หน่วยการเรียนรู้ที่ 2 การสร้างปุ่มกดให้กับตู้จำหน่ายเครื่องดื่ม

อุปกรณ์ Push Button

Digital Input

บอร์ด Arduino มี Digital I/O PINs สำหรับใช้งาน ซึ่งคำว่า I/O หมายถึง Input และ Output นั่นหมายความว่า นอกจาก Pins เหล่านี้จะสามารถเป็น Output เพื่อไปควบคุมอุปกรณ์ต่าง ๆ ได้แล้ว มันยังสามารถใช้เป็นตัว Input หรือตัวรับสัญญาณทางไฟฟ้าต่าง ๆ เพื่อใช้ควบคุมการทำงานของวงจรได้อีกด้วย

Digital คืออะไร

ทุกคนคงรู้จักคำว่า Digital ซึ่งในที่นี้จะหมายถึงสัญญาณทางไฟฟ้าประเภทหนึ่งที่มีค่าการทำงาน 2 ค่า คือ 1 และ 0 ซึ่ง 1 หมายถึง on (HIGH) และ 0 หมายถึง off (LOW) สัญญาณดิจิตอลนั้นสามารถนำไปใช้ประโยชน์ได้มากมายและมีความแม่นยำ

Digital Input

การที่เราใช้ Digital Pins ของ Arduino เป็นตัวรับค่าสัญญาณ คือการที่เราให้ Pins นั้น ๆ เป็นตัว Input โดยค่าที่ Arduino อ่านได้จะมีอยู่ 2 ค่า คือ HIGH และ LOW ขึ้นอยู่กับ Pin นั้น มีสัญญาณหรือกระแสไฟฟ้าเข้ามาหรือไม่

Pull-up , Pull-Down

เป็นอีกหนึ่งสิ่งที่จำเป็นต้องรู้ เมื่อเรานำสวิซท์หรือปุ่ม button ต่อเข้ากับ Input Pin ของ Arduino เราจำเป็นต้องมีตัวต้านทานด้วย เพราะ Input Pin นั้น ๆ จะอ่านค่าได้ไม่แน่นอนว่าค่าที่รับมาเป็น HIGH หรือ LOW กันแน่ เนื่องจากหลายๆครั้งที่แรงดันจะอยู่ครึ่งๆกลางๆระหว่าง HIGH กับ LOW เราจึงต้องมีการใส่ตัวต้านทาน Pull-up หรือ Down ลงไป เพื่อกำหนดสถานะให้แน่ชัดไปเลย ว่าจะสูงหรือต่ำกันแน่

คำว่า Pull จึงหมายถึงการดึงหรือการบังคับให้เป็นสูงหรือต่ำ

• Pull-Up คือการบังคับให้ Input มีสถานะ HIGH ตลอดเวลา จนกว่าจะกดปุ่ม เมื่อกดปุ่ม จะเป็น LOW เมื่อไม่กดจะเป็น HIGH หรือเรียกว่า Active Low

• Pull-Down คือบังคับให้ Input เป็น LOW ตลอดเวลา จนกว่าจะกดปุ่ม เมื่อกดปุ่มจะเป็น HIGH เมื่อไม่กดจะเป็น LOW หรือเรียกว่า Active High

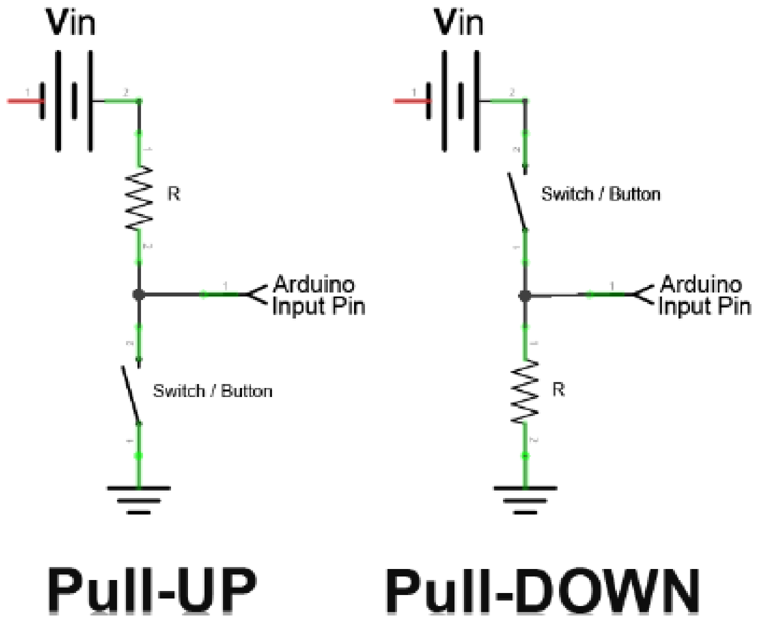
วงจร Pull-Up

จะต่อตัวต้านทานค่าหนึ่ง เรียกว่า Pull-Up Resistor ไว้ที่ไฟเลี้ยง Vin เพื่อเป็นการรักษาระดับแรงดันไฟฟ้าให้เป็น HIGH ตลอด ,Arduino Input Pin จึงอ่านค่าได้เป็น HIGH ตลอด แต่เมื่อกดปุ่ม Button , กระแสไฟฟ้าจะไหลลง Ground ทันที Input Pin จึงอ่านค่าได้เป็น LOW

วงจร Pull-Down

จะต่อตัวต้านทาน หรือ Pull-Down Resistor ไว้ที่ Ground แทน ส่วนปุ่ม Button จะต่อเข้ากับ Vin เมื่อไม่กดจะมีสถานะเป็น LOW แต่เมื่อกดจะเป็น HIGH เพราะกระแสไฟฟ้าจะเลือกไหลเข้าหา Arduino Pin ที่มีความต้านทานน้อยกว่าแทน

จะเห็นได้ว่าวงจรทั้ง 2 มีสถานะการทำงานที่ตรงกันข้าม ขึ้นอยู่กับจะเลือกใช้แบบไหน



การต่อวงจร Pull-Up / Pull-Down

ตัวต้านทานที่นำมาใช้เป็นตัว Pull-Up / Down Resistor นิยมใช้ค่า 5K - 20K Ω

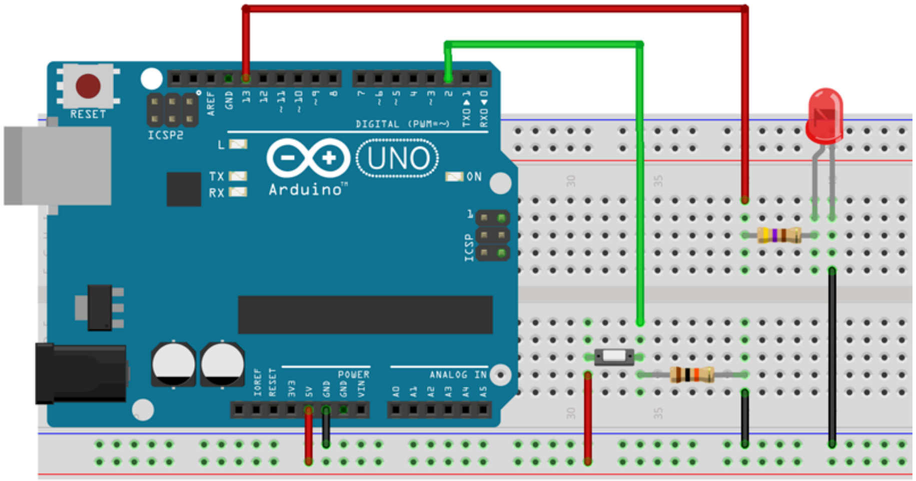
ตัวอย่าง การต่อ Button มาควบคุมไฟ LED

มาลองทำวงจร Arduino ง่ายๆแต่ได้ความรู้ คือวงจรที่มีทั้ง Input และ Output แบบ Digital โดยเราจะทำวงจรที่ใช้ Input Pin ของ Arduino มารับค่าการกดปุ่ม แล้วนำค่านั้นไปควบคุม LED อีกทีโดยใช้ Output Pin โดยเราจะใช้วงจรแบบ **Pull-Down** เพื่อกำหนดให้ LED ติดเมื่อกดปุ่มเท่านั้น

การต่อวงจร

อุปกรณ์ที่ใช้ มีดังนี้

- Arduino UNO R3
- ปุ่มกดติดปล่อยดับ
- หลอด LED
- ตัวต้านทาน 10K Ω
- ตัวต้านทาน 470 Ω
- สายไฟต่าง ๆ และ Protoboard



วงจรนี้ เป็นวงจรแบบ Pull-Down โดยต่อปุ่ม Button เข้ากับไฟ 5 V และมีตัวต้านทานค่าตั้งแต่ 5K - 20K Ω ในที่นี้ใช้ 10K Ω ทำหน้าที่เป็น Pull-Down Resistor และใช้ Digital PIN 2 เป็นตัว Input รับค่าการกดปุ่ม ส่วนตัว Output คือ Digital PIN 13 ต่อเข้ากับ LED โดยมีตัวต้านทาน 470 Ω ช่วยจำกัดกระแส ป้องกันกระแสเกิน

เขียน Code และ Upload

```
// Digital Input with Pull-Down Push Button
// www.commandronestore.com

int buttonPin = 2;
int ledPin = 13;
int buttonState = 0; //ตัวแปรสถานะของปุ่ม ตั้งค่าเริ่มต้นให้เป็น 0 ไว้ก่อน

void setup() {
pinMode(ledPin, OUTPUT); //กำหนดให้ Pin 13 เป็น Output
pinMode(buttonPin, INPUT); //กำหนดให้ Pin 2 เป็น Input
}

void loop(){
buttonState = digitalRead(buttonPin);

if (buttonState == HIGH) {
digitalWrite(ledPin, HIGH); //ถ้ากดปุ่ม ไฟจะติด
}
else {
digitalWrite(ledPin, LOW); //ถ้าไม่กดปุ่ม ไฟจะดับ
}
}
```

อุปกรณ์ 4 Pin Push Button (ปุ่มกดแบบ 4 ขา)

การใช้งานปุ่มกดติดปล่อยดับ (Push Button)

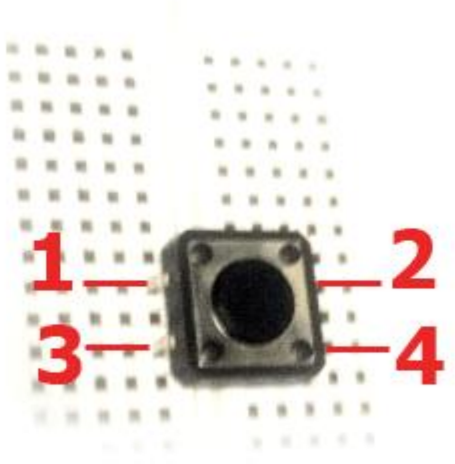
Push button คือ สวิชท์วงจรอิเล็กทรอนิกส์ชนิดหนึ่ง มีหน้าที่ควบคุมการเปิด และ ปิด ของวงจรส่วนนั้น ๆ โดยทั่วไปอาจมี 2 ขา หรือ 4 ขา โดยปุ่มกดติดปล่อยดับนั้น เมื่อทำการกดจะเป็นการปิดวงจร ทำให้กระแสไฟฟ้าสามารถไหลผ่านวงจรได้ เมื่อไม่ได้กด จะทำให้วงจรเปิด กระแสไฟฟ้าจะไม่สามารถไหลผ่านวงจรได้

4 Pins Push Button (ปุ่มกดติดปล่อยดับ 4 ขา)

ปุ่มกดติดปล่อยดับ 4 ขา เป็นที่นิยมกันอย่างมาก การต่อวงจรอาจดูเหมือนยุ่งยาก แต่อันที่จริงแล้ว สามารถต่อได้ง่ายดายมากถ้าเข้าใจหลักการทำงานของมัน

หลักการทำงานของ 4 Pins Push Button

การทำงานของมันจะแยกเป็น 2 ส่วน คือ ตอนที่ยังไม่กด (Not Pressed) และ ตอนที่กด (Pressed) ซึ่งเมื่อเราลองจัดวางปุ่ม Button นี้ ดังรูป โดยให้ด้านหน้าและหลังของปุ่ม ไม่มีขา ส่วนด้านซ้ายเป็นขา 1 กับ 3 และด้านขวาเป็นขา 2 กับ 4

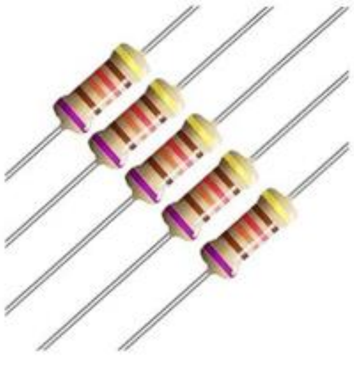


เมื่อวงจรเปิด (Not Pressed) : ขาที่ 1 จะเชื่อมอยู่กับขาที่ 2 / ขาที่ 3 จะเชื่อมอยู่กับขาที่ 4

เมื่อวงจรปิด (Pressed) : ขาที่ 1 จะเชื่อมอยู่กับขาที่ 3 / ขาที่ 2 จะเชื่อมอยู่กับขาที่ 4

หรือพูดง่ายๆคือ เมื่อเรายังไม่ได้กดปุ่ม ขา ที่ 1 จะเชื่อมอยู่กับขาที่ 2 และ ขาที่ 3 จะเชื่อมอยู่กับขาที่ 4 แต่เมื่อเรากดปุ่มแล้ว จะเกิดการสลับคู่ของขา เพื่อเปลี่ยนแปลงเส้นทางการจ่ายกระแสไฟฟ้าในวงจรนั่นเอง เราสามารถใช้หลักการนี้มาควบคุมการเปิด และ ปิด ของวงจรอิเล็กทรอนิกส์ได้ง่ายๆ

ตัวต้านทาน (resistor)



ตัวต้านทาน (resistor) คืออะไร

- ตัวต้านทานทำหน้าที่ ***"ควบคุมปริมาณกระแสไฟฟ้าและแรงดันระหว่างสองจุดในวงจร"***
- และยังใช้งานอื่น ๆ ได้อีก เช่น เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานความร้อน ตรวจวัดอุณหภูมิ สวิตช์เปิดปิดด้วยแสง

ตัวต้านทางมีกี่แบบ

ตัวต้านทานแบ่งตามการใช้งานได้ 3 แบบ

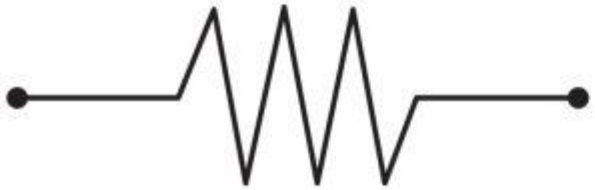
1. ตัวต้านทานชนิดค่าคงตัว
2. ตัวต้านทานปรับค่าได้
3. ตัวต้านทานปรับค่าตามสภาพแวดล้อม

ในบทเรียนนี้จะทำความรู้จักกับ **ตัวต้านทานชนิดค่าคงตัว (fixed resistor)** มีค่าความต้านทานคงที่ มีให้เลือกหลายวัสดุและกำลังวัตต์ โดยขนาดตัวต้านทานจะยึดค่ากำลังไฟฟ้าสูงสุดที่ต้านได้เป็นเกณฑ์ มีตั้งแต่ 0.25-100 วัตต์ โดยใช้รหัสแถบสีเป็นตัวบอกค่า

"ตัวต้านทานชนิดค่าคงตัว แบ่งตามชนิดวัสดุเป็น 5 แบบ"

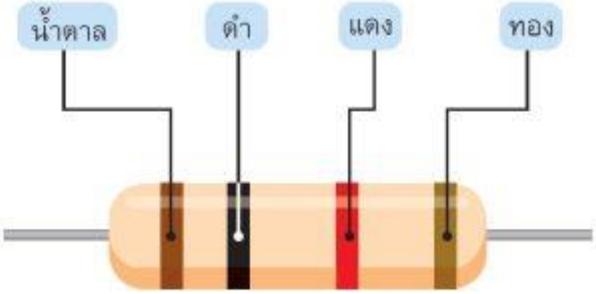
- แบบคาร์บอนผสม
- แบบฟิล์มโลหะออกไซด์
- แบบฟิล์มคาร์บอน
- แบบลวดพัน
- แบบฟิล์มโลหะ

สัญลักษณ์



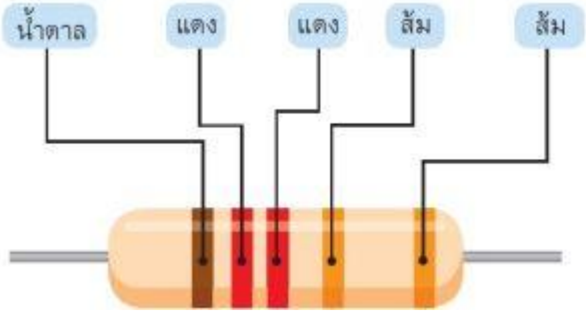
ตัวอย่างการอ่านค่าตัวต้านทาน

จากภาพ ตัวต้านทานมีค่าความต้านทานเท่าใด



จากการอ่านค่าแถบสีของตัวต้านทาน จะได้ว่า
ส้ม ส้ม น้ำตาล ทอง = $10 \times 10^{2}\ \Omega \pm 5\%$
$= 1.0\ k\Omega \pm 50\ \Omega$
$= 0.95\ k\Omega - 1.05\ k\Omega$
จะได้ว่า ตัวต้านทานในภาพมีค่าความต้านทานอยู่ในช่วง 0.95 กิโลโอห์ม ถึง 1.05 กิโลโอห์ม

จากภาพ ตัวต้านทานมีค่าความต้านทานเท่าใด



จากการอ่านค่าแถบสีของตัวต้านทาน จะได้ว่า
น้ำตาล แดง แดง ส้ม ส้ม = $122 \times 10^{3}\ \Omega \pm 3\%$
$= 122\ k\Omega \pm 3.66\ k\Omega$
$= 118.34\ k\Omega - 125.66\ k\Omega$
จะได้ว่า ตัวต้านทานในภาพมีค่าความต้านทานอยู่ในช่วง 118.34 กิโลโอห์ม ถึง 125.66 กิโลโอห์ม

การอ่านค่าความต้านทานจากแถบสีบนตัวต้านทานชนิดค่าคงตัว ทำได้โดยอ่านจากแถบสีที่ระบุบนตัวต้านทาน ซึ่งสามารถเทียบกับตารางแสดงค่าของแถบสีบนตัวต้านทาน โดยทั่วไปแถบสีบนตัวต้านทานจะมี 4 แถบ โดยแต่ละแถบสีมีความหมาย ดังนี้

ตารางค่าของแถบสีบนตัวต้านทาน

แถบสี	แถบสีที่ 1	แถบสีที่ 2	แถบสีที่ 3	แถบสีที่ 4
ดำ	0	0	1	-
น้ำตาล	1	1	10^1	$\pm 1\%$
แดง	2	2	10^2	$\pm 2\%$
ส้ม	3	3	10^3	$\pm 3\%$
เหลือง	4	4	10^4	$\pm 4\%$
เขียว	5	5	10^5	$\pm 0.5\%$
น้ำเงิน	6	6	10^6	$\pm 0.25\%$
ม่วง	7	7	-	$\pm 0.1\%$
เทา	8	8	-	-
ขาว	9	9	-	-
ทอง	-	-	10^{-1}	$\pm 5\%$
เงิน	-	-	10^{-2}	$\pm 10\%$
ไม่มีสี	-	-	-	$\pm 20\%$

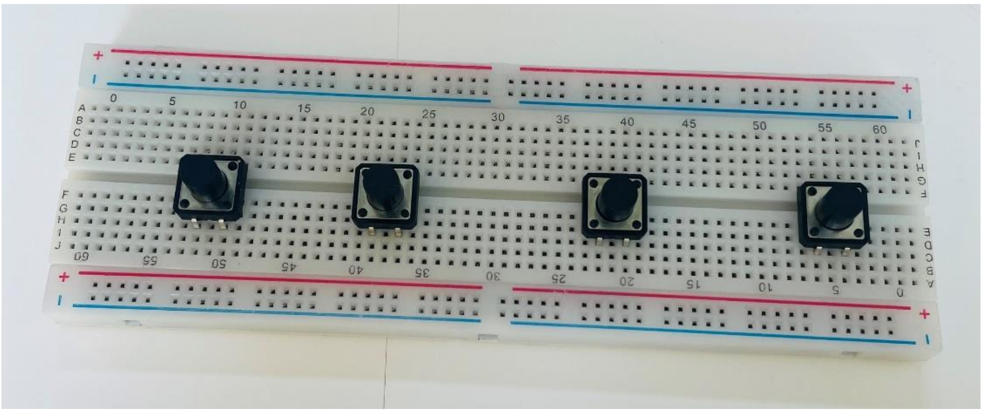
การต่อปุ่มเข้ากับเบรดบอร์ด

อุปกรณ์ที่ใช้

1. ใช้ปุ่ม 4 ปุ่ม
2. เบรดบอร์ด
3. สายแพตัวผู้ – เมีย

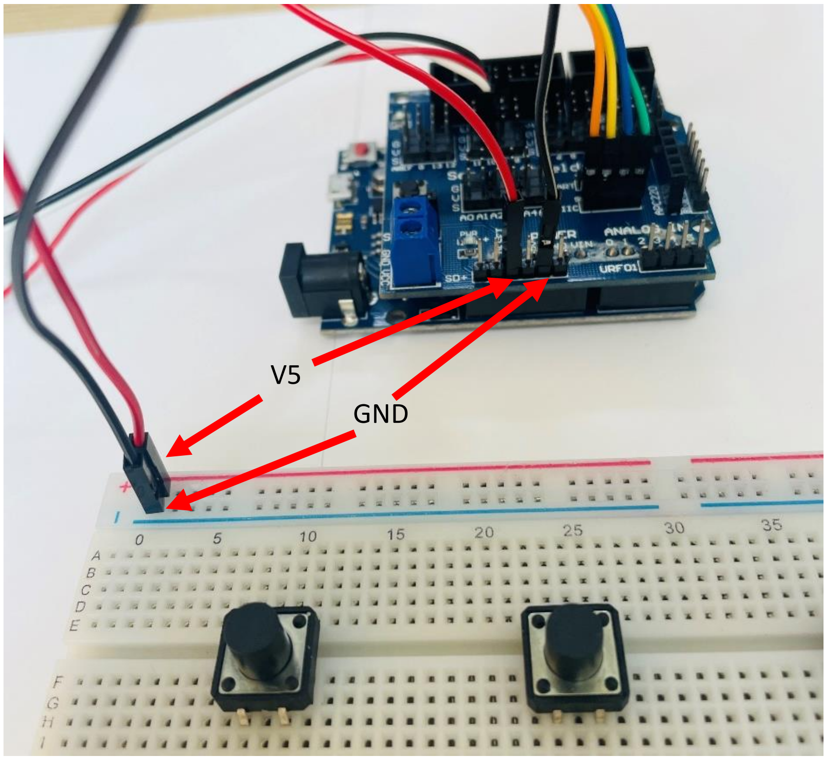
การต่อวงจร

1. นำปุ่มเสียบเข้ากับเบรดบอร์ด

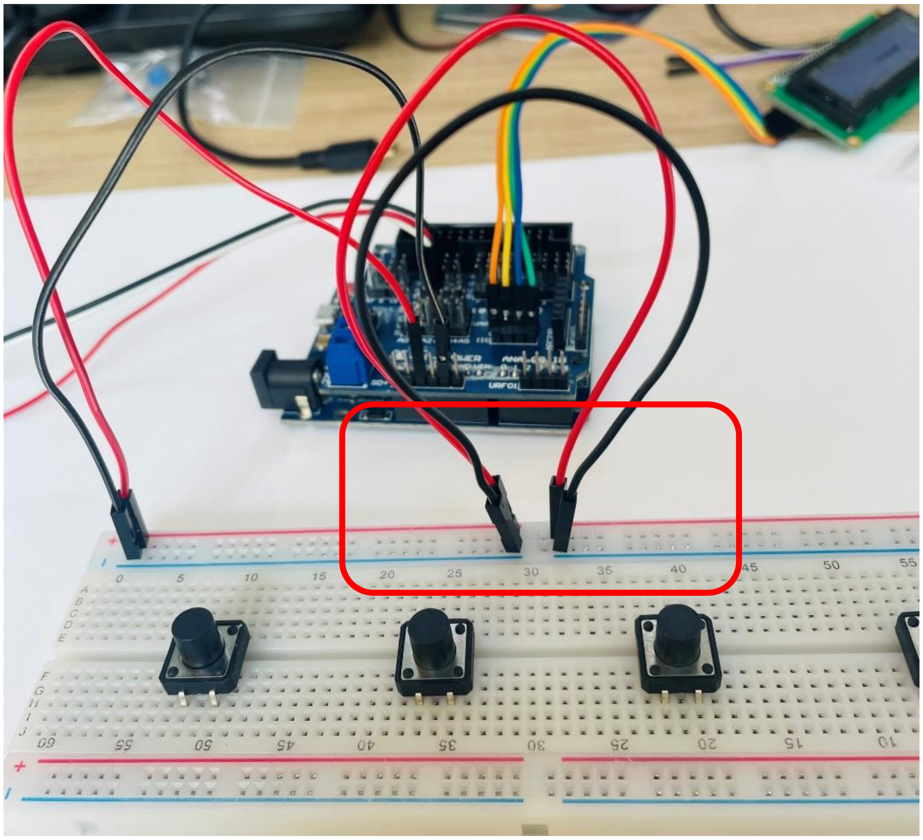


2. นำสายแพเสียบที่บอร์ด Arduino ต่อเข้าเบรดบอร์ด

Arduino		เบรดบอร์ด
V5	=>	+
GND	=>	-

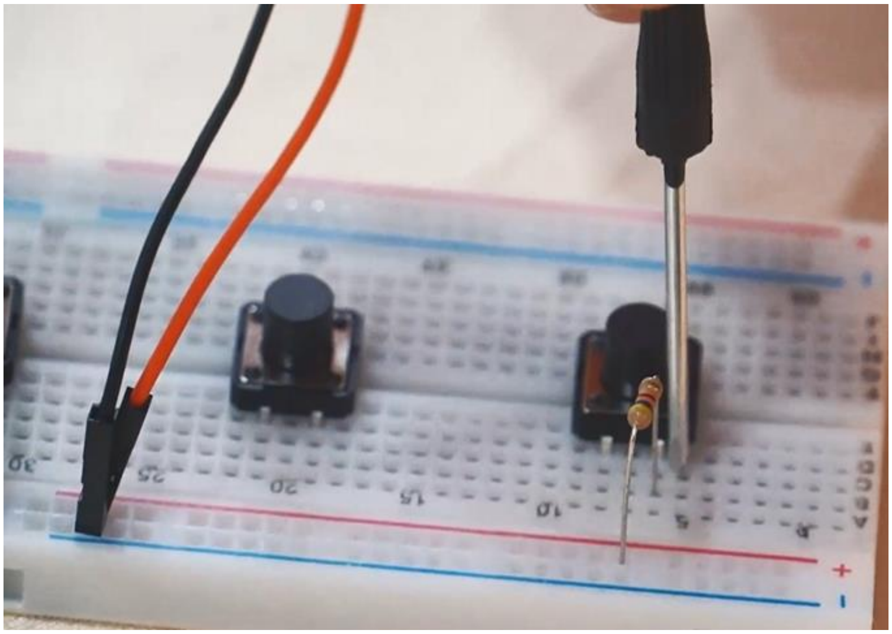


3. กรณีที่เบรดบอร์ดมีการแบ่งครึ่ง เราต้องจั๊มไฟกับกราวด์ ให้ถึงกัน

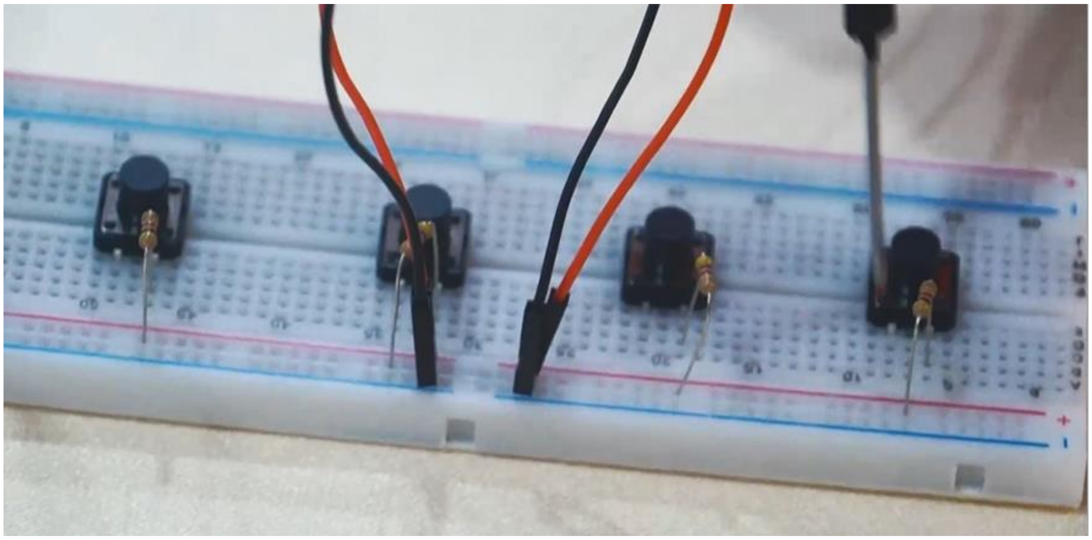


4. ใช้ตัวต้านทาน 4.7 โอม 4 ตัว มางอขาเป็นตัวยู

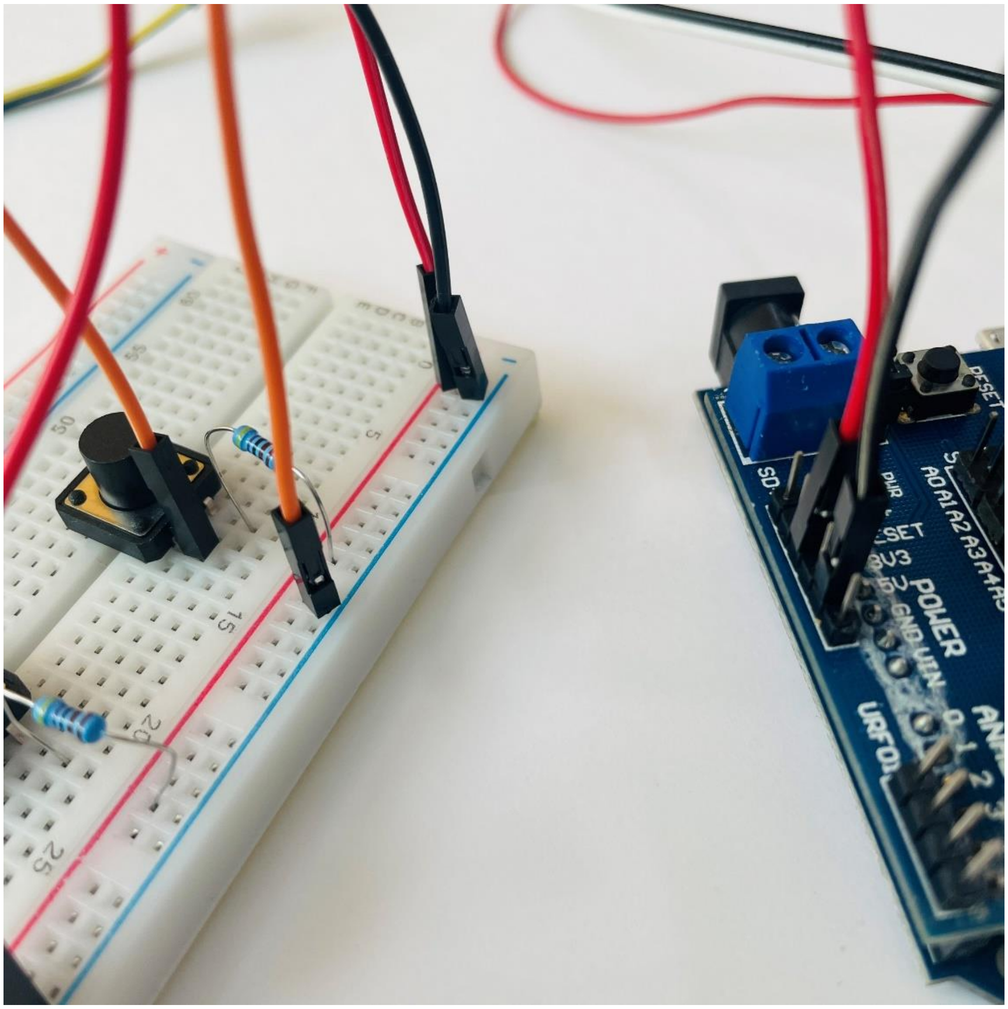
5. นำตัวต้านทานมาเสียบช่องตรงกับขาของปุ่ม อีกฝั่งให้ไปเสียบที่ขั้วบวกบนเบรดบอร์ดตามภาพ



เสียบตัวต้านทานให้ครบทั้ง 4 ปุ่ม



นำสายไฟให้เสียบที่ขาของปุ่ม และสายไฟอีกฝั่งให้เสียบที่ขั้วลบ เสียบให้ครบทั้ง 4 ปุ่ม



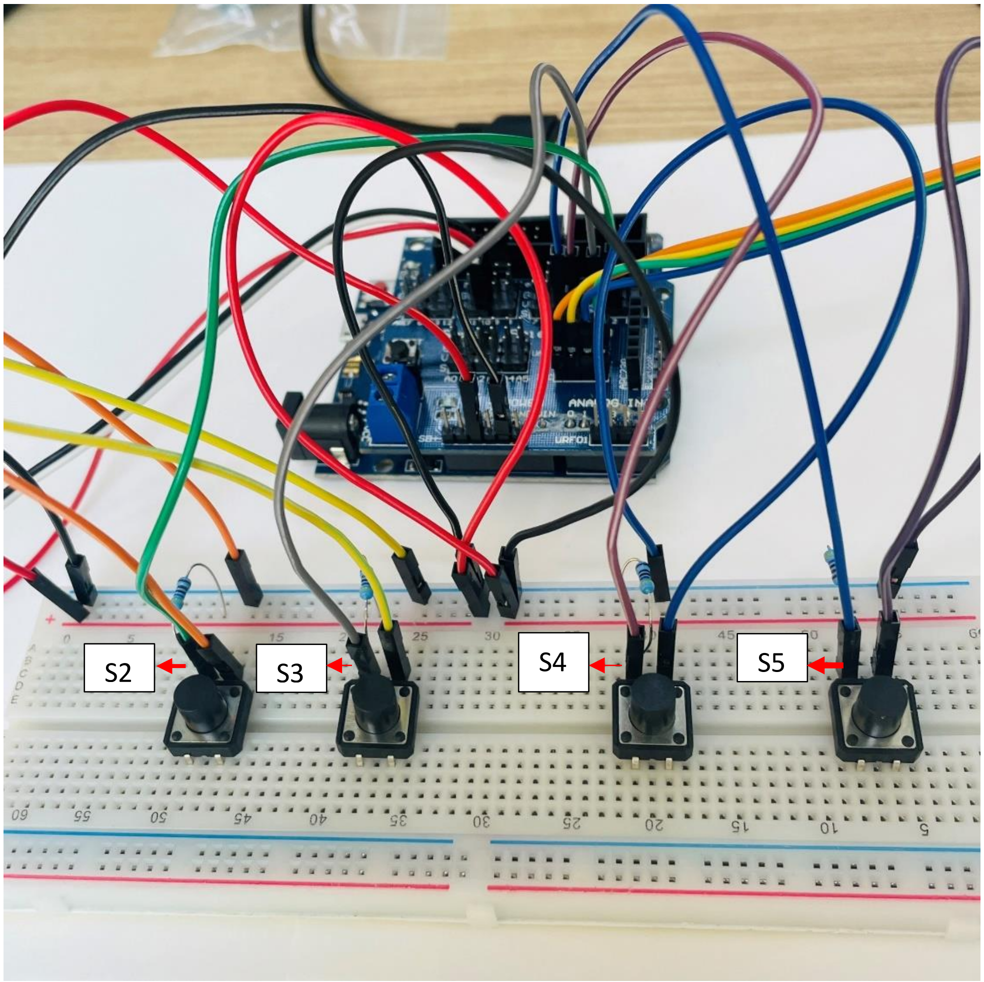
6. เตรียมสายไฟ ผู้-เมีย 4 เส้น เพื่อเสียบกับขาที่ต่อกับตัวต้านทาน และอีกฝั่งไปยังบอร์ด Arduino

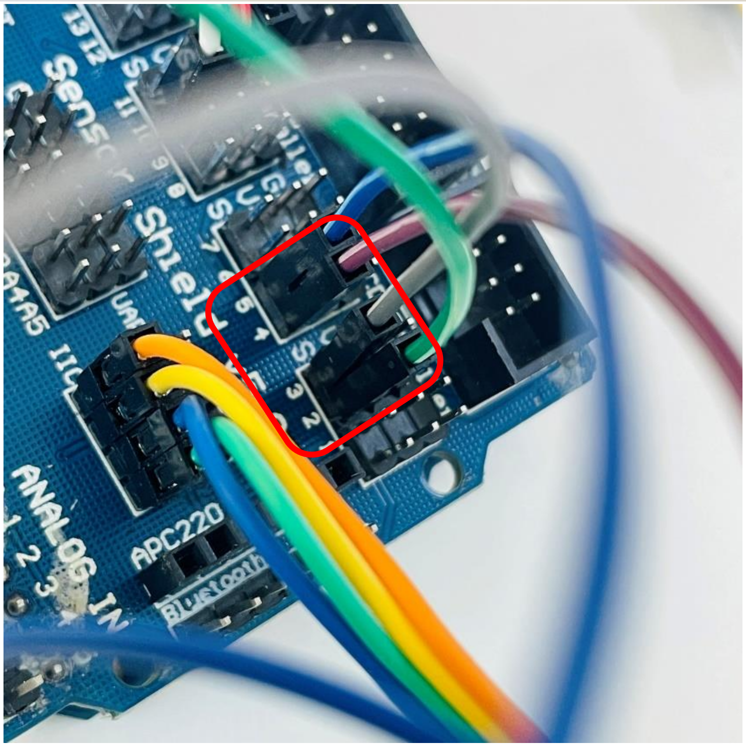
สายไฟที่ต่อจากปุ่มที่ 1 เสียบที่บอร์ดช่องสัญญาณที่ S2

สายไฟที่ต่อจากปุ่มที่ 2 เสียบที่บอร์ดช่องสัญญาณที่ S3

สายไฟที่ต่อจากปุ่มที่ 3 เสียบที่บอร์ดช่องสัญญาณที่ S4

สายไฟที่ต่อจากปุ่มที่ 4 เสียบที่บอร์ดช่องสัญญาณที่ S5


S2
S3
S4
S5


APC220

เริ่มต้นเขียนคำสั่งให้ปุ่ม

ปุ่มที่ 1 ใช้เลือกเมนูที่ 1

ปุ่มที่ 2 ใช้เลือกเมนูที่ 2

ปุ่มที่ 3 ให้เป็นปุ่ม OK (ตกลง)

ปุ่มที่ 4 ให้เป็นปุ่ม Cancel (ยกเลิก)

กำหนดตัวแปรของปุ่ม

```
// ปุ่มที่ 1
int buttonsw_1 = 1;
bool chk_sw1 = true;
int pin_sw1 = 2;

// ปุ่มที่ 2
int buttonsw_2 = 1;
bool chk_sw2 = true;
int pin_sw2 = 3; // กำหนดช่องที่เสียบสาย

// ปุ่มที่ 3
int buttonsw_ok = 1;
bool chk_sw3 = true;
int pin_swok = 4; // กำหนดช่องที่เสียบสาย

// ปุ่มที่ 4
int buttonsw_cancel = 1;
bool chk_sw4 = true;
int pin_swcancel = 5; // กำหนดช่องที่เสียบสาย

bool menu1_select = false;
bool menu2_select = false;
```

ประกาศการรับค่าของตัวแปร

```
pinMode (pin_coin_10, INPUT); // กำหนดรับค่าข้อมูล

pinMode (pin_sw1, INPUT_PULLUP); // กำหนดปุ่มให้รับค่าแบบ Pullup
pinMode (pin_sw2, INPUT_PULLUP)
pinMode (pin_swok, INPUT_PULLUP)
pinMode (pin_swcancel, INPUT_PULLUP)
```

สร้างฟังก์ชัน Select Menu ทดสอบการรับค่าของปุ่ม

```
void select_menu(){
  buttonsw_1 = digitalRead (pin_sw1); // กำหนดให้ปุ่มอ่านค่า
  buttonsw_2 = digitalRead (pin_sw2);
  buttonsw_ok = digitalRead (pin_swok);
  buttonsw_cancel = digitalRead (pin_swcancel);
  Serial.print(buttonsw_1);
  Serial.print(buttonsw_2);
  Serial.print(buttonsw_ok);
  Serial.print(buttonsw_cancel);

}
```

เขียนคำสั่งต่อในส่วนของ Setup กำหนดค่า ให้แสดงผลออกทาง monitor ก่อน

```
void setup() {
  Serial.begin (9600); // ให้แสดงผลออก monitor
  lcd.begin(16,2); //เปิดใช้งาน Lcd หน้าจอแสดง 16 ตัวอักษร 2 บรรทัด
  lcd.setBacklight (255); //กำหนดค่าความสว่างของไฟbacklight มีค่าตั้งแต่ 0-255
```

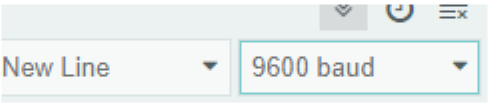
และไปเปิดใช้งานฟังก์ชัน select_menu ที่เราคอมเม้นไว้

```
//
void loop() { // การทำงานใน loop
  if(count_coin_10 == 0){ //ถ้ายังไม่มีการหยอดเหรียญให้ทำฟังก์ชัน sensor_coin
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Insert coin");
    lcd.setCursor(0,1);
    lcd.print("Get coin 10B");
    sensor_coin();
  }else if(count_coin_10 == 1){ //ถ้า มีการหยอดไปแล้ว 1 เหรียญให้ทำฟังก์ชัน sensor_coin
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("Insert coin");
    lcd.setCursor(0,1);
    lcd.print("coin balance 1");
    sensor_coin();
  }else{ //ถ้าหยอดเหรียญครบ 2 เหรียญแล้วให้ทำฟังกชัน select_menu
    select_menu();
    lcd.clear();
    lcd.setCursor(0,0);
    lcd.print("select menu");

  }
```

จากนั้นทำการ upload คำสั่งลงบอร์ด

ที่โปรแกรม Arduino IDE คลิกที่ serial monitor

เลือกเป็น 9600 ตามที่เราเขียนคำสั่งไว้

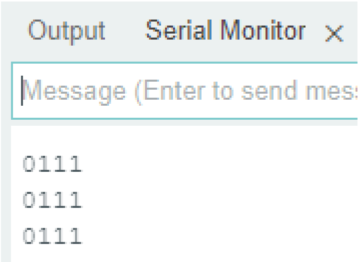


คำสั่งในการทดสอบปุ่มกดให้แสดงผลออกทาง Monitor

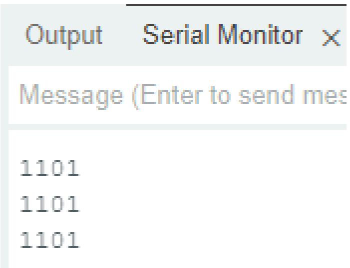
```
void select_menu(){
  buttonsw_1 = digitalRead (pin_sw1); // กำหนดให้ปุ่มอ่านค่า
  buttonsw_2 = digitalRead (pin_sw2);
  buttonsw_ok = digitalRead (pin_swok);
  buttonsw_cancel = digitalRead (pin_swcancel);
  Serial.print(buttonsw_1);
  Serial.print(buttonsw_2);
  Serial.print(buttonsw_ok);
  Serial.println(buttonsw_cancel);

}
```

ทดสอบการกดปุ่ม

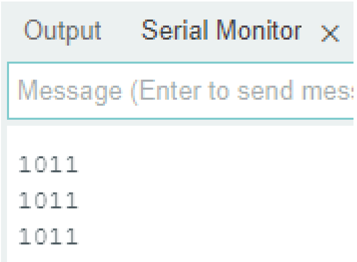
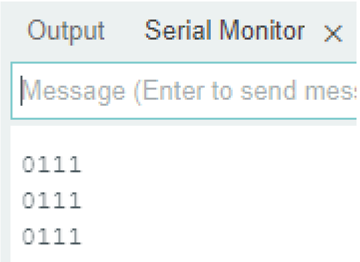
กดปุ่มที่ 1 ผลที่ได้



กดปุ่มที่ 2 ผลที่ได้



กดปุ่มที่ 3 ผลที่ได้



กดปุ่มที่ 4 ผลที่ได้