin, HIGH);
24    }else{           //ถ้าอุณหภูมิไม่มากกว่า 90 องศาเซลเซียส ให้ LED ไม่ทำงาน
25        digitalWrite(ledpin, LOW);
26    }
27
28 }
```

ระบบแจ้งเตือนไฟใหม่จำลอง

End Hour 2

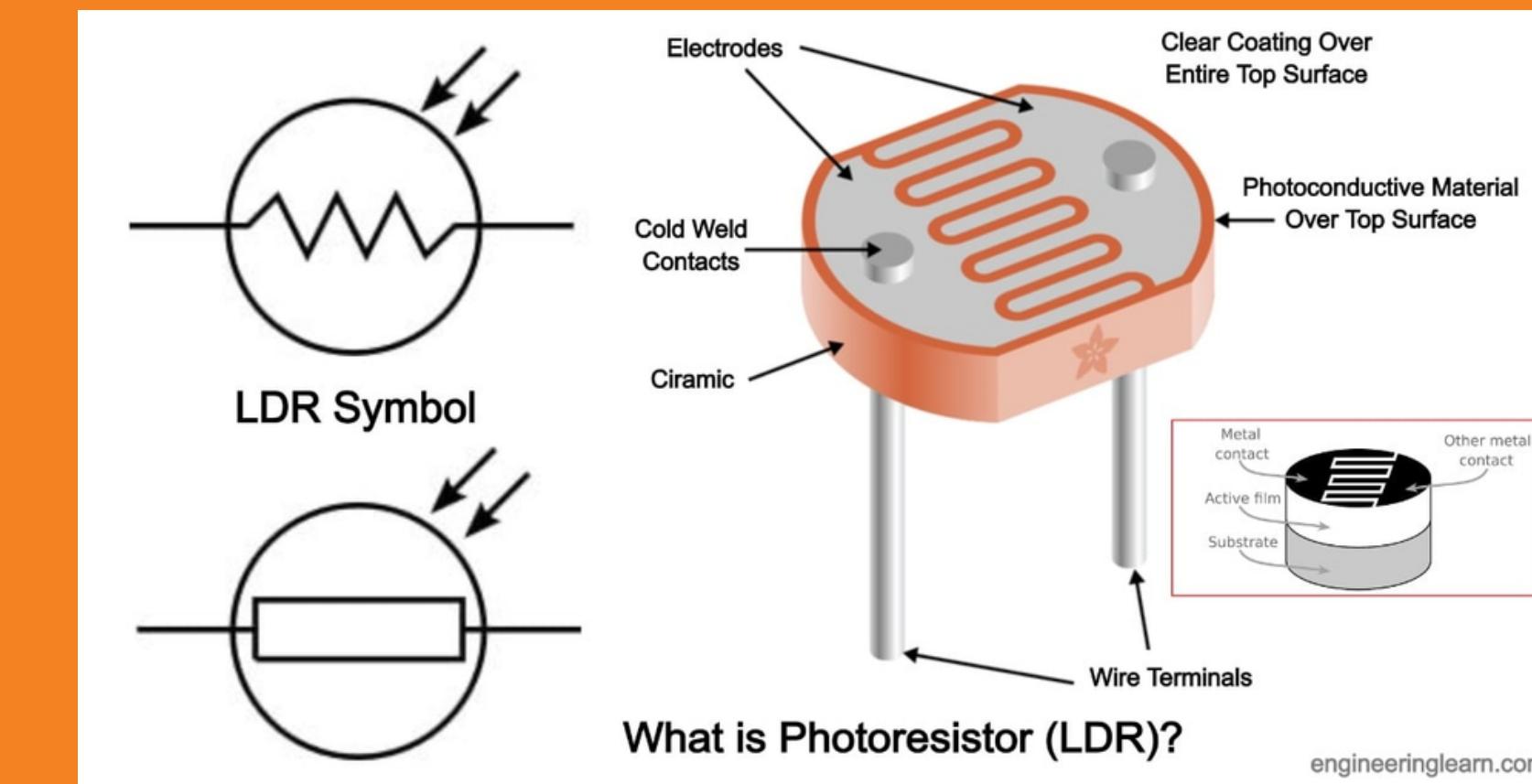
Advanced Circuit

LDR (Light-Dependent Resistor)

หรือ **Photoresistors** คือตัวต้านทานปรับค่าได้ตามความเข้มของแสงที่ตกกระทบบนหน้าสัมผัสของมัน ความต้านทานจะเปลี่ยนตามความเข้มของแสงดังนี้

- แสงน้อย ความต้านทานมาก
- แสงมาก ความต้านทานน้อย

เมื่อไม่มีแสงตกกระทบ ค่าความต้านทานอาจมากถึงหลักล้านกึ่งล้านโอม ($0.5 - 10\text{M}\Omega$ โดยประมาณ) เมื่อมีแสงค่าความต้านทานจะลดลงตามความเข้มแสง ($5 - 10 \text{ K}\Omega$) แต่ถ้าหากแสงเข้มมากๆ เช่น แสงไฟฟ้า ค่าความต้านทานอาจเหลืออยู่เพียงหลักร้อยโอมเท่านั้น



LDR (Light-Dependent Resistor)

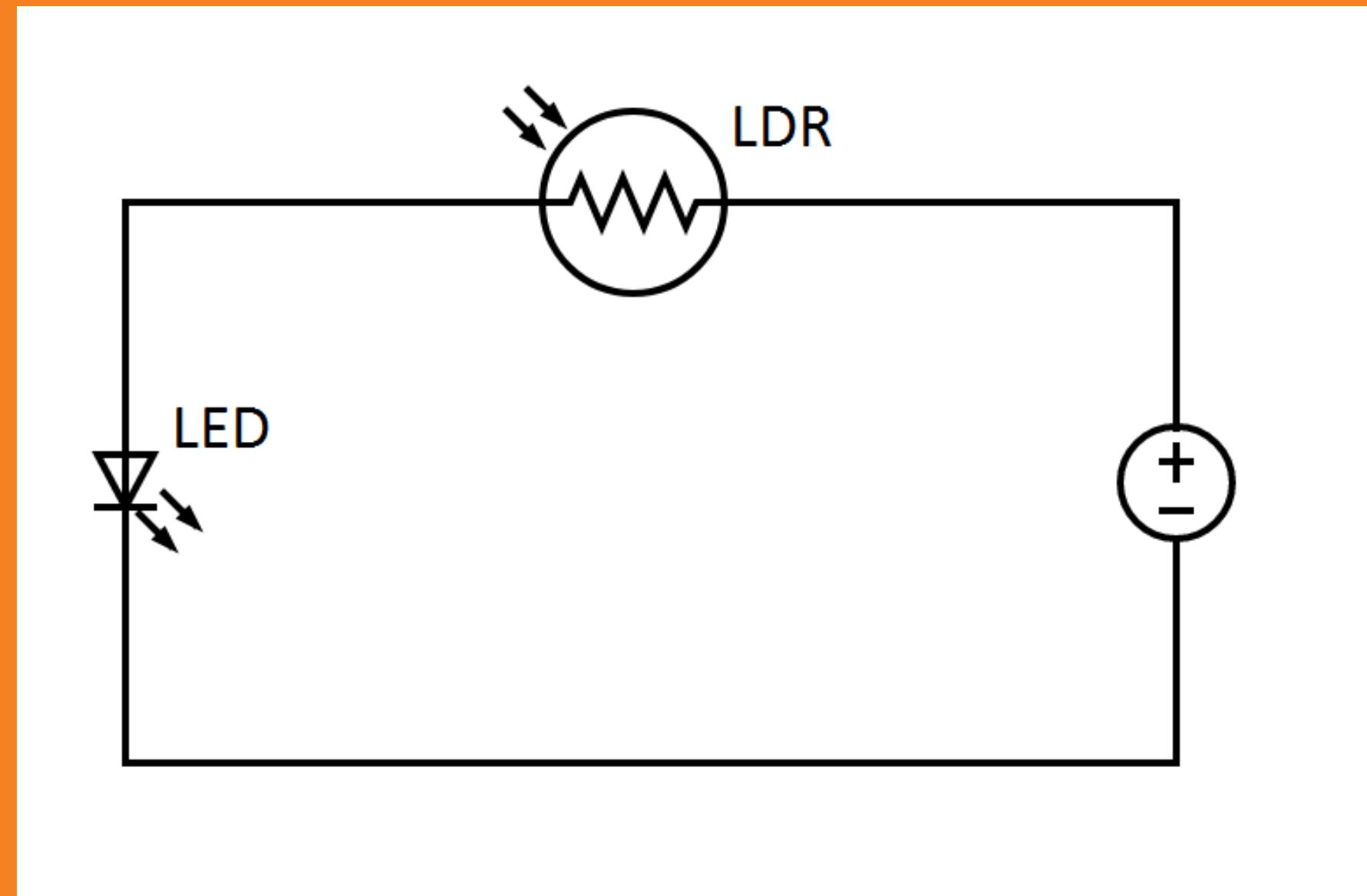
- แสงน้อย ความต้านทานมาก
- แสงมาก ความต้านทานน้อย

ตัวอย่างค่าความต้านทานที่วัดได้ด้วยโอห์มมิเตอร์อย่างคร่าวๆ

ชนิดของแสง	ระยะห่างจากแหล่งกำเนิดแสง	ความต้านทาน
ห้องมีดหึบ	-	$> 1 \text{ M}\Omega$
ห้องสว่าง (เปิดไฟ)	$> 2 \text{ m}$	$< 40 \text{ K}\Omega$
ห้องสว่าง	$< 1 \text{ m}$	$< 20 \text{ K}\Omega$
ห้องสว่าง	$< 10 \text{ cm}$	$< 1 \text{ K}\Omega$
แสงแฟลช	$< 10 \text{ cm}$	$< 100 \Omega$
แสงอาทิตย์ยามบ่าย	-	$< 20 \Omega$

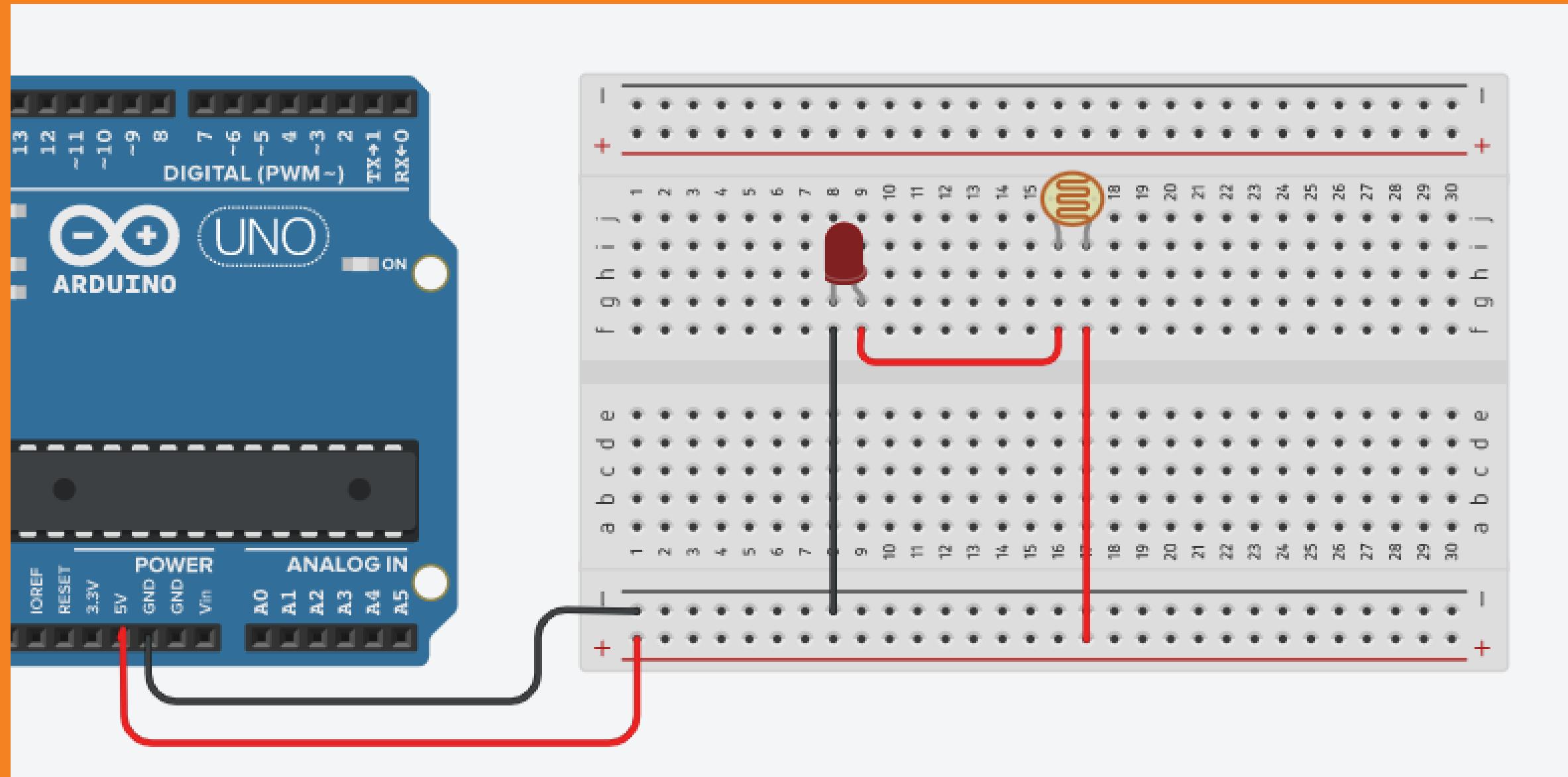
LDR (Light-Dependent Resistor)

เนื่องจาก LDR เป็นตัวต้านทาน เราจึงสามารถใช้มันเป็นตัวต้านทานได้เลย ดังรูป



LDR (Light-Dependent Resistor)

ต่อตามวงจรด้านบน ได้ดังนี้



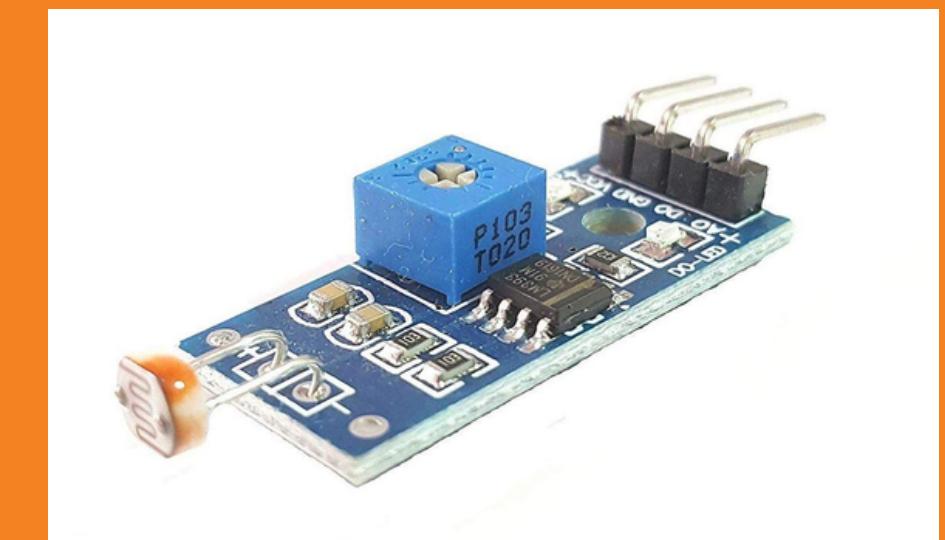
กดตรง LDR บ้องๆสามารถปรับค่าของมัน ดูการเปลี่ยนแปลงได้

LDR (Light-Dependent Resistor)

จากการทดลอง น้องๆจะได้รู้ว่า ถ้าหากแสงน้อยจะทำให้ ความต้านทานมาก
แสงมาก ความต้านทานจะต่ำ

แล้วเราจะรู้ได้ไขว่า ค่าความเข้มแสงมีค่าเท่าใด จะวัดยังงัย?

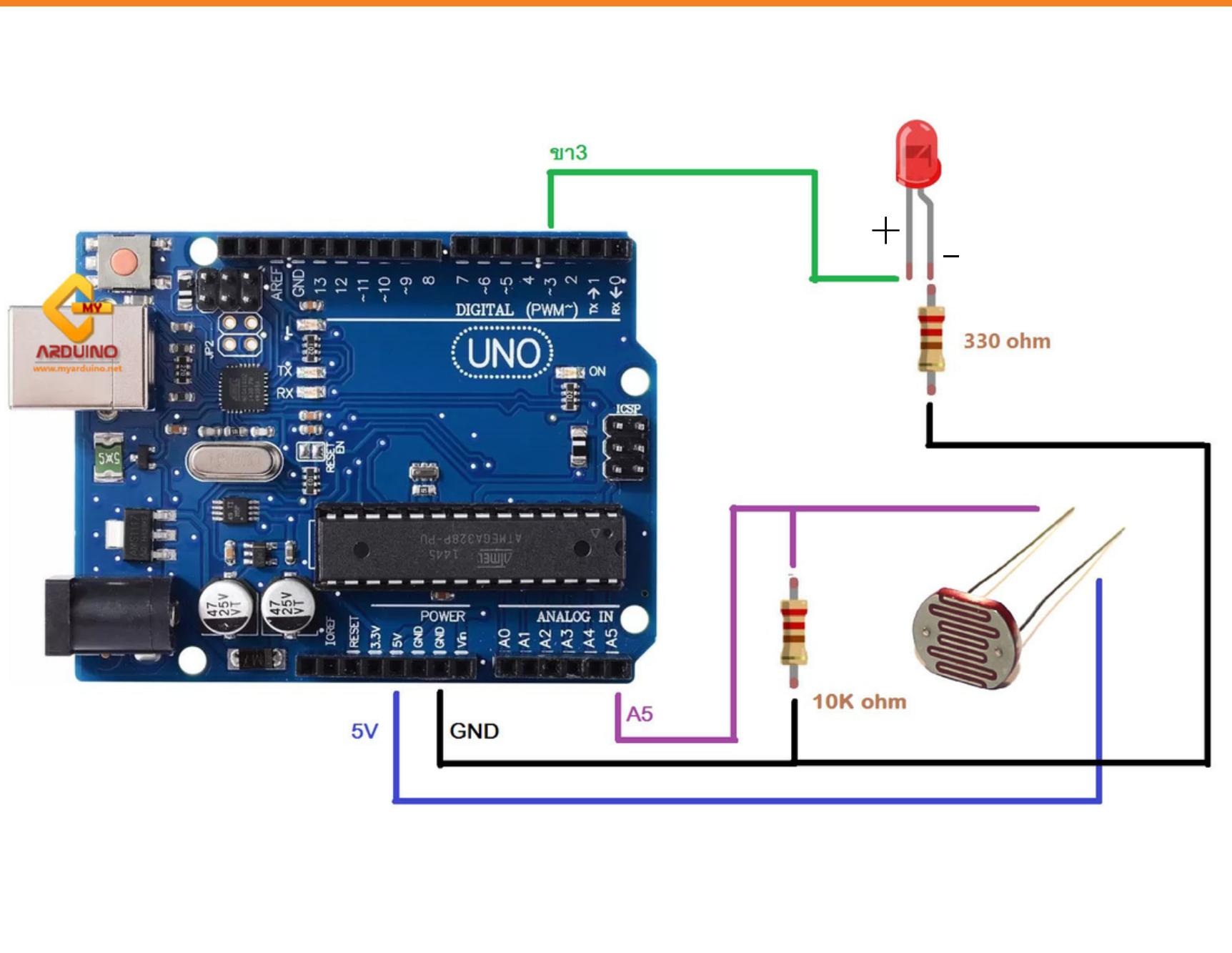
มนุษย์จึงสร้างโมดูลสำหรับขึ้นมา
เพื่อให้สะดวกต่อการใช้งาน
เสียบปุ๊ป เขียนๆ ใช้ได้เลย



เราจะมาสร้างโมดูลนี้ด้วยกันด้านล่างครับ

Activity 4

ให้น้องๆ ต่อวงจรตามรูป ลงใน Photo Board ให้ถูกต้อง ตามภาพ

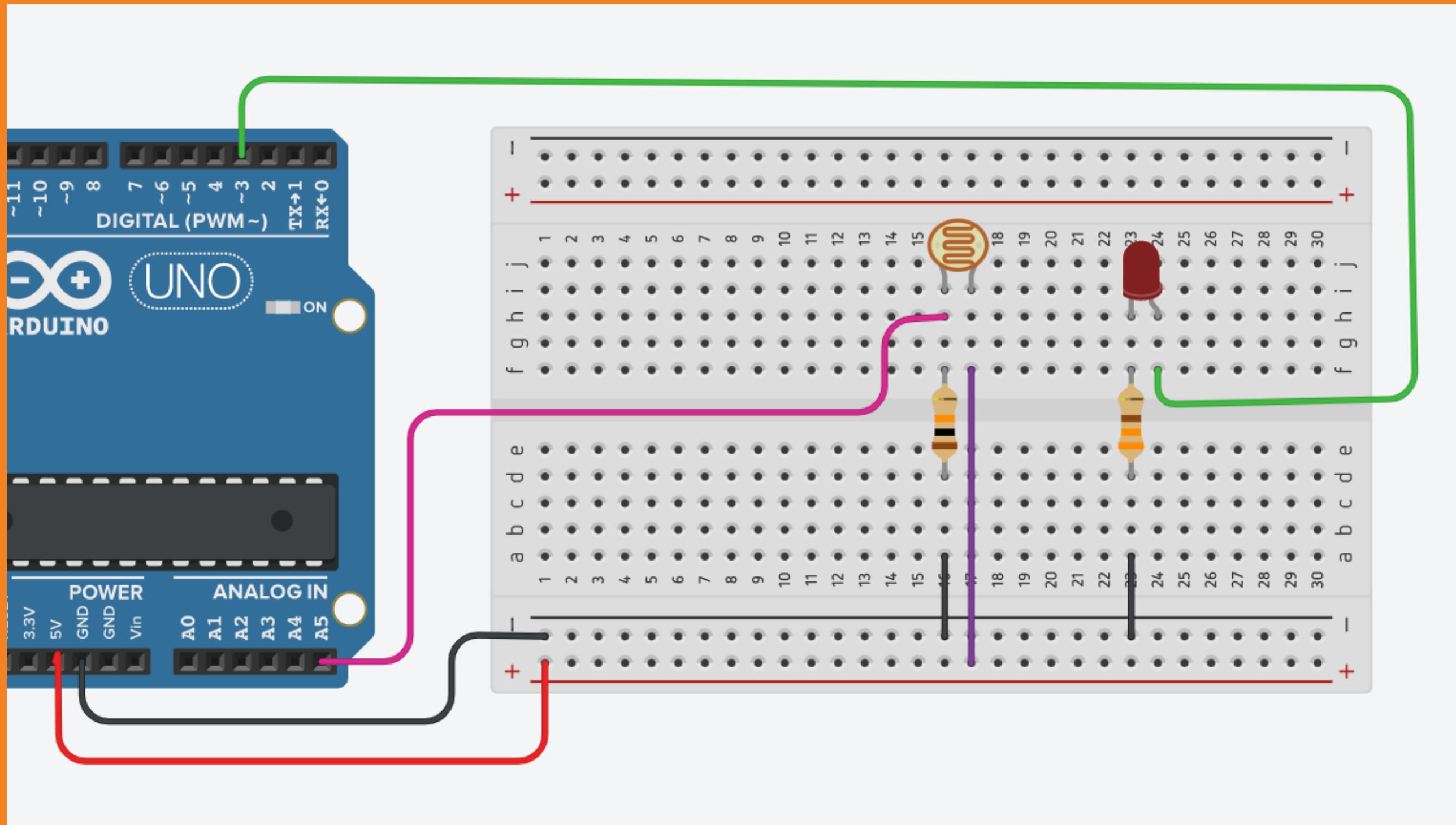


และ เขียนโปรแกรมตามด้านล่าง และลองรันโปรแกรม
ให้ Serial Monitor แสดงผลค่าจาก LDR ออกมายังหน้าจอ

```
1 int ledPin = 3;
2 int analogPin = 5; // ประกาศตัวแปร ให้ analogPin แทนขา analog ขาที่ 5
3 int val = 0;
4 void setup() {
5     pinMode(ledPin, OUTPUT); // sets the pin as output
6     Serial.begin(9600);
7 }
8
9 void loop() {
10    val = analogRead(analogPin); // อ่านค่าสัญญาณ analog ขา 5 ที่ต่อ กับ LDR
11    Serial.print("val = "); // พิมพ์ข้อมูลความสัมภាន "val = "
12    Serial.println(val); // พิมพ์ค่าของตัวแปร val
13    if (val < 100) { // ค่า 100 สามารถกำหนดปรับได้ตามค่าแสงในห้องต่างๆ
14        digitalWrite(ledPin, HIGH); // สั่งให้ LED ติดสว่าง
15    }
16    else {
17        digitalWrite(ledPin, LOW); // สั่งให้ LED ดับ
18    }
19    delay(100);
20 }
```

Activity 4

ເລັກຕົວອ່າງວັຈ



End Hour 3



THANK YOU!

เจอกัน 17.50

Project Section

Project Section

Group Announcement:

ມີກຶ່ງມາດ 8 ກລຸ່ມ
ກລຸ່ມລະ 2 ດນ

Project Section

Project Title

G1

ป้องกันขโมย

G2

แจ้งเตือนภัย

G3

การเกษตร

G4

แจ้งเตือนภัย

G5

การเกษตร

G6

ป้องกันขโมย

G7

แจ้งเตือนภัย

G8

ป้องกันขโมย

Project Section

แต่ละทีมต้อง . . .

- มีอุปกรณ์ใน Input 1 ตัวอย่างน้อย
- มีอุปกรณ์ใน Output 1 ตัวอย่างน้อย
- ต่อวงจรใน Photo Board เก่านั้น
- สายไฟ รบกวนแยกสีด้วย 5v ต้องสีแดง GND ต้องสีดำ ที่เหลือสีไรก็ได้
- พรีเซนเตอร์ชิวๆ ไม่เครียด อย่าคิดมาก เป็นแฉ แต่ขอให้เห็นภาพ
- ต้องเล่าถึงที่มาและความสำคัญ รวมถึงโชว์การทดลองอุปกรณ์ด้วย



Presentation

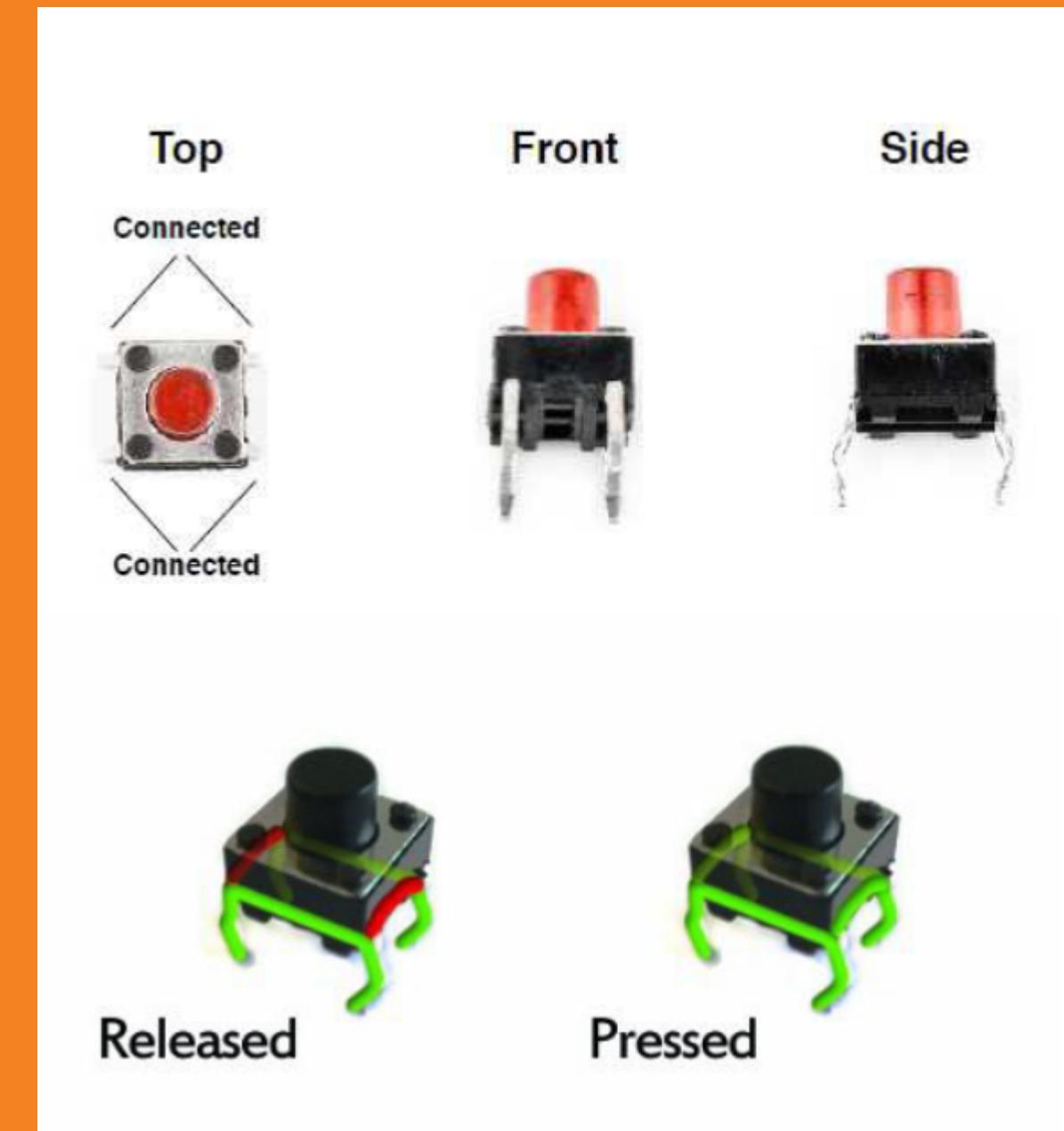


THANK YOU!



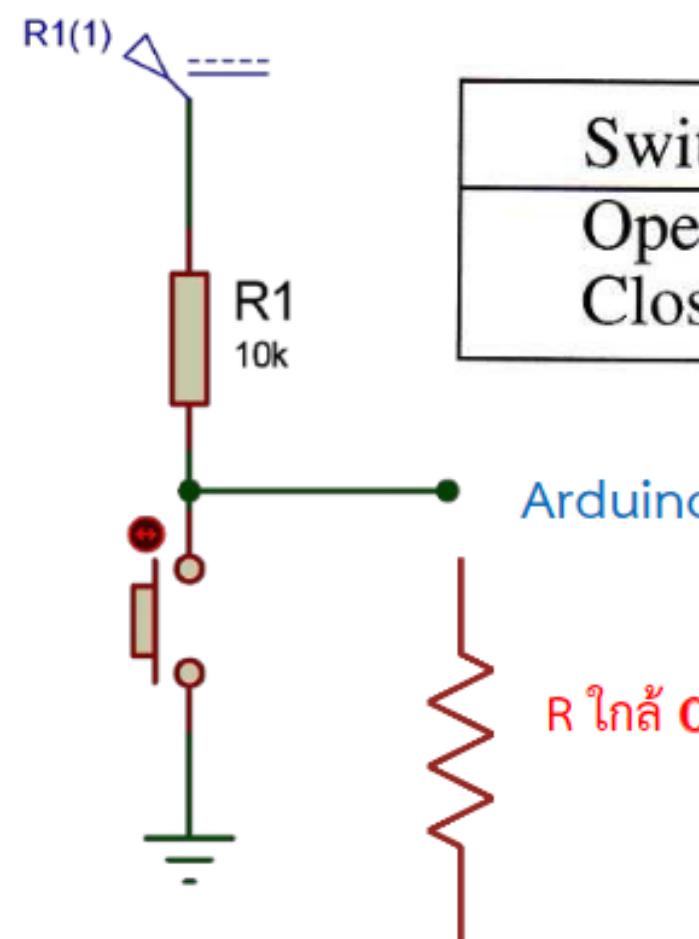
Special Gift: P' New Portfolio

Switch and Pullup Pulldown

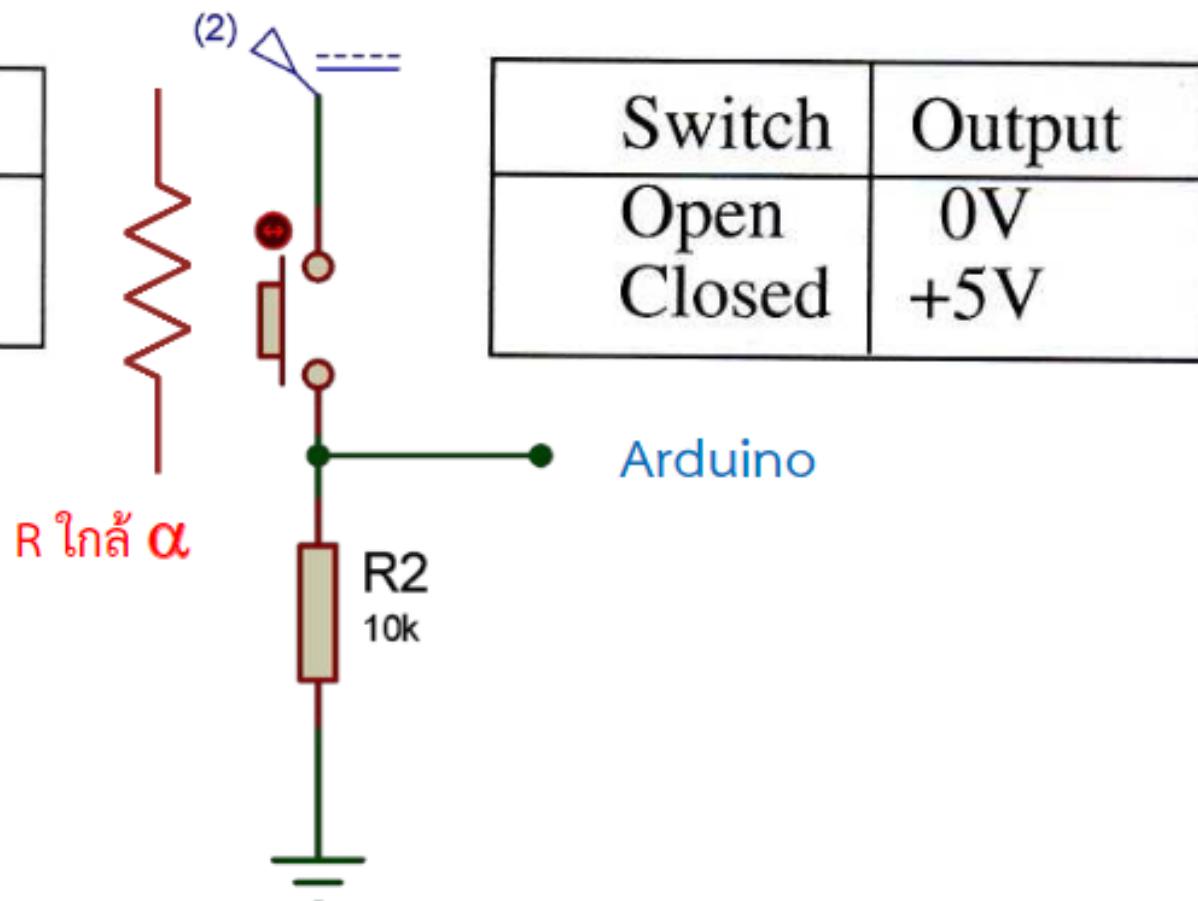


Pullup Pulldown

Pull Up



Pull Down



$$R_{\text{total}} = R_1 \parallel R_2 = \frac{R_1 \cdot R_2}{R_1 + R_2}$$

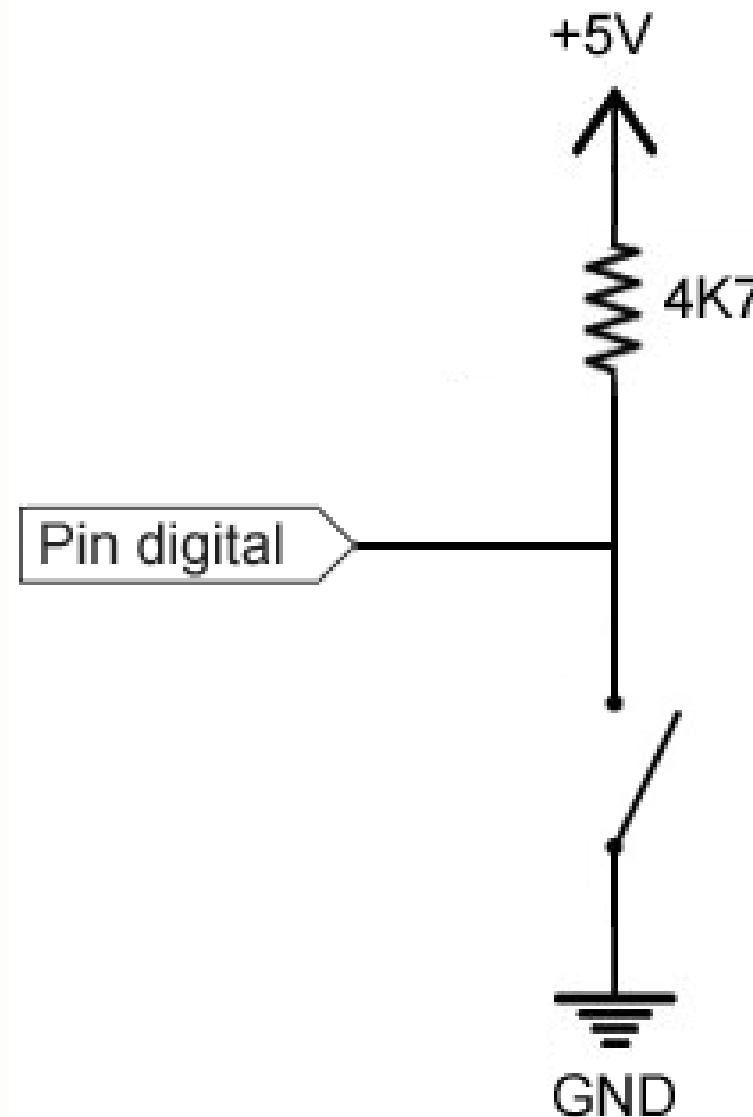
ทริคการจำ

Pull up = กดแล้วดับ
หรือ กดแล้วเป็น LOW

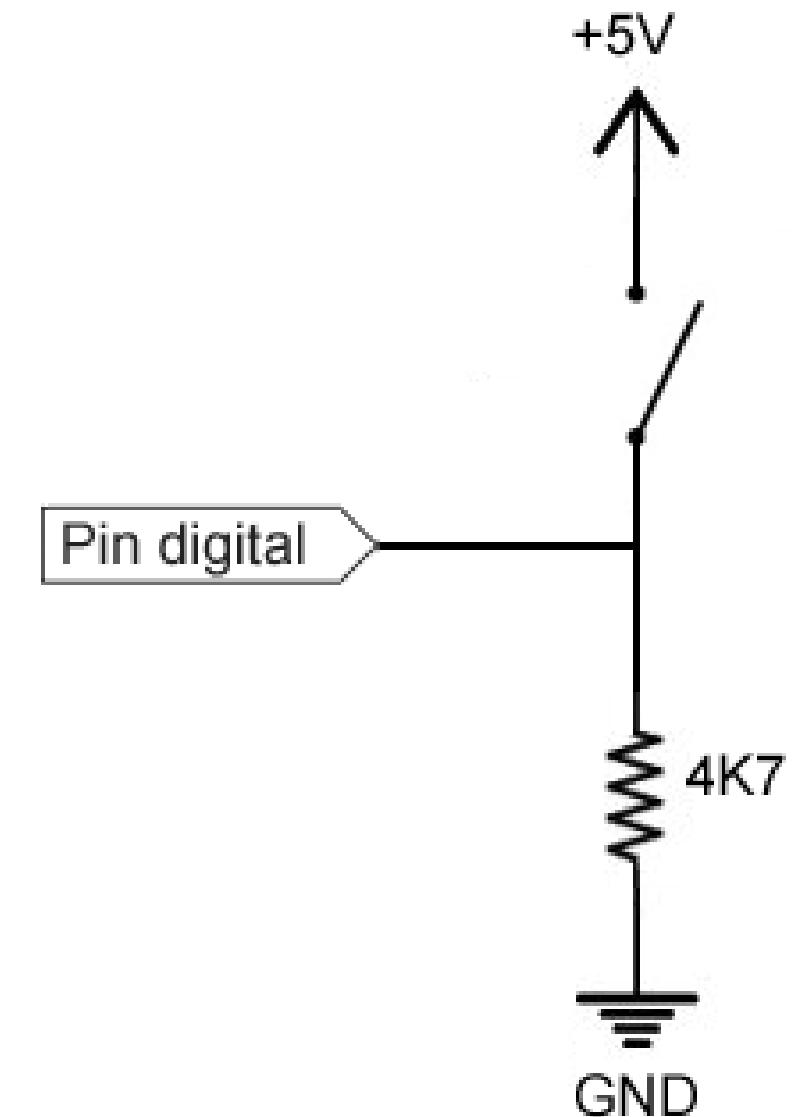
Pull down = กดแล้วติด
หรือ กดแล้วเป็น HIGH

Pullup Pulldown

RESISTENCIA PULL UP

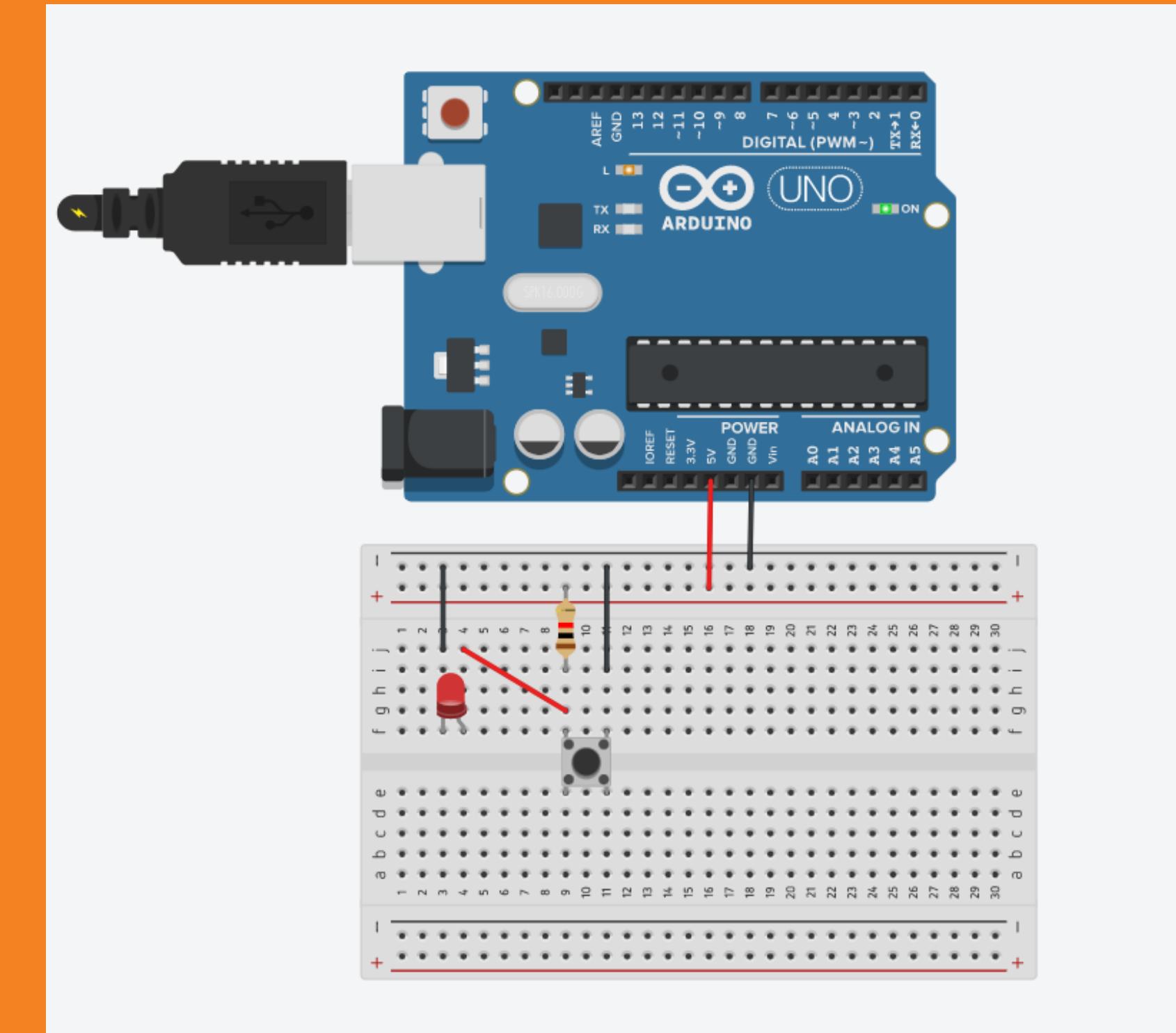
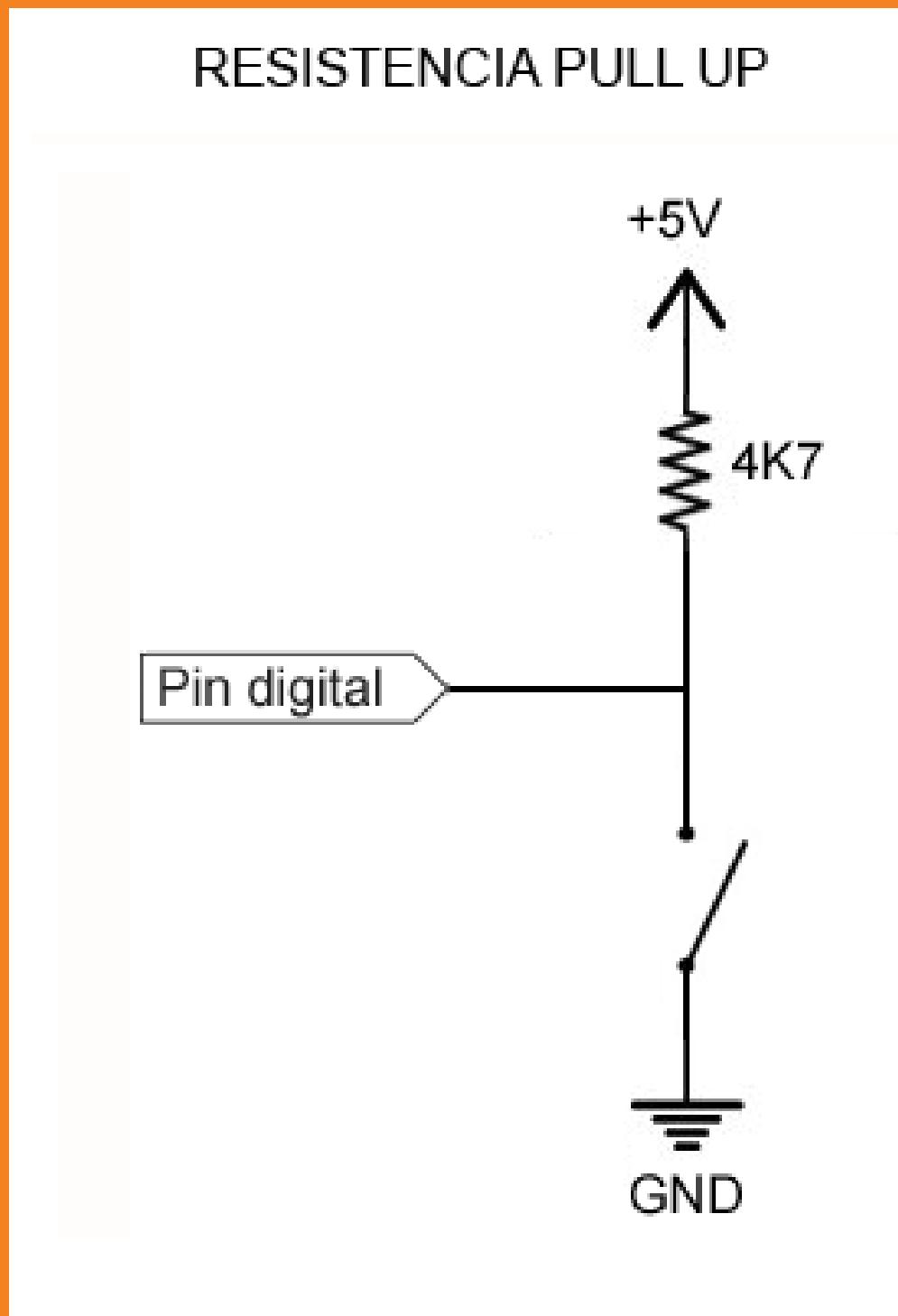


RESISTENCIA PULL DOWN



การต่อแบบไม่ต้องเขียนโปรแกรม (ฝึกให้เข้าใจการทำงาน)

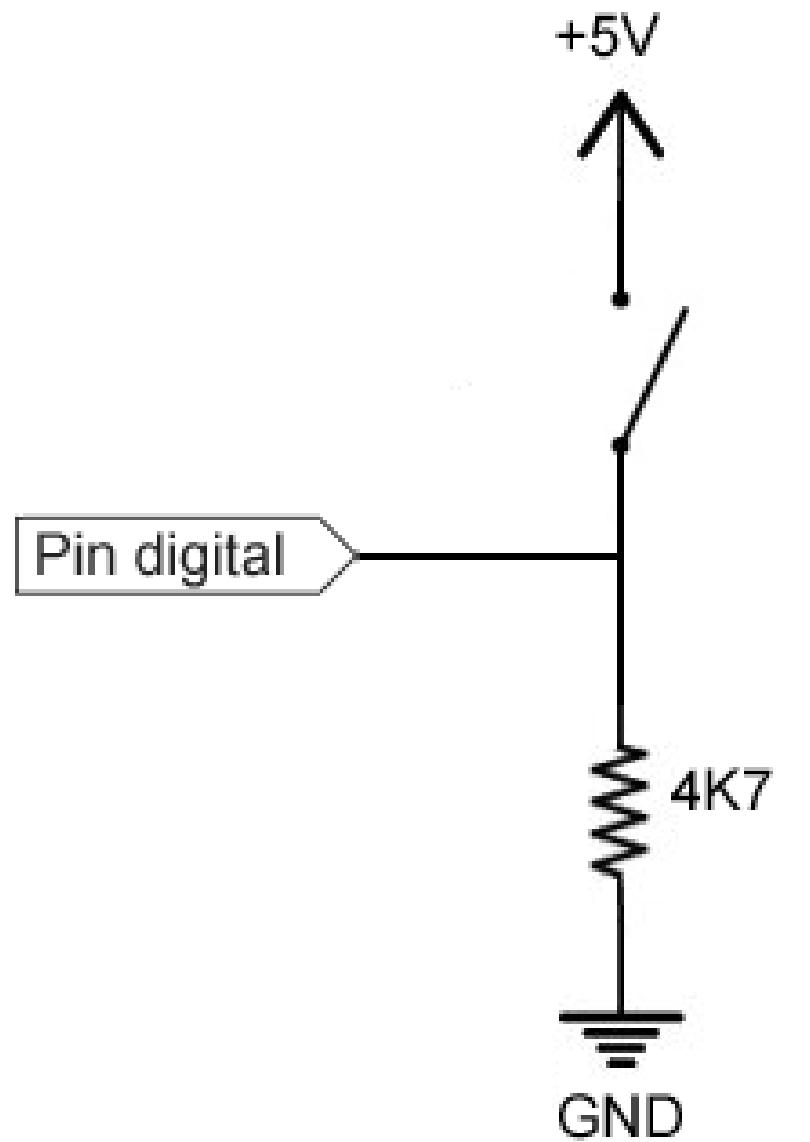
Pull Up



Activity 3

ให้น้องๆต่อวงจร Pull Down โดยไม่ต้องเขียนโปรแกรม
(กดให้ติด ปล่อยให้ดับ)

RESISTENCIA PULL DOWN



Digital Input

SETUP PINMODE

Syntax:

pinMode(pin, mode)

เปลี่ยนเป็น Input เพราะต้องรับค่าเข้ามา

Parameter:

pin: the number of the pin whose mode you wish to set

mode: INPUT, OUTPUT or INPUT_PULLUP.

Digital Input

DIGITAL INPUT PROGRAMMING (ON-OFF)

Syntax:

`digitalRead(pin)` อ่านค่าจากการกด จะอ่านเป็น 0 และ 1
หรือ LOW และ HIGH

Parameter:

`pin`: the number of the pin whose mode you wish to set

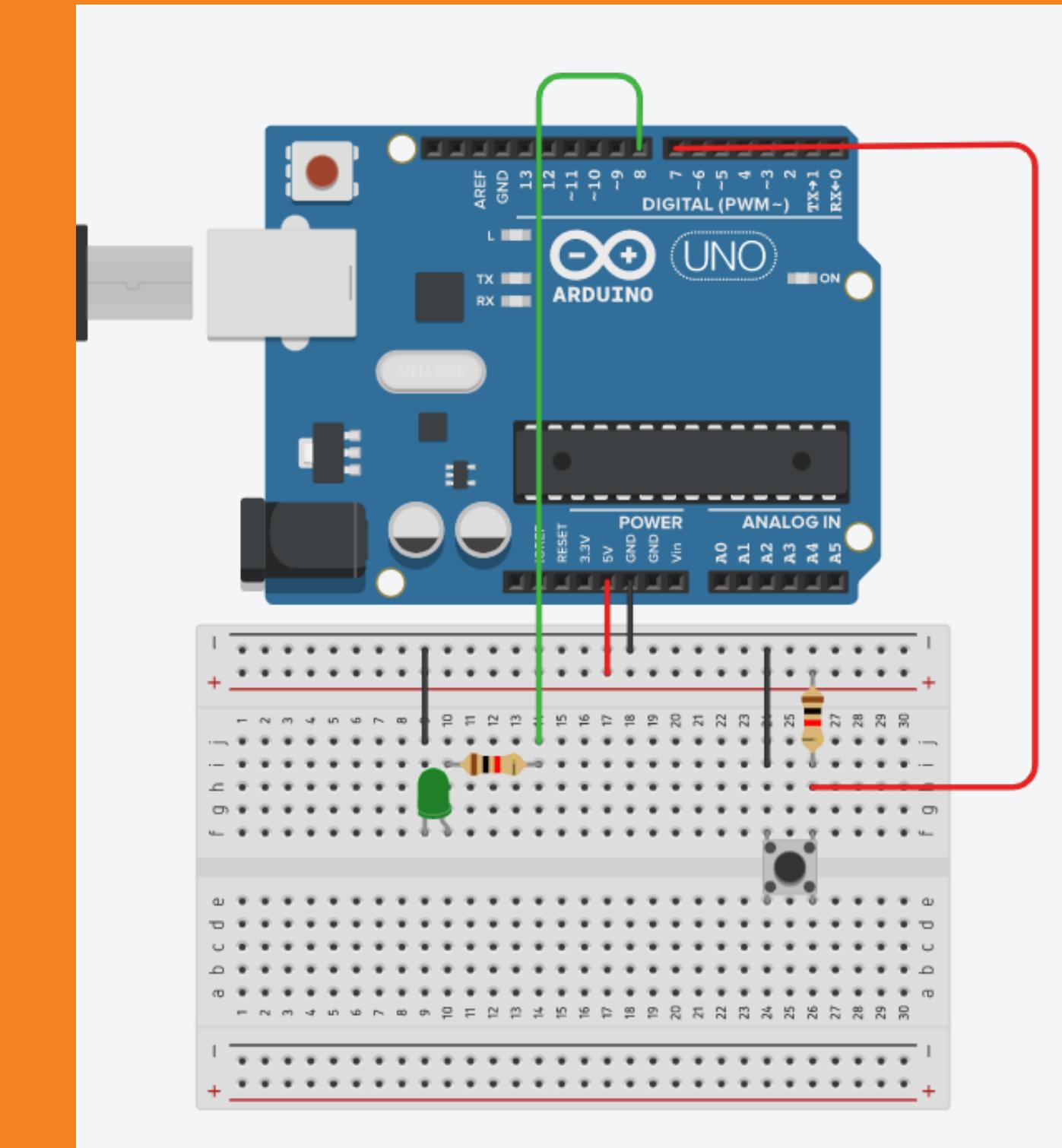
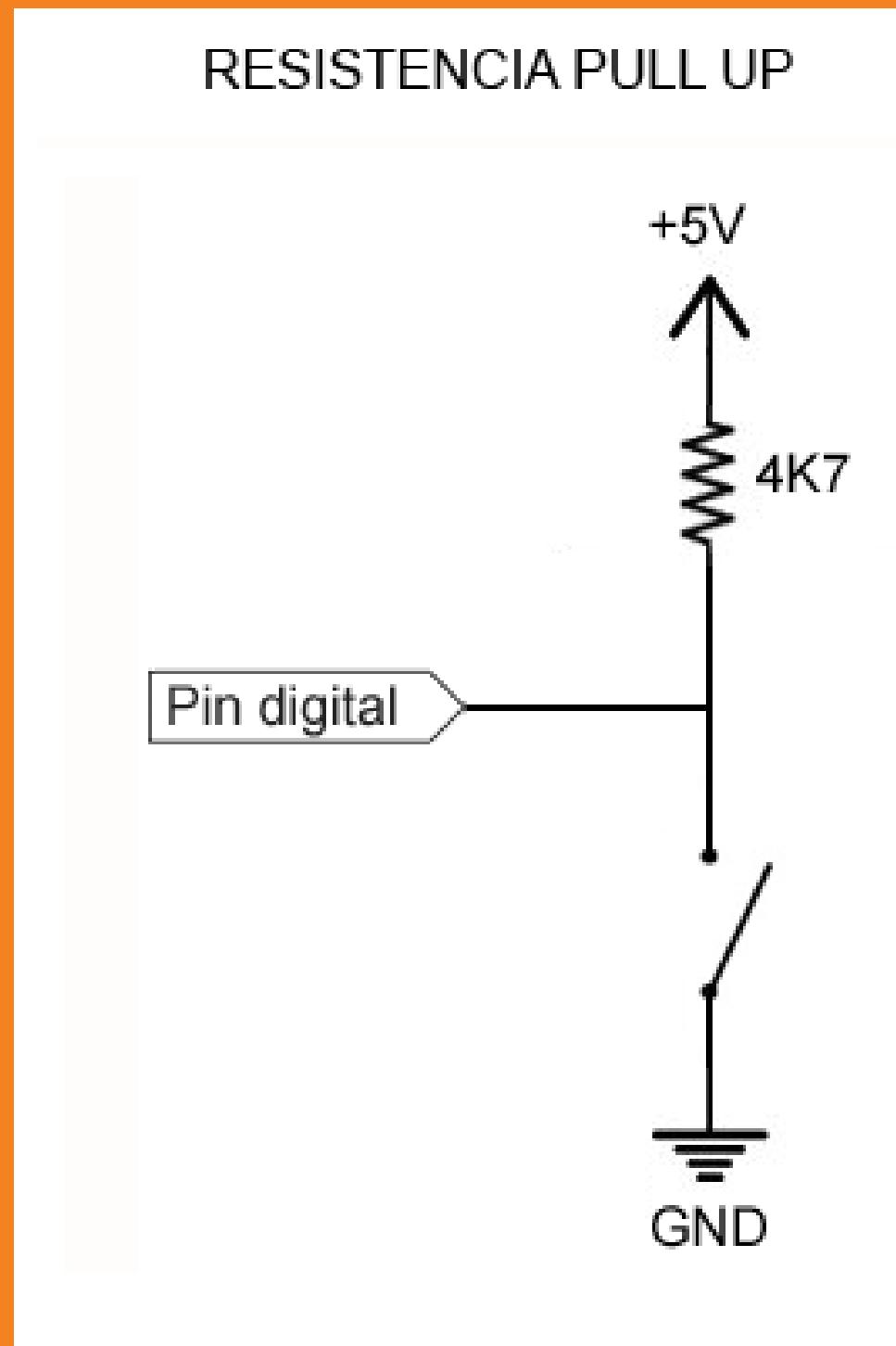
Return:

HIGH: when the logic is HIGH

LOW: when the logic is LOW

ตัวอย่างการต่อเมื่อต้องการรับ Input จาก คำสั่ง digitalRead()

Pull Up



Digital Input

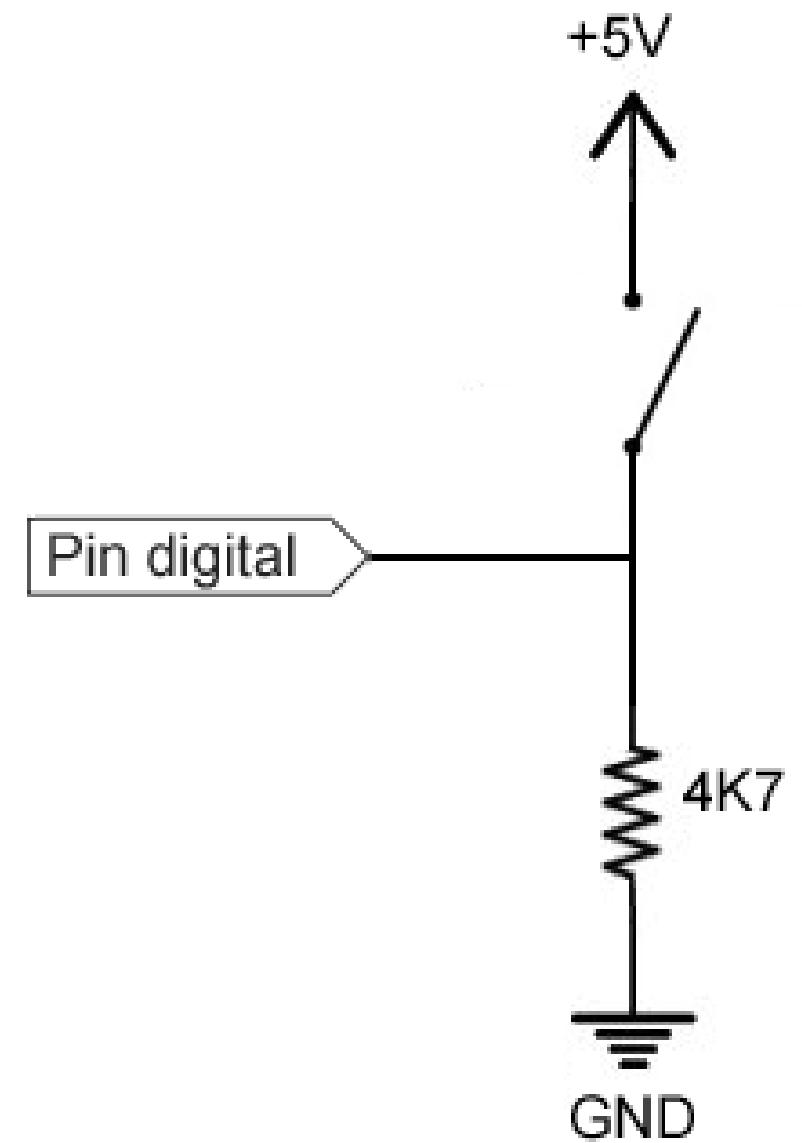
```
1 #define LEDpin 8
2 #define button 7
3
4 void setup() {
5     pinMode(LEDpin, OUTPUT);
6     pinMode(button, INPUT);
7 }
8 void loop() {
9
10    if(digitalRead(button) == HIGH) {
11        digitalWrite(LEDpin, HIGH);
12    }
13
14    else{
15        digitalWrite(LEDpin, LOW);
16    }
17 }
```

Activity 4

จากตัวอย่างเมื่อกี้ ให้น้องๆเปลี่ยนเป็น Pull Down
และลองรันโปรแกรม จะสังเกตุผลลัพธ์

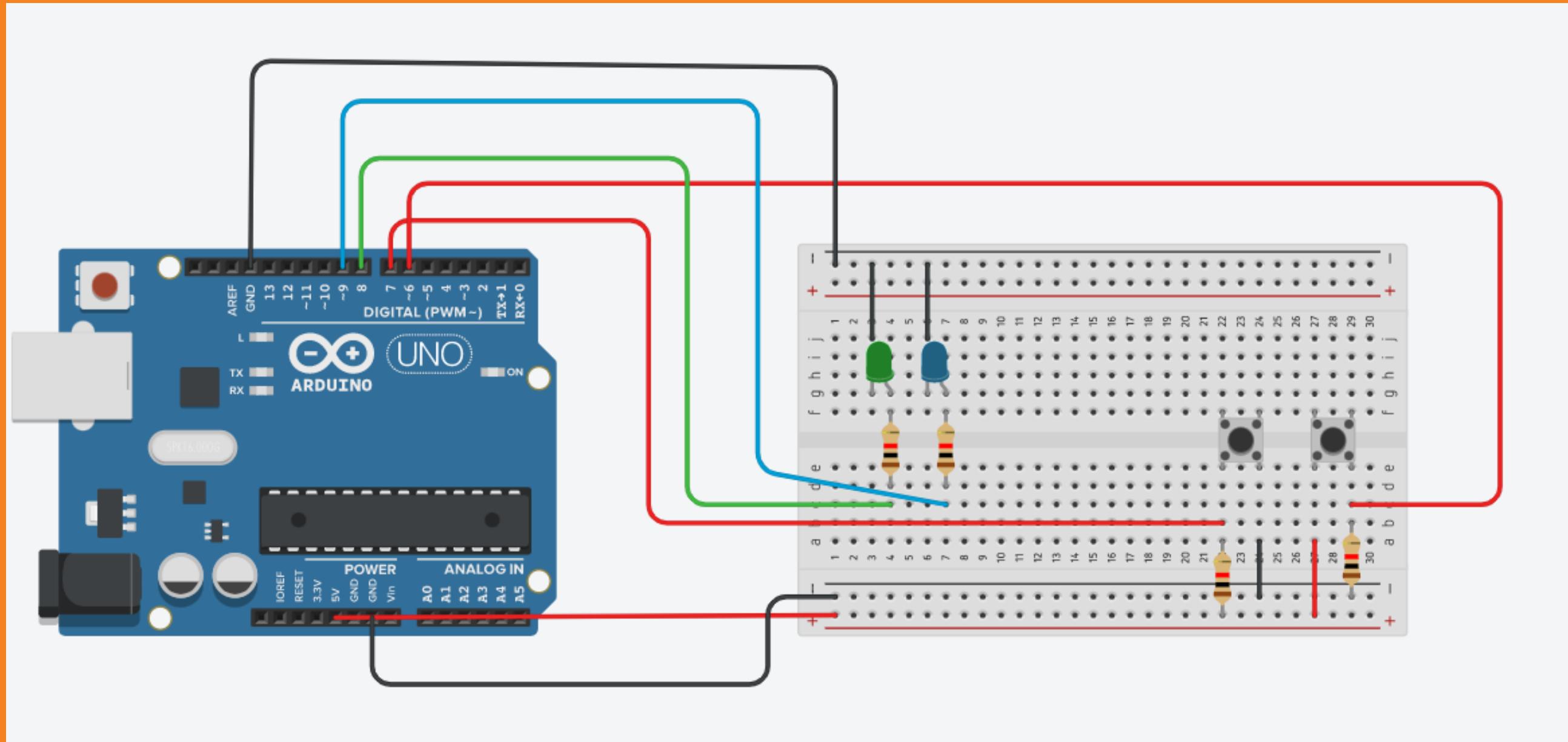
5 นาที

RESISTENCIA PULL DOWN



Activity 5

ให้หน่องๆ ต่อทิ้งสองแบบ และเขียนโปรแกรมควบคุมทิ้ง 2 ปุ่ม



คำใบ้

1. define อีกปุ่มและอีกหลอดขึ้นมา
2. ก็อปโค๊ดเก่า และว่าง
3. แกนที่ตัวแปร

สลับสถานะของ LED ด้วยปุ่มกด

สิ่งที่เราต้องทำคือสลับสถานะของ LED เมื่อคุณกด + ปล่อยปุ่ม
ดังนั้น ในครั้งแรกที่คุณปล่อยปุ่ม ไฟ LED จะเปิดขึ้น ครั้งที่สองก็จะดับลง เป็นต้น
ภาษาชาวบ้านคือ กดแล้วติดจนกว่าจะมีการกดอีกทีถึงดับ



สลับสถานะของ LED ด้วยการปุ่มกด (on/off)

```
#define LED_PIN 8
#define BUTTON_PIN 7
byte lastButtonState = LOW;
byte ledState = LOW;

void setup() {
    pinMode(LED_PIN, OUTPUT);
    pinMode(BUTTON_PIN, INPUT);
}

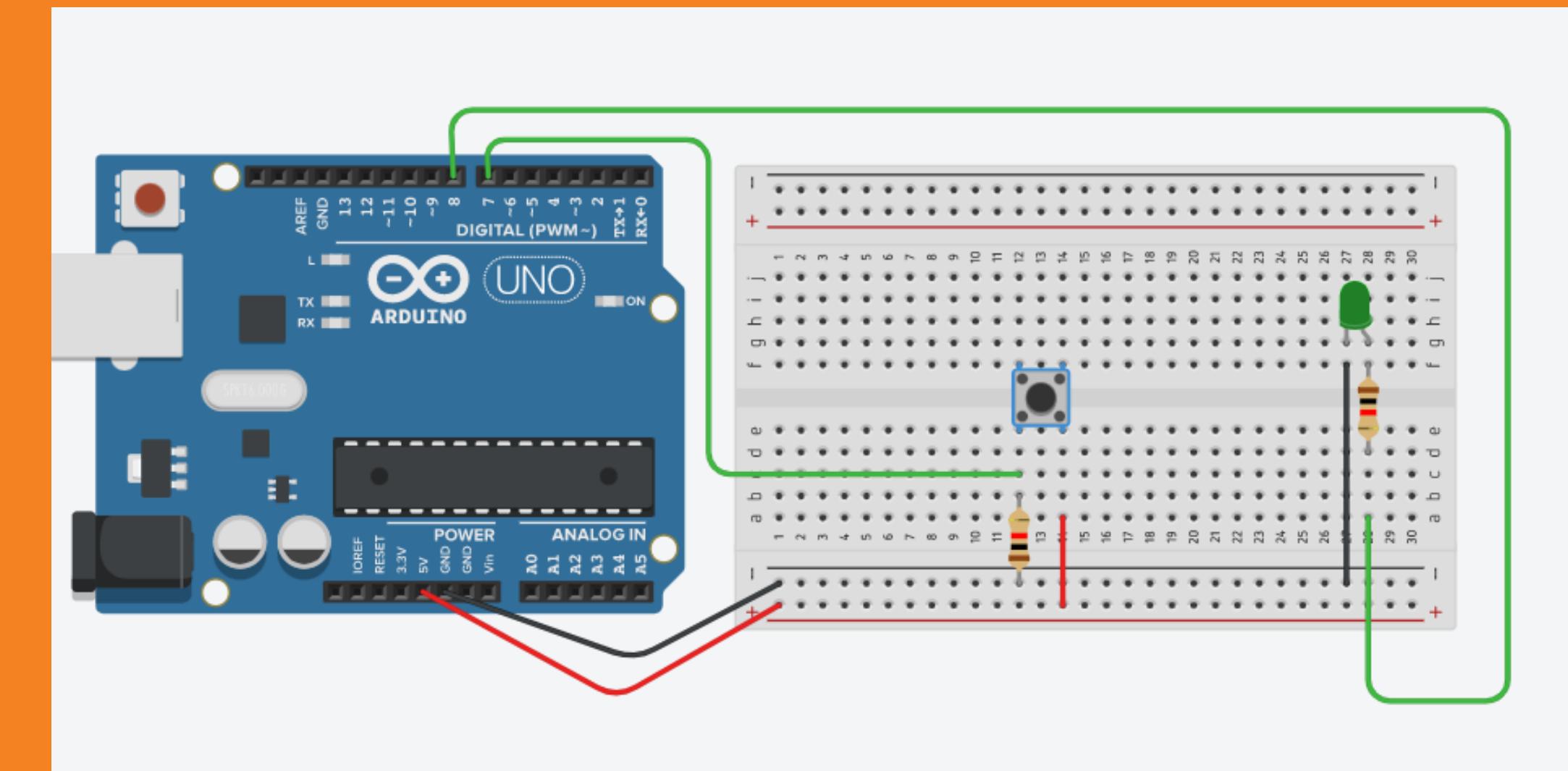
void loop() {
    byte buttonState = digitalRead(BUTTON_PIN);

    if (buttonState != lastButtonState) {
        lastButtonState = buttonState;

        if (buttonState == HIGH) {
            if(ledState == HIGH){
                ledState = LOW;
                digitalWrite(LED_PIN, ledState);

            }else{
                ledState = HIGH;
                digitalWrite(LED_PIN, ledState);
            }
        }
    }
}
```

Pull Down



Activity 6

เขียนโปรแกรมดังนี้

1. ไฟสองหลอด Pull up, Pull Down
 2. ปุ่มที่ 1 ถ้ากดครึ่งแรกไฟหลอดที่ 1 ต้องดับค้างไว้
 3. ปุ่มที่ 2 ถ้ากดครึ่งแรกไฟหลอดที่ 2 ต้องติดค้างไว้

