ใบงานการทดลอง
เรื่อง เริ่มต้นกับ Arduino โปรแกรม "Blink"

จุดประสงค์การเรียนรู้

1. นักเรียนสามารถอธิบายคำสั่ง pinMode(), digitalWrite(), และ delay() ได้
2. นักเรียนสามารถต่อวงจรไฟ LED เข้ากับบอร์ด ARDUINO ได้อย่างถูกต้องปลอดภัยและเขียนโค้ดพร้อมทั้งอัปโหลดได้ปลอดภัย
3. นักเรียนเอาใจใส่ต่องานที่ปฏิบัติเมื่อใช้งานเสร็จจัดเก็บอุปกรณ์เรียบร้อย (A)

เครื่องมือและอุปกรณ์การทดลอง

1. LED
2. บอร์ด Arduino
3. สายจั๊มเปอร์
4. สาย USB
5. ตัวต้านทาน
6. โพโต้บอร์ด
7. โมดูลไฟจราจร

ภาคทฤษฎี

1. โครงสร้างพื้นฐานของโปรแกรม Arduino

โปรแกรม Arduino ทุกโปรแกรมจะมีโครงสร้างหลักอยู่ 2 ส่วนเสมอ คือ void setup() และ void loop()

- void setup() { ... }

  - เป็นส่วนของโค้ดที่จะทำงาน เพียงครั้งเดียว เมื่อ Arduino เริ่มต้นทำงาน (เช่น เมื่อเปิดเครื่อง หรือกดปุ่ม Reset)

  - เรามักจะใช้ส่วนนี้ในการกำหนดค่าเริ่มต้นต่างๆ เช่น การตั้งค่าขา (Pin) ของ Arduino ว่าจะให้เป็นขา Input (รับสัญญาณเข้า) หรือ Output (ส่งสัญญาณออก)

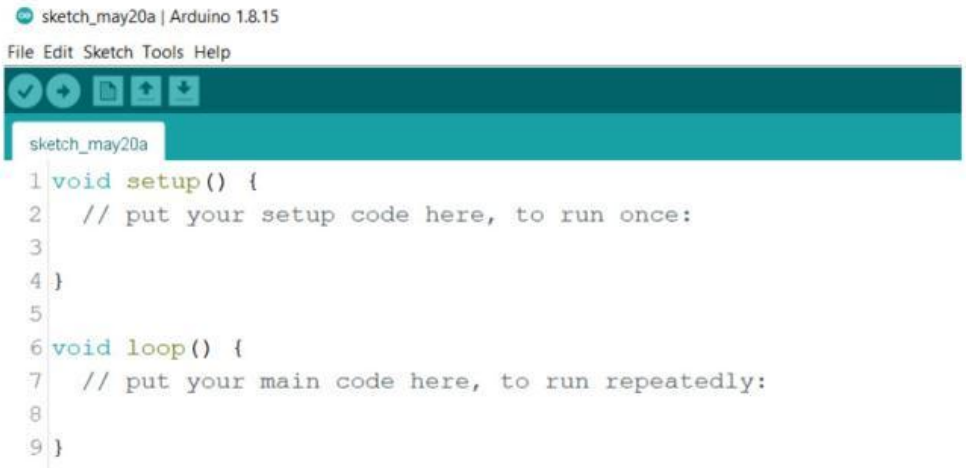
  - ตัวอย่าง: pinMode(13, OUTPUT);

- void loop() { ... }

  - เป็นส่วนของโค้ดที่จะทำงาน วนซ้ำไปเรื่อยๆ อย่างไม่มีที่สิ้นสุด ตราบใดที่ Arduino ยังคงทำงานอยู่

  - คำสั่งที่เราต้องการให้ Arduino ทำซ้ำๆ จะถูกเขียนไว้ในส่วนนี้ เช่น การเปิด-ปิดไฟ LED, การอ่านค่าจากเซ็นเซอร์ หรือการควบคุมมอเตอร์

  - ตัวอย่าง: digitalWrite(13, HIGH);



LIVEWORKSHEETS

2. คำสั่ง pinMode() (กำหนดโหมดการทำงานของขา)

คำสั่งนี้ใช้เพื่อกำหนดว่าขา (Pin) ใดของ Arduino จะทำหน้าที่เป็นอะไร

- OUTPUT (ขาออก) ใช้เมื่อต้องการให้ Arduino ส่งสัญญาณออกไป เพื่อควบคุมอุปกรณ์ต่างๆ เช่น เปิด/ปิดไฟ LED, หมุนมอเตอร์
- INPUT (ขาเข้า) ใช้เมื่อต้องการให้ Arduino รับสัญญาณเข้ามา จากอุปกรณ์ภายนอก เช่น อ่านค่าจากปุ่มกด, เซ็นเซอร์

รูปแบบการใช้งาน pinMode(หมายเลขขา, โหมดการทำงาน);

- หมายเลขขา คือหมายเลขขา Digital Pin บนบอร์ด Arduino (เช่น 0-13) หรือใช้ LED_BUILTIN ซึ่งหมายถึง LED ที่ติดมากับบอร์ด (ส่วนใหญ่คือขา 13)
- โหมดการทำงาน: คือ OUTPUT หรือ INPUT

ตัวอย่าง

- pinMode(13, OUTPUT); // กำหนดให้ขา 13 เป็นขา Output
- pinMode(LED_BUILTIN, OUTPUT); // กำหนดให้ LED ที่มากับบอร์ดเป็นขา Output (ส่วนใหญ่คือขา 13)

3. คำสั่ง digitalWrite() (ส่งสัญญาณ Digital)

คำสั่งนี้ใช้เพื่อ ส่งสัญญาณ Digital (เปิด-ปิด) ไปยังขาที่กำหนดไว้ว่าเป็น OUTPUT

รูปแบบการใช้งาน digitalWrite(หมายเลขขา, ค่าสัญญาณ);

- หมายเลขขา: คือหมายเลขขา Digital Pin ที่กำหนดเป็น OUTPUT ไว้แล้ว
- ค่าสัญญาณ:
  - HIGH: ส่งสัญญาณแรงดันสูง (ประมาณ 5V) หรือสถานะ "เปิด" (ทำให้ LED ติด)
  - LOW: ส่งสัญญาณแรงดันต่ำ (ประมาณ 0V) หรือสถานะ "ปิด" (ทำให้ LED ดับ)

ตัวอย่าง

- digitalWrite(13, HIGH); // ส่งสัญญาณ HIGH ไปที่ขา 13 (LED ติด)
- digitalWrite(13, LOW); // ส่งสัญญาณ LOW ไปที่ขา 13 (LED ดับ)

4. คำสั่ง delay() (หน่วงเวลา)

คำสั่งนี้ใช้เพื่อ หยุดหรือหน่วงเวลา การทำงานของโปรแกรมชั่วคราวตามเวลาที่กำหนด

รูปแบบการใช้งาน delay(เวลาเป็นมิลลิวินาที);

- เวลาเป็นมิลลิวินาที: ตัวเลขที่ระบุเวลาเป็นหน่วยมิลลิวินาที (ms)
  - 1000 มิลลิวินาที = 1 วินาที
  - 500 มิลลิวินาที = 0.5 วินาที

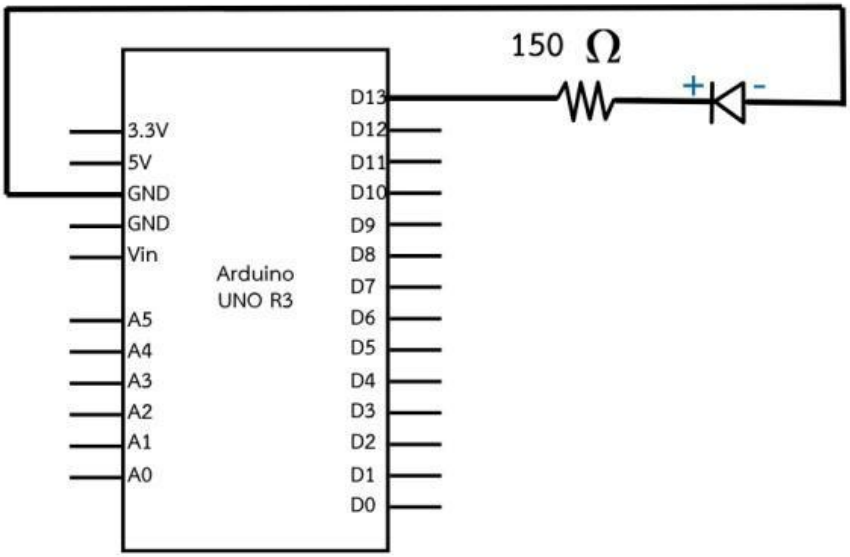
ตัวอย่าง

- delay(1000); // โปรแกรมจะหยุดรอ 1 วินาที
- delay(500); // โปรแกรมจะหยุดรอ 0.5 วินาที

LIVEWORKSHEETS

การทดลองที่ 1 วงจรควบคุมไฟกระพริบ

ขั้นตอนที่ 1 การต่อวงจรไฟ LED



ต่อวงจรอนุกรมกับอุปกรณ์จริงตามรูปด้านบนและเพื่อป้องกัน LED เสียหาย เราต้องต่อตัวต้านทานอนุกรมกับ LED เสมอ

1.1 นำ ไฟ LED มาเสียบลงบน Breadboard โดยให้ขาที่ยาวกว่า (ขา Anode) เสียบอยู่ในแถวหนึ่งในแนวตั้ง และขาที่สั้นกว่า (ขา Cathode) อยู่ในอีกแถวหนึ่ง ที่ไม่เชื่อมกัน

1.2 นำ ตัวต้านทาน 150 โอห์ม มาต่ออนุกรมกับขาที่ยาวกว่า (Anode) ของ LED

1.3 นำ สายไฟจัมเปอร์สีแดงหรือสีส้ม มาต่อจากขาที่เหลือของตัวต้านทาน (ปลายที่ไม่ได้ต่อกับ LED) ไปยัง ขา Digital Pin 13 บนบอร์ด Arduino

1.4 นำ สายไฟจัมเปอร์สีดำ มาต่อจาก ขาที่สั้นกว่า (Cathode) ของ LED ไปยัง ขา GND บนบอร์ด Arduino ฝั่งพอร์ต POWER

ขั้นตอนที่ 2 เตรียมพร้อมอุปกรณ์และซอฟต์แวร์

2.1 เปิดโปรแกรม Arduino IDE บนคอมพิวเตอร์ของคุณ (ตรวจสอบว่าได้ติดตั้งเรียบร้อยแล้ว)

2.2 เชื่อมต่อบอร์ด Arduino เข้ากับคอมพิวเตอร์ด้วยสาย USB

2.3 ตั้งค่าบอร์ดและพอร์ต:

- ไปที่เมนู Tools (เครื่องมือ) > Board (บอร์ด) > เลือก Arduino Uno
- ไปที่เมนู Tools (เครื่องมือ) > Port (พอร์ต) > เลือกพอร์ตที่ปรากฏชื่อ Arduino Uno (เช่น COM3, COM4 หรือ /dev/cu.usbmodemxxxx)

2.4 ตรวจสอบวงจร : ดูรูปภาพประกอบและตรวจสอบว่าวงจรของนักเรียนต่อถูกต้องหรือไม่ หากไม่แน่ใจให้สอบถามคุณครู

LIVEWORKSHEETS

ขั้นตอนที่ 3 การเขียนโปรแกรม "Blink"

3.1 ในโปรแกรม Arduino IDE ให้ลบโค้ดเดิมทั้งหมด (ถ้ามี)

3.2 เขียนโค้ดด้านล่างนี้ ในหน้าต่างโปรแกรม Arduino IDE:

```
void setup() {
        pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
      digitalWrite(LED_BUILTIN, HIGH);   // อุปกรณ์  LED  ส่งสัญญาณ HIGH
      delay(1000);                       // หน่วงเวลาการทำงาน 1 วินาที
      digitalWrite(LED_BUILTIN, LOW);  // อุปกรณ์ LED ส่งสัญญาณ HIGH
      delay(1000);                       // หน่วงเวลาการทำงาน 1 วินาที
}
```

ขั้นตอนที่ 4 การตรวจสอบและอัปโหลดโปรแกรม

4.1 คลิกปุ่ม "Verify" (เครื่องหมายถูก) ที่มุมซ้ายบนของโปรแกรม Arduino IDE เพื่อตรวจสอบความถูกต้องของโค้ด หากไม่มีข้อผิดพลาด จะขึ้นข้อความ "Done compiling."

4.2 คลิกปุ่ม "Upload" (ลูกศรขวา) เพื่ออัปโหลดโปรแกรมไปยังบอร์ด Arduino

สังเกตผล: หากอัปโหลดสำเร็จ ไฟ LED ที่คุณต่อไว้ และ LED เล็ก ๆ บนบอร์ด Arduino (จะเขียนว่า "L" หรืออยู่ใกล้ขา 13) จะเริ่มกระพริบ!

ตารางบันทึกผลการทดลอง (ให้เรียนทดลองปรับเปลี่ยนค่า delay() ในโค้ด และบันทึกผล)

ครั้งที่	ค่า delay( ) เมื่อ LED ติด	ค่า delay( ) เมื่อ LED ดับ	ลักษณะการกะพริบของ LED
1	1000	1000	
2	500	500	
3	200	200	
4	50	50	

LIVEWORKSHEETS

คำถามท้ายการทดลองที่ 1

1. คำสั่ง pinMode(LED_BUILTIN, OUTPUT); ทำหน้าที่อะไรในโปรแกรม?

2. ถ้านักเรียนต้องการให้ LED ติดค้างตลอดเวลา จะต้องแก้ไขโค้ดในส่วน void loop() อย่างไร?

3. ถ้านักเรียนต้องการให้ LED กระพริบเร็วมาก ๆ จนมองแทบไม่ทันทำ ควรปรับค่า delay() เป็นเท่าไร? (ลองเปลี่ยนและสังเกตผล)

4. นักเรียนคิดว่า เพราะเหตุใดเราจึงต้องต่อตัวต้านทานอนุกรมกับ LED เสมอในการใช้งานกับ Arduino?

สรุปผลการทดลองที่ 1

LIVEWORKSHEETS

การทดลองที่ 2 วงจรควบคุมไฟจราจร

ขั้นตอนที่ 1 เตรียมอุปกรณ์และซอร์ฟแวร์

1.1 เปิดโปรแกรม Arduino IDE พร้อมทั้งสร้างไฟล์ใหม่ด้วยคำสั่ง File → New

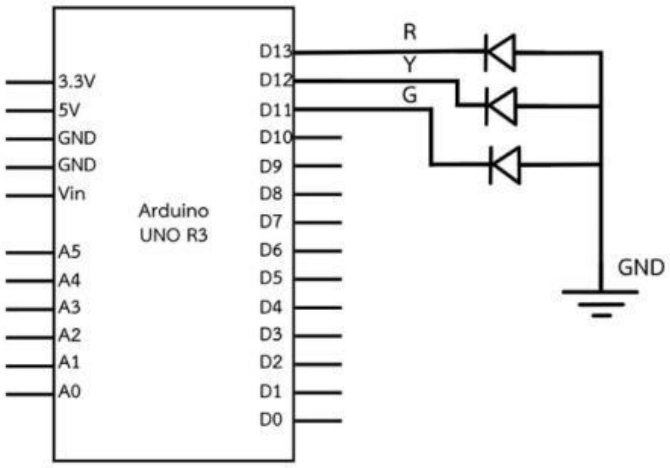
1.2 เชื่อมต่อบอร์ด Arduino เข้ากับคอมพิวเตอร์ด้วยสาย USB ให้นักเรียนเลือกพอร์ตและบอร์ดให้ถูกต้อง

ขั้นตอนที่ 2 การต่อวงจรไฟ LED โมดูลไฟจราจร

2.1 ให้นักเรียนสังเกตุโมดูลไฟจราจร จะมีทั้งหมด 4 ขา ระบุค่าดังนี้ R, Y, G, GND

2.2 ต่อวงจรตามรูปด้านล่าง โดยนำสายจั๊มเปอร์ชนิดผู้-เมีย ด้านเมียต่อเข้าโมดูลไฟจราจร ด้านผู้ต่อเข้า Digital Pin ดังนี้

- ขาของ R ต่อเข้าที่ Digital Pin 2
- ขาของ Y ต่อเข้าที่ Digital Pin 3
- ขาของ G ต่อเข้าที่ Digital Pin 4
- ขาของ GND ต่อเข้าที่ POWER Pin GND



ขั้นตอนที่ 3 การเขียนโปรแกรมโมดุลไฟจราจร

3.1 โค้ดเริ่มต้นใน void setup() ให้นักเรียนกำหนด pinMode() ของขา Digital Pin 2, 3, 4 ให้เป็น OUTPUT

```
void setup() {
        pinMode(13, OUTPUT); // ขาไฟแดง
        pinMode(12, OUTPUT); // ขาไฟเหลือง
        pinMode(11, OUTPUT); // ขาไฟเขียว
}
```

3.2 ให้นักเรียนเขียนโค้ดในส่วน void loop() เพื่อควบคุมไฟจราจรให้ทำงานตามลำดับ โดยแต่ละสถานะจะมีไฟติดและดับต่างกันไป

LIVEWORKSHEETS

```
void loop() {
  // สถานะที่ 1: ไฟเขียวติด, ไฟแดงและเหลืองดับ
  digitalWrite(13, LOW);  // ไฟแดงดับ
  digitalWrite(12, LOW);  // ไฟเหลืองดับ
  digitalWrite(11, HIGH); // ไฟเขียวติด
  delay(5000);            // หน่วงเวลา 5 วินาที

  // สถานะที่ 2: ไฟเหลืองติด, ไฟแดงและเขียวดับ
  digitalWrite(13, LOW);   // ไฟแดงดับ
  digitalWrite(12, HIGH);  // ไฟเหลืองติด
  digitalWrite(11, LOW);   // ไฟเขียวดับ
  delay(2000);             // หน่วงเวลา 2 วินาที

  // สถานะที่ 3: ไฟแดงติด, ไฟเหลืองและเขียวดับ
  // ให้นักเรียนเขียนโค้ดตรงนี้เอง
  // ...
  // ...
  // ...
  // delay(7000);            // หน่วงเวลา 7 วินาที
}
```

3.2 ให้นักเรียน Verify และ upload โค้ดไปยังบอร์ด Arduino

คำถามท้ายการทดลอง

1. อธิบายความหมายของ HIGH และ LOW ในการเปิด-ปิด LED

2. อธิบายวิธีการใช้คำสั่ง delay()

LIVEWORKSHEETS

ขั้นตอนที่ 4 บันทึกผลลงในตาราง

ลำดับ	ไฟแดง (ติด/ดับ)	ไฟเหลือง (ติด/ดับ)	ไฟเขียว (ติด/ดับ)	ระยะเวลา (ms)	ลักษณะการทำงาน
1	ดับ	ดับ	ติด	5000	ไฟเขียวติดค้าง 5 วินาที
2					
3					
4					

คำถามท้ายการทดลอง

1. จากการทดลองหมายนักเรียนใช้ Digital Pin ใดบ้างในการต่อ LED แต่ละสีของหลอดไฟ

2. ถ้าคุณต้องการให้ไฟเขียวติดนานขึ้นเป็นเวลา 10 วินาที จะต้องแก้ไขโค้ดอย่างไร

3. ลองเพิ่มสถานะของไฟจราจรให้ซับซ้อนขึ้น เช่น เมื่อไฟแดงติดแล้วให้ไฟเหลืองติดพร้อมกันชั่วครู่ก่อนที่จะเปลี่ยนเป็นไฟเดียว คุณจะเขียนโค้ดเพิ่มเติมอย่างไร

4. นักเรียนคิดว่าระบบไฟจราจรที่เราสร้างขึ้นสามารถนำไปประยุกต์ใช้อะไรในชีวิตประจำวันได้อีกบ้าง

LIVEWORKSHEETS

สรุปผลการทดลองที่ 2

จากการทดลองโปรแกรม "ไฟจราจร" ในครั้งนี้ นักเรียนได้เรียนรู้และเข้าใจอะไรบ้างเกี่ยวกับการใช้งานโครงสร้างโปรแกรม void setup() และ void loop() รวมถึง การใช้คำสั่ง pinMode(), digitalWrite() (พร้อมสัญญาณ HIGH/LOW), และ delay()

คำถามท้ายบท

คำชี้แจง ให้นักเรียนทำเครื่องหมายกากบาท (X) ลงบนคำตอบที่ถูกที่สุดเพียงคำตอบเดียว

1. โค้ดส่วนใดในโปรแกรม Arduino ที่จะทำงาน เพียงครั้งเดียว เมื่อบอร์ดเริ่มต้นทำงาน?

   ก. void loop() ข. void setup() ค. digitalWrite() ง. delay()

2. โค้ดส่วนใดในโปรแกรม Arduino ที่จะทำงาน วนซ้ำไปเรื่อยๆ อย่างไม่มีที่สิ้นสุด?

   ก. void setup() ข. void loop() ค. pinMode() ง. Serial.begin()

3. คำสั่ง pinMode(5, OUTPUT); มีจุดประสงค์เพื่ออะไร?

   ก. สั่งให้ขา Digital Pin 5 ส่งสัญญาณไฟฟ้าออกไปทันที

   ข. กำหนดให้ขา Digital Pin 5 ทำหน้าที่เป็นขาที่ส่งสัญญาณไฟฟ้าออกไป

   ค. กำหนดให้ขา Digital Pin 5 ทำหน้าที่เป็นขาที่รับสัญญาณไฟฟ้าเข้ามา

   ง. หน่วงเวลาการทำงานของโปรแกรมที่ขา Digital Pin 5 เป็นเวลา 5 มิลลิวินาที

4. เมื่อใช้คำสั่ง digitalWrite(LED_PIN, HIGH); สำหรับ LED ที่ต่ออยู่กับขา LED_PIN ผลลัพธ์ที่เกิดขึ้นคืออะไร?

   ก. LED จะดับลง ข. LED จะติดสว่างขึ้น

   ค. LED จะกะพริบเร็วมาก ง. โปรแกรมจะหยุดทำงาน

LIVEWORKSHEETS

5. หากต้องการให้ LED ที่ขา Digital Pin 7 ดับลง ควรใช้คำสั่งใด?

ก. digitalWrite(7, HIGH); ข. digitalWrite(7, LOW);

ค. pinMode(7, LOW); ง. delay(7000);

6. คำสั่ง delay(2000); ในโปรแกรม Arduino มีหน้าที่อะไร?

ก. ทำให้โปรแกรมทำงานเร็วขึ้น 2000 เท่า

ข. หน่วงเวลาการทำงานของโปรแกรมเป็นเวลา 2 วินาที

ค. ส่งสัญญาณ Digital ไปยังขาหมายเลข 2000

ง. ตรวจสอบสถานะของขา Digital Pin 2000

7. ในคำสั่ง digitalWrite(PIN, HIGH); คำว่า HIGH หมายถึงอะไร?

ก. สัญญาณไฟฟ้าที่มีแรงดันต่ำ (ปิด)

ข. สัญญาณไฟฟ้าที่มีแรงดันสูง (เปิด)

ค. ขา Digital Pin ที่เสียหาย

ง. การหน่วงเวลาเป็นเวลานาน

8. ในคำสั่ง digitalWrite(PIN, LOW); คำว่า LOW หมายถึงอะไร?

ก. สัญญาณไฟฟ้าที่มีแรงดันต่ำ (ปิด)

ข. สัญญาณไฟฟ้าที่มีแรงดันสูง (เปิด)

ค. ขา Digital Pin ที่กำลังทำงานหนัก

ง. การหน่วงเวลาสั้นๆ

9. เหตุใดเราจึงต้องต่อ ตัวต้านทานอนุกรม กับ LED เสมอในการใช้งานกับ Arduino?

ก. เพื่อให้ LED สว่างมากขึ้น

ข. เพื่อป้องกันไม่ให้ LED เสียหายจากกระแสไฟฟ้าที่มากเกินไป

ค. เพื่อให้ LED สามารถเปลี่ยนสีได้

ง. เพื่อให้ Arduino ตรวจจับการทำงานของ LED ได้ง่ายขึ้น

10. ในการจำลองไฟจราจร หากตอนนี้ ไฟเขียวติดสว่างอยู่ โค้ดที่เหมาะสมสำหรับไฟแดงและไฟเหลืองควรเป็นอย่างไร?

ก. ไฟแดง HIGH, ไฟเหลือง HIGH

ข. ไฟแดง LOW, ไฟเหลือง HIGH

ค. ไฟแดง LOW, ไฟเหลือง LOW

ง. ไฟแดง HIGH, ไฟเหลือง LOW

LIVEWORKSHEETS