

## ใบงานที่ 4.1

### การทดสอบบอร์ดเบื้องต้น

#### 1. จุดประสงค์ทั่วไป

เพื่อให้สามารถทดสอบบอร์ดเบื้องต้นได้

#### 2. จุดประสงค์เชิงพฤติกรรม

1. บอกวิธีการทดสอบบอร์ดเบื้องต้นได้
2. บอกขั้นตอนการต่อวงจรเพื่อทดลองบนบอร์ดทดลองได้
3. ทดสอบการทำงานของบอร์ด Arduino ได้
4. ปฏิบัติงานตามใบงานเสร็จทันเวลาที่กำหนด

#### 3. เครื่องมือและอุปกรณ์

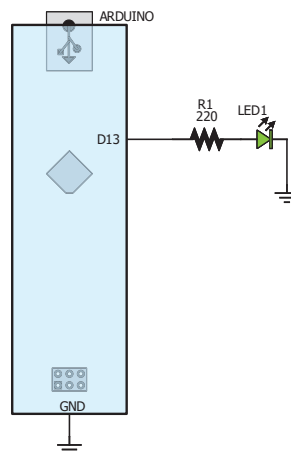
- |                                    |           |
|------------------------------------|-----------|
| 1. บอร์ดทดลอง Breadboard 830 Point | 1 แผ่น    |
| 2. อุปกรณ์อิเล็กทรอนิกส์ประกอบด้วย |           |
| 2.1 ตัวต้านทาน 220 ohm             | 1 ตัว     |
| 2.2 LED 3mm                        | 1 ตัว     |
| 3. บอร์ด Arduino Nano 3.0          | 1 ตัว     |
| 4. สายเชื่อมต่อ USB (Mini USB)     | 1 เส้น    |
| 5. คอมพิวเตอร์                     | 1 เครื่อง |

#### 4. ลำดับขั้นตอนการปฏิบัติงาน

1. ศึกษาจุดประสงค์ทั่วไปจุดประสงค์เชิงพฤติกรรมและทฤษฎีพื้นฐานที่เกี่ยวข้อง
2. ดำเนินการต่อวงจรลงบอร์ดทดลองตามวงจรที่กำหนด
3. เขียนโปรแกรมควบคุมและทดสอบการทำงานของวงจร
4. สรุปผลการปฏิบัติงาน

## 5. ทฤษฎีพื้นฐาน

ใบงานนี้เป็นใบงานสำหรับเตรียมเครื่องมือและทดสอบการทำงานเบื้องต้นของบอร์ด Arduino ว่าใช้งานได้ปกติหรือไม่ โดยการเขียนโค้ดโปรแกรมให้ LED ที่อยู่บนบอร์ด Arduino ซึ่งต่ออยู่ที่ขา D13 ติดและดับสลับกัน ในกรณีที่ต้องการต่อวงจร LED เพิ่มภายนอกจะต้องต่อวงจร LED ให้สามารถติดได้เมื่อรับลอจิก 1 ดังนั้นจะต้องต่อ LED อนุกรมกับตัวต้านทานลงกราวด์ดังรูป



รูปที่ 4.1-1 วงจร Arduino ที่เชื่อมต่อกับ LED จำนวน 1 ตัว

ตัวต้านทานที่ใช้ต่ออนุกรมกับ LED เพื่อจำกัดปริมาณกระแสไฟฟ้าสามารถคำนวณหาได้ด้วยกฎของโอห์ม โดยมีข้อกำหนดที่ค่าแรงดันตกคร่อมและกระแสของ LED เป็น 2V และ 15mA ตามลำดับ ดังนั้นค่าความต้านทานจึงมีค่าเท่ากับ

$$\begin{aligned}
 R &= \frac{V_R}{I} = \frac{V_{CC} - V_{LED}}{I} \\
 &= \frac{5 - 2}{15 \times 10^{-3}} = 200\Omega
 \end{aligned}$$

## 6. ฟังก์ชันที่ใช้งานในใบงาน

รูปแบบของฟังก์ชันที่ใช้งานในใบงานนี้มีดังนี้

1. ฟังก์ชันกำหนดโหมดการทำงานให้กับขาพอร์ต์ สามารถกำหนดได้ทั้งขาดิจิตอลโดยใส่เพียงตัวเลขของขา (0, 1, 2, ..., 13) และขาแอนาล็อกที่ต้องการให้ทำงานในโหมดดิจิตอลได้จะต้องใส่ A นำหน้า ซึ่งใช้ได้เฉพาะ A0, A1, ..., A5 ส่วนขา A6 และ A7 (ที่มีในบอร์ด Arduino รุ่น Mini และ Nano) ไม่สามารถใช้งานในโหมดดิจิตอลได้ รูปแบบของฟังก์ชันเป็นดังนี้

```
pinMode(pin, mode);
```

pin: หมายเลขขาที่ต้องการเซตโหมด

mode: INPUT, OUTPUT, INPUT\_PULLUP

### ตัวอย่างเช่น

`pinMode(13,OUTPUT);`      หมายถึง กำหนดให้ขา D13 ทำงานเป็นเอาต์พุตพอร์ต

`pinMode(12, INPUT_PULLUP);`      หมายถึง กำหนดให้ขา D12 ทำงานเป็นอินพุตพอร์ต  
ที่ใช้ตัวต้านทานพูลอัพภายในชิพ

2. พังก์ชันส่งค่าลอจิกดิจิทัลไปยังขาพอร์ต โดยค่า HIGH เป็นการส่งลอจิก 1 และค่า LOW เป็นการส่งลอจิก 0 ออกไปยังขาพอร์ต พังก์ชันนี้จะทำงานได้ต้องมีการใช้พังก์ชัน `pinMode` ก่อน

**`digitalWrite(pin, value);`**

pin: หมายเลข ขาที่ต้องการเขียนลอจิกออกพอร์ต

value: HIGH or LOW

### ตัวอย่างเช่น

`digitalWrite(13,HIGH);`      หมายถึง กำหนดให้ส่งลอจิก 1 ไปที่ขา D13

`digitalWrite(13,LOW);`      หมายถึง กำหนดให้ส่งลอจิก 0 ไปที่ขา D13

3. พังก์ชันหน่วงเวลาหรือพังก์ชันหยุดค้าง การใช้งานสามารถกำหนดตัวเลขของเวลาที่ต้องการหยุดค้างโดยตัวเลขที่ใส่เป็นตัวเลขของเวลาหน่วยเป็นมิลลิวินาที ตัวเลขของเวลาที่ใส่ได้สูงสุดคือ 4,294,967,295 ซึ่งเป็นขนาดของตัวแปร unsigned long

**`delay(ms);`**

ms: ตัวเลขที่หยุดค้างของเวลาหน่วยมิลลิวินาที (unsigned long)

### ตัวอย่างเช่น

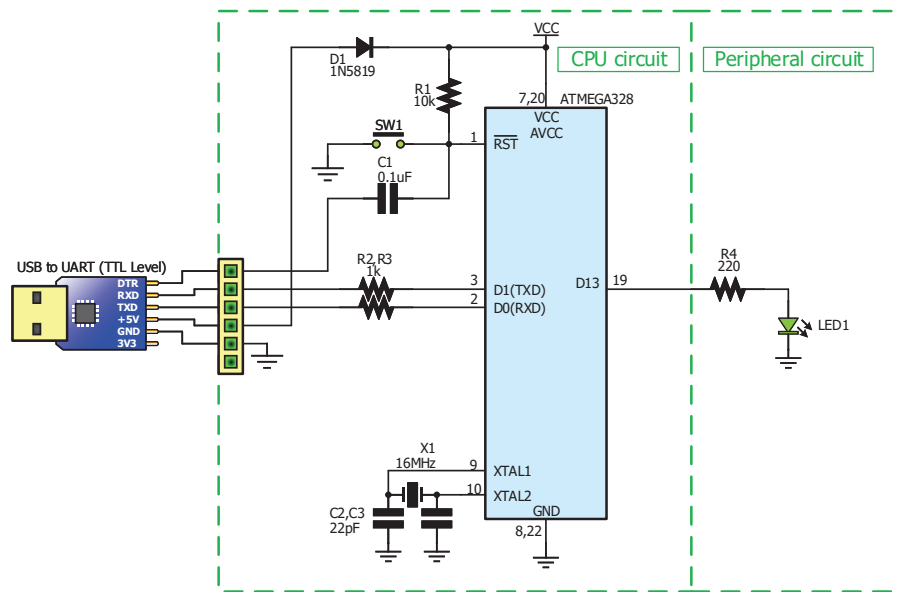
`delay(1000);`      หมายถึง หยุดค้าง (หน่วงเวลา) ไว้เป็นเวลา 1000 มิลลิวินาที (1 วินาที)

## 7. วงจรที่ใช้ทดลอง

วงจรเพื่อใช้ทดลองในใบงานสามารถทำได้ 3 แนวทางคือ

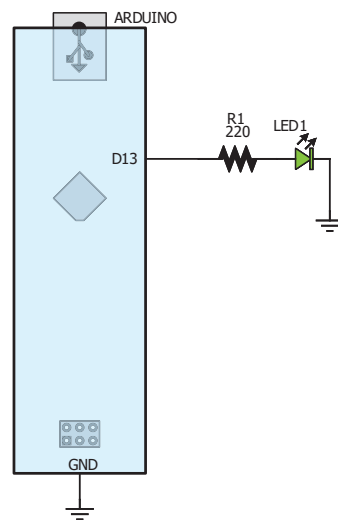
1. วงจรไมโครคอนโทรลเลอร์ Arduino ที่สร้างเองจากไมโครคอนโทรลเลอร์ AVR
2. วงจรไมโครคอนโทรลเลอร์ Arduino ที่ใช้บอร์ดโมดูล Arduino สำเร็จรูป
3. ทดลองด้วยการจำลองการทำงานด้วยโปรแกรม Proteus

กรณีที่ใช้วงจรที่สร้างขึ้นเองจากไอซีไมโครคอนโทรลเลอร์ AVR ที่ลงบูตโหลดเดอร์เป็น Arduino เรียบร้อยแล้ว ต่อวงจรดังรูป



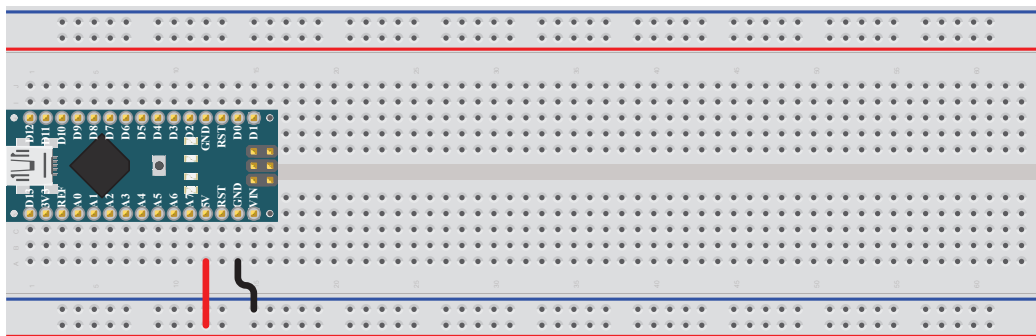
รูปที่ 4.1-2 วงจรที่ใช้ไอซี AVR ในการทดลอง

กรณีที่ใช้ Arduino ในการทดลอง ต้องวงจรดังรูป



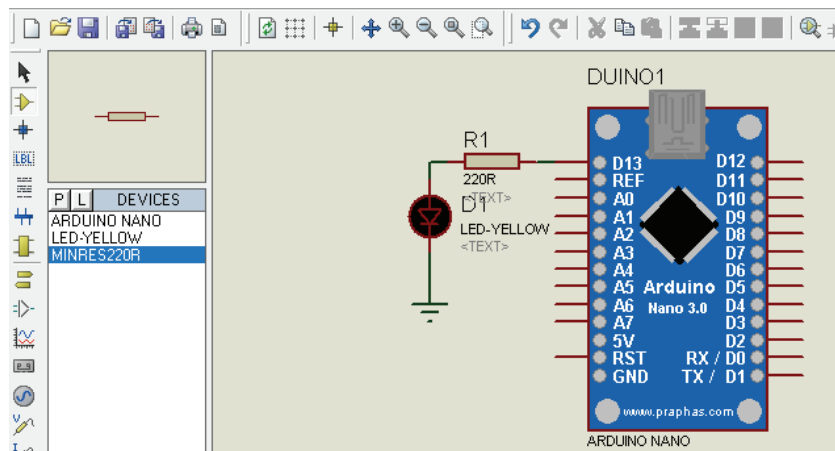
รูปที่ 4.1-3 วงจรที่ใช้บอร์ด Arduino ในการทดลอง

การต่อวงจรไมโครคอนโทรลเลอร์ Arduino ที่ใช้บอร์ดโมดูล Arduino สำเร็จรูปลงบอร์ดทดลอง



รูปที่ 4.1-4 การต่อลงบอร์ดทดลอง

การต่อวงจรเพื่อทดลองด้วยการจำลองการทำงานด้วยโปรแกรม Proteus



รูปที่ 4.1-5 การต่อวงจรทดลองในโปรแกรมจำลองการทำงาน

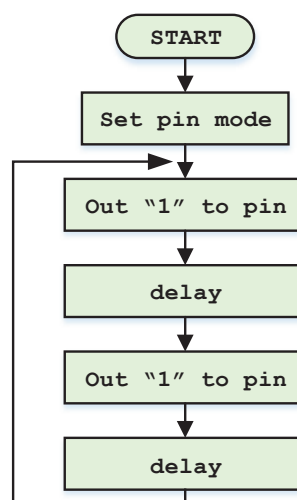
## 8. การเขียนโค้ดโปรแกรมควบคุม

ใบงานนี้เป็นใบงานทดสอบการทำงานของบอร์ด Arduino ว่าทำงานได้สมบูรณ์หรือไม่สามารถรับการอัปเดตจากโปรแกรม Arduino IDE ได้หรือไม่ การทดสอบเป็นการเขียนโค้ดโปรแกรมให้ LED ที่อยู่บนบอร์ดติดดับสลับกันในเวลาที่กำหนดซึ่ง LED ดังกล่าวต่ออยู่กับขาดิจิตอลขา D13 การเขียนโปรแกรมมีขั้นตอนดังนี้

ลำดับงาน การเขียนโปรแกรมจากโจทย์ที่กำหนดข้างต้นเขียนลำดับงานได้ดังนี้

1. กำหนดชื่อตัว LED กับขาพอร์ตที่ต้องใช้งาน	5. ส่งค่า LOW ไปยังขาพอร์ตเพื่อให้ LED ดับ
2. กำหนดโหมดขาที่เชื่อมต่อ LED	6. หน่วงเวลา
3. ส่งค่า HIGH ไปยังขาพอร์ตเพื่อให้ LED ติด	7. วนกลับไปทำลำดับที่ 3 ซ้ำ
4. หน่วงเวลา	

แปลงลำดับงานเป็นผังงาน จากลำดับงานสามารถเขียนเป็นผังงานได้ดังนี้



แปลงผังงานเป็นโปรแกรม จากผังงานสามารถเขียนเป็นโค้ดโปรแกรมควบคุม Arduino

```

1  #define LED 13
2  void setup()
3  {
4      pinMode(LED, OUTPUT);
5  }
6  void loop()
7  {
8      digitalWrite(LED, HIGH); // turn the LED on (HIGH is the voltage level)
9      delay(1000);             // wait for a second
10     digitalWrite(LED, LOW);  // turn the LED off by making the voltage LOW
11     delay(1000);             // wait for a second
12 }

```

รายละเอียดโค้ดโปรแกรม

- บรรทัดที่ 1 ประกาศชื่อ LED แทนตัวเลข 13
- บรรทัดที่ 4 กำหนดโหมดของขาพอร์ตที่เชื่อมต่อ LED ให้ทำงานเป็นเอาต์พุตพอร์ต
- บรรทัดที่ 8 ส่งค่าลอจิก 1 ออกทางขาพอร์ตที่ต่อ LED
- บรรทัดที่ 9 หน่วงเวลา 1 วินาที (1000 mS)
- บรรทัดที่ 10 ส่งค่าลอจิก 0 ออกทางขาพอร์ตที่ต่อ LED
- บรรทัดที่ 11 หน่วงเวลา 1 วินาที (1000 mS)

ทดลองการทำงาน

1. จำลองการทำงานด้วยโปรแกรม Proteus
2. ทดลองด้วยวงจรจริง
3. ทดลองแก้โปรแกรมจาก delay(1000); เป็นตัวเลขอื่น แล้วสังเกตผล

## 9. สรุปผลการปฏิบัติงาน

---

---

---

---

---

---

---

---