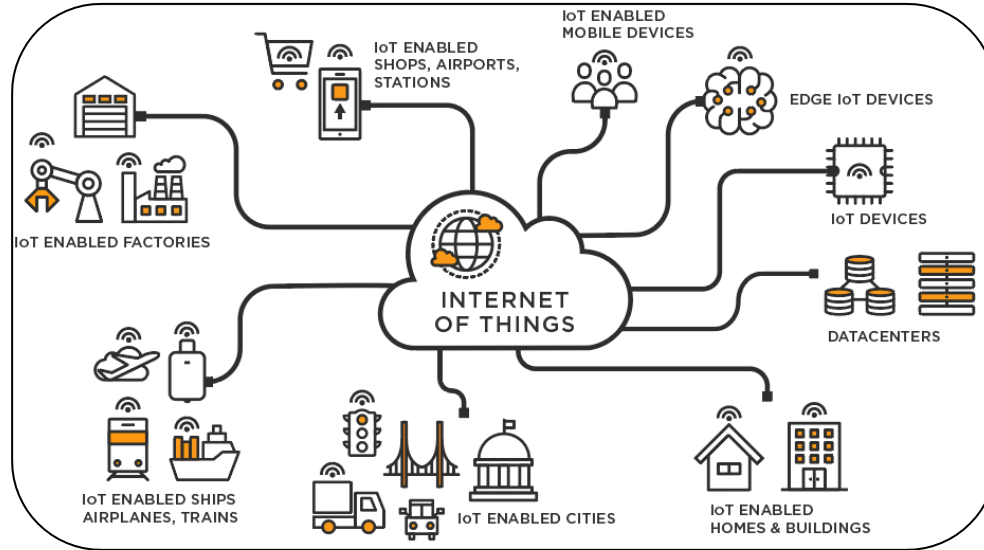


บทที่ 3 การประยุกต์ใช้ Arduino ควบคุมหลอด LED ด้วยสวิตช์



Last update :
19.12.2566

IOT: Internet of Things

Asst. Dr. Nattapong Songneam



กลุ่มที่ 6 ควบคุมความเร็วของพัดลมตามระดับอุณหภูมิ



8 คะแนน

เพิ่มเติม สิ่งที่แตกต่างกันจาก
โครงการอื่น ๆ

Temp + PM2.5

Temp + PM2.5 + ตรวจจับควัน แก๊ส

กลุ่มที่ 5 แขนกลหุ่นยนต์

The screenshot shows a Zoom meeting interface. The main window displays a presentation slide titled "Project robot arm". The slide content includes:

- รายวิชา (Subject)
- อินเทอร์เน็ตในทุกสิ่ง (4123412) (Internet of Everything (4123412))
- อาจารย์ผู้สอน (Instructor)
- ศส. ดร. นัฐพงศ์ ส่องเนียม (Mr. Dr. Nattapong Songniam)
- สาขา เทคโนโลยีสารสนเทศ คณะวิทยาศาสตร์และเทคโนโลยี (Faculty of Science and Technology, Department of Information Technology)
- มหาวิทยาลัยราชภัฏพระนคร (Rajabhat Mahachulalongkornrajavidyalaya University)

On the right side of the Zoom window, there is a vertical list of participants:

- Top: A small video feed of a participant.
- Second: A participant named "IT-3030".
- Third: A participant named "52_005 รุ่งสฤษฎ์ แฉมดวง".
- Bottom: A summary card showing a profile icon, the letter "A", and the text "อีก 2 คน" (2 more people).

At the top of the Zoom window, there is a status bar that says "คุณกำลังนำเสนอต่อทุกคน" (You are sharing this application window.) and a "Stop Sharing" button.

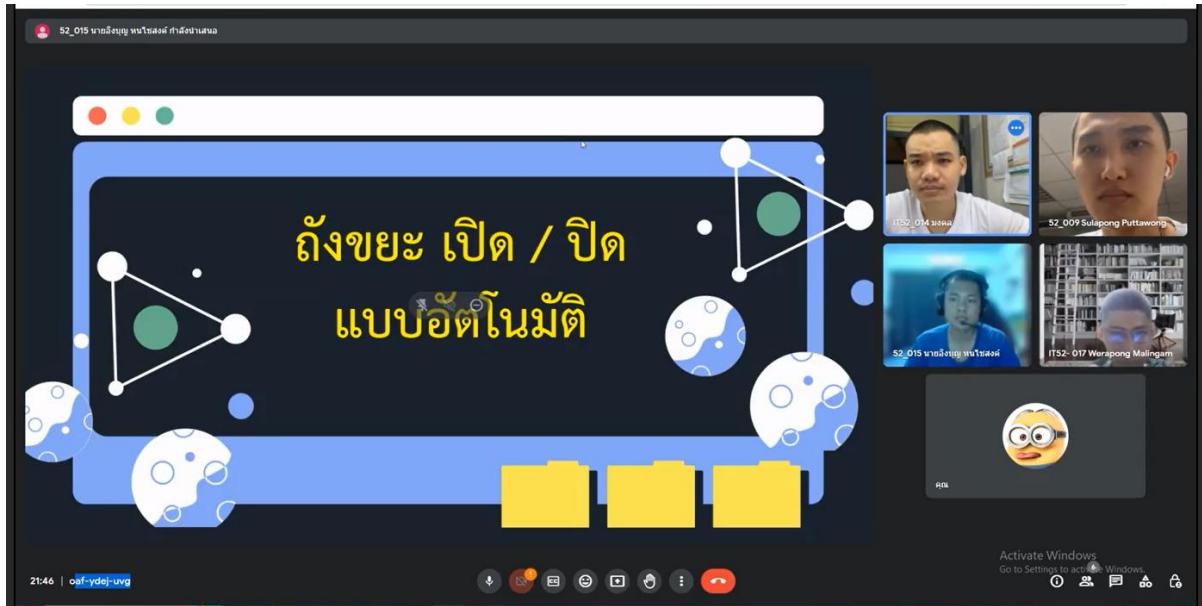
9 คะแนน

เพิ่มเติม สิ่ง
แตกต่างจาก
โครงการอื่น ๆ

Blynk

ฟังก์ชันการควบคุม เช่น ขึ้นลง หมุนซ้ายขวา ยกได้ก็ยกได้

กลุ่มที่ 4 ถึงขยะเปิดปิดอัตโนมัติ



8 คะแนน

เพิ่มเติม สิ่ง
แตกต่างจาก
โครงการอื่น ๆ

ทำถึงขยะ จากวัสดุเหลือใช้ หรือ อื่น ๆ

การแจ้งเตือนผ่านโทรศัพท์ + ปริมาณขยะ

กลุ่มที่ 3 ถึงขยะเปิดปิดอัตโนมัติ



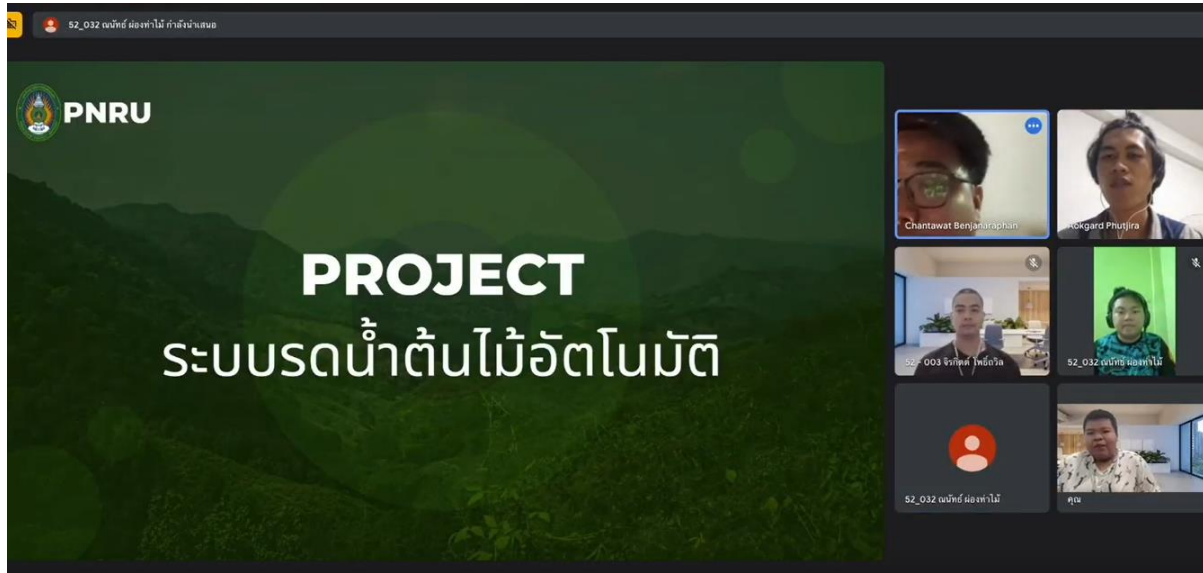
เพิ่มเติม สิ่งที่
แตกต่างจาก
โครงการอื่น ๆ

...

วัดระดับปริมาณของเหลวในขวด

7 คะแนน

กลุ่มที่ 3 ระบบปลูกผัก



7 คะแนน

เพิ่มเติม สิ่งที่
แตกต่างจาก
โครงการอื่น ๆ

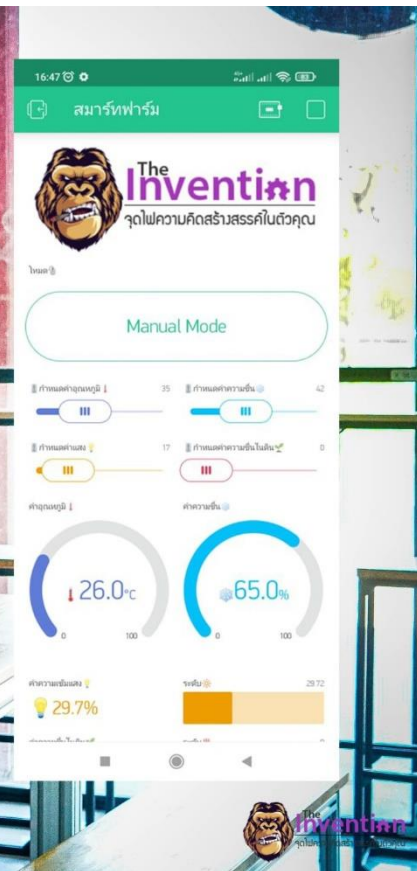
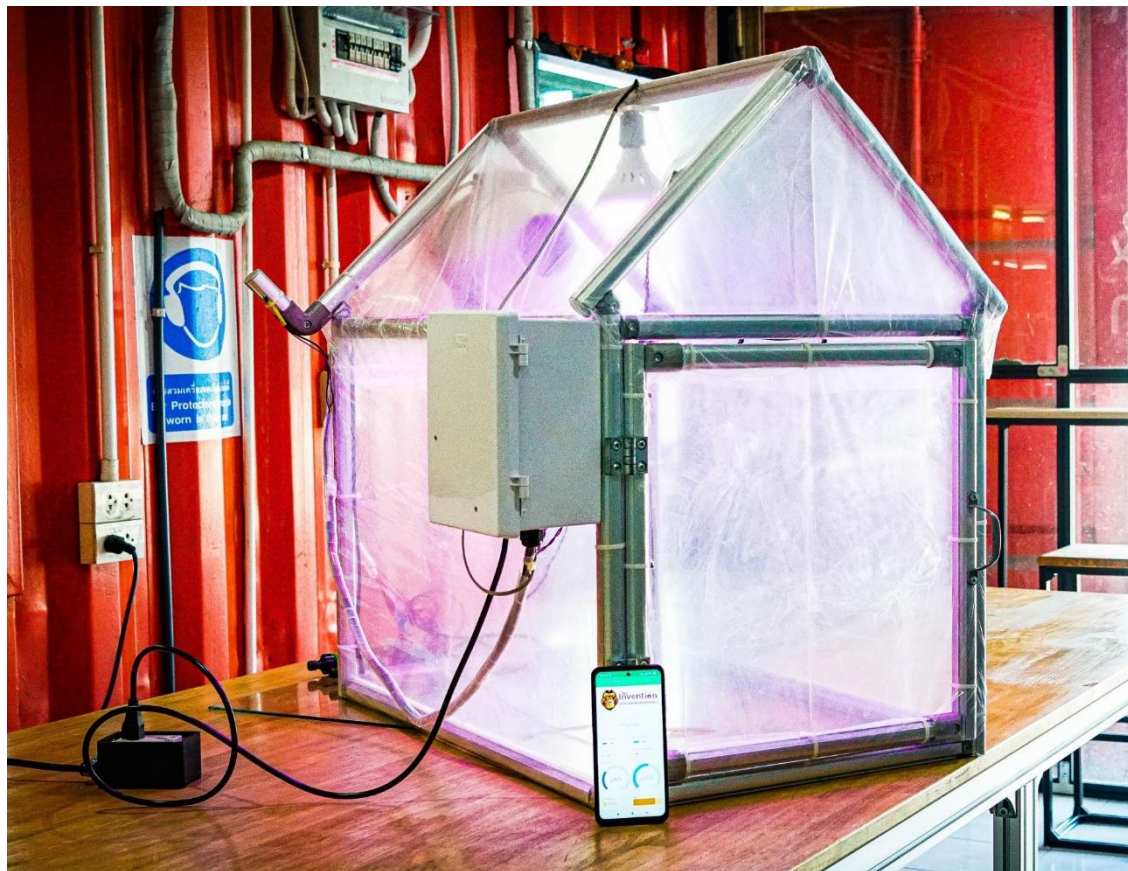
...

วัตถุดิบภาพดิน ผลิต



Smart Green House

โรงเรือนขนาดเล็ก 18 ตารางเมตร
สำหรับปลูกพืชในรั้วบ้าน



กรณีศึกษา

ปั๊มน้ำสร้างรอยยิ้ม WeWy





สาขาวิชาวิศวกรรมดิจิทัล ม.รังสิต

19 ชม. · 🌐

🌟 **ขอแสดงความยินดี** กับนักศึกษาสาขาวิชาวิศวกรรมดิจิทัล นำเสนอแผนธุรกิจกับคณะกรรมการจากธนาคารออมสิน ในโครงการ SMART STARTUP COMPANY การพัฒนาไปสู่การเป็นผู้ประกอบการ SME Startup ในอนาคตได้รับทุนสนับสนุนการจัดตั้งธุรกิจนวัตกรรมเริ่มต้นจำนวน 200,000 บาท 🙌🙌

📍 เรียนครบรอบใจหายด้านไอที และนวัตกรรมทันสมัย ที่ทุกองค์กรต้องการ

📅 สมัครและลงทะเบียนเรียนตั้งแต่วันที่ - 31 พ.ค. 66 รับทุน RSU Early Bird 5,000 บาท

🔥 เปิดรับสมัครนักศึกษาใหม่ ระดับปริญญาตรี ภาคการศึกษา S/2566

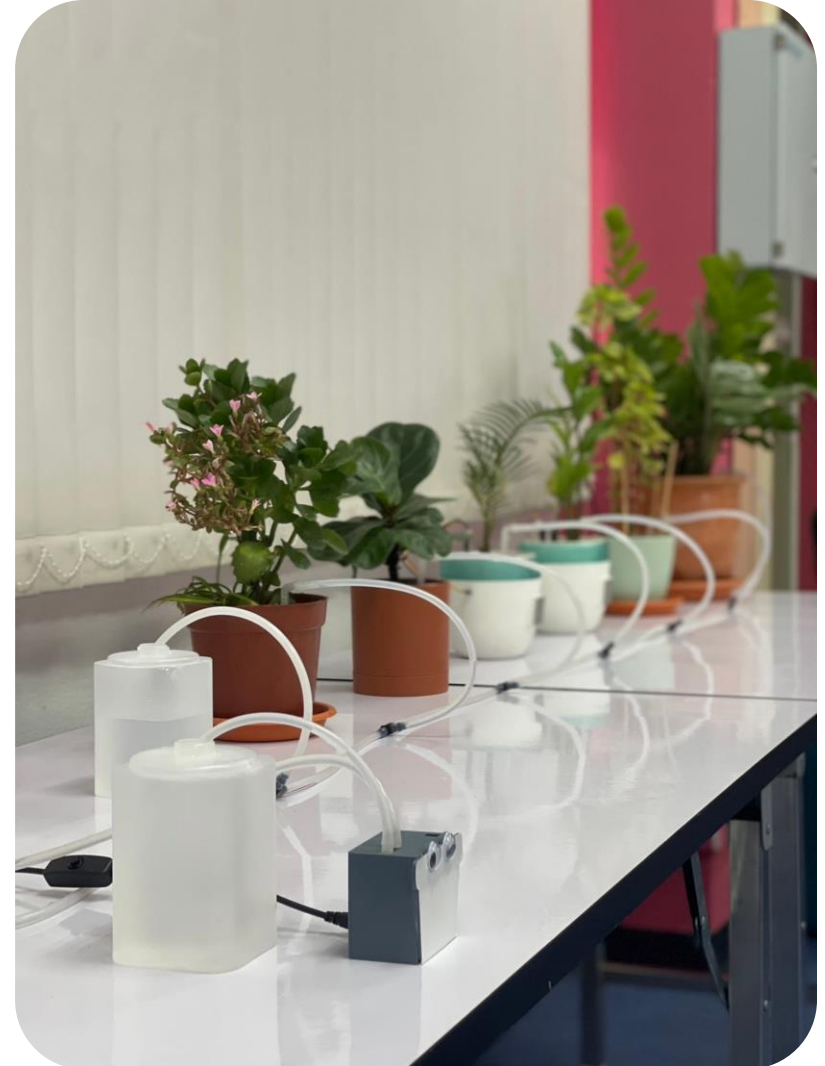
📞 สมัครผ่านระบบออนไลน์ : <https://admissiononline.rsu.ac.th>

📍 อย่าลืมนะบบสาขาวิชาที่สมัครเรียนเป็น "1402-สาขาวิชาวิศวกรรมดิจิทัล"

"1407-สาขาวิชาวิศวกรรมดิจิทัล" (ภาคพิเศษ)

📱 ทักไลน์ มาสอบถามข้อมูลเพิ่มเติมได้ที่ : @ITranssit

#นวัตกรรมดิจิทัล #DITE



ป๊มน้ำสร้างรอยยิ้ม WeWy



กลุ่มที่ 1 ระบบตรวจจับ

S Suwanan Wankaew is presenting.

ระบบตรวจจับ
แก๊สรั่ว
และระบายอากาศ
อัตโนมัติ



ประชุมออนไลน์

Pittinan Jaideejing

Suwanan Wankaew

Suwanan Wankaew

52_018_Ruechuta B...

ประชุมออนไลน์

Supachai lewch...

you

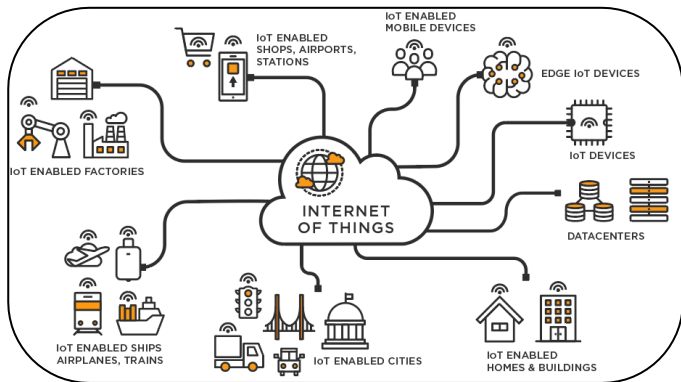
7 คะแนน

เพิ่มเติม สิ่งที่
แตกต่างจาก
โครงการอื่น ๆ

ถ้าจำลองไฟไหม้ในโรงพยาบาล แสดงเส้นทางหนีไฟ

??? + เสียงเตือน + สถานะไฟ

บทที่ 3 การประยุกต์ใช้ Arduino ควบคุมหลอด LED



เนื้อหา

1

อุปกรณ์พื้นฐานในการควบคุมหลอด LED

2

ประเภทสัญญาณ Digital และ Analog

3

การรับค่าจากปุ่ม / สวิตช์

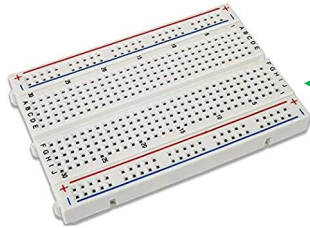
4

ตัวอย่างโปรแกรม

3.1 อุปกรณ์พื้นฐานในการควบคุมหลอด LED



บอร์ดอาดูยโน้ (Arduino Uno R3)



บอร์ดทดลอง (Bread Board)

720 pin

480 pin

ตัวต้านทาน (Resistor)

สายจับเปอร์ (Jumper Pair)

สวิตช์กด (Switch)

หลอด Diode LED

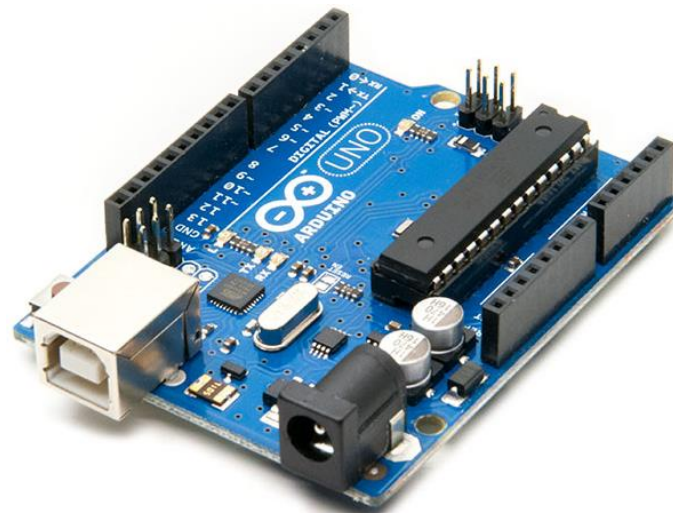


3.1 อุปกรณ์พื้นฐานในการควบคุมหลอด LED

3.1.1 Arduino Uno R3

อุปกรณ์หลักที่ใช้ในการพัฒนาระบบด้วยแนวคิด IoT เป็นเหมือนจุดที่ใช้ในการจ่ายไฟ หรือส่งต่อข้อมูลไปยังอุปกรณ์ต่าง ๆ โดยวงจรจะสามารถจะต้องมีแหล่งจ่ายไฟ ซึ่งแหล่งจ่ายไฟพื้นฐานได้แก่

- GND ย่อมาจาก GROUND คือ จุดต่ำสุดศูนย์รวม แรงดันอ้างอิง 0 V โดยอุปกรณ์ภายในวงจร จะมีกระแสไหลมาครบรอบที่กราวด์ ถูกเรียกว่า ครบวงจร
- VIN คือ ขั้วต่อแหล่งจ่ายภายนอก ที่เข้าบอร์ด จากกรณีของ Node MCU เราใช้ USB อะแดปเตอร์เสียบจ่ายไฟเข้าวงจร ดังนั้น vin ก็คือ แรงดัน (Voltage) ที่ได้จาก USB 5V
- VCC หรือ VDD คือ แรงดันที่ปรับค่าแล้ว ให้เหมาะสมกับวงจรนั้น ๆ ซึ่งกรณี NodeMCU เราใช้ USB อะแดปเตอร์เสียบจ่ายไฟ 5V จากนั้นในวงจรจะมีการปรับลดแรงดันลงเหลือ 3.3 v



ภาพที่ 3.1 บอร์ด Arduino Uno R3

3.1 อุปกรณ์พื้นฐานในการควบคุมหลอด LED

3.1.2 หลอด LED

LED ย่อมาจาก Light Emitting Diode เป็นหลอดชนิดหนึ่งที่สามารถเปล่งแสงได้เมื่อจ่ายไฟในปริมาณที่เหมาะสมที่ 2 ขา คือ แอโนด (Anode) และ แคโทด (Cathode) โดย แอโนดจะต่อกับไฟ ขั้วบวก สังเกตได้จาก แอโนดจะมีขาที่ยาว ส่วนแคโทดจะต่อกับ GND และมีขาที่สั้นกว่า



ภาพที่ 3.2 หลอด LED

3.1 อุปกรณ์พื้นฐานในการควบคุมหลอด LED

3.1.3 ตัวต้านทาน

เป็นอุปกรณ์ไฟฟ้าชนิดหนึ่งที่มีคุณสมบัติในการต้านการไหลผ่านของกระแสไฟฟ้า ทำด้วยลวดต้านทานหรือถ่านคาร์บอน เป็นต้น นั่นคือ ถ้าอุปกรณ์นั้นมีความต้านทานมาก กระแสไฟฟ้าที่ไหลผ่านจะน้อยลง เป็นอุปกรณ์ไฟฟ้าชนิดพาสซีฟสองขั้ว ที่สร้างความต่างศักย์ไฟฟ้าคร่อมขั้วทั้งสอง (V) โดยมีสัดส่วนมากน้อยตามปริมาณกระแสไฟฟ้าที่ไหลผ่าน (I) อัตราส่วนระหว่างความต่างศักย์ และปริมาณกระแสไฟฟ้า ก็คือ ค่าความต้านทานทางไฟฟ้า หรือค่าความต้านทานของตัวนำมีหน่วยเป็นโอห์ม (สัญลักษณ์ : Ω)

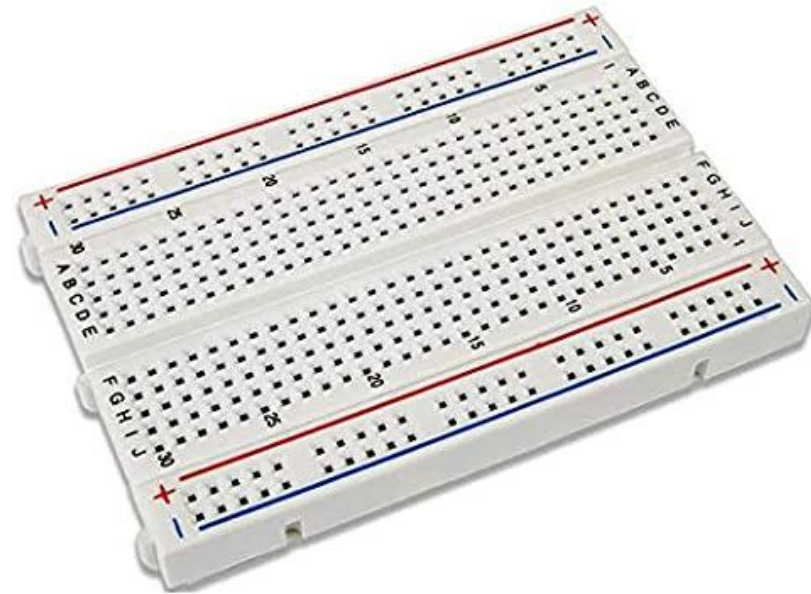


ภาพที่ 3.3 ตัวต้านทาน (Resistor)

3.1 อุปกรณ์พื้นฐานในการควบคุมหลอด LED

3.1.4 บอร์ดทดลอง

บอร์ดที่ใช้ทดลองวงจรอิเล็กทรอนิกส์ ลักษณะเป็นแผ่นพลาสติกหนาสีขาว บนแผ่นมีรูเรียงกันจำนวนมาก ภายในรูมีตัวนำไฟฟ้าซึ่งเชื่อมต่อกันในรูปแบบที่มีการกำหนดไว้ เวลาทดลองก็เสียบขาของอุปกรณ์อิเล็กทรอนิกส์ลงไปให้ตัวนำภายในเชื่อมวงจรถึงกัน และอาจใช้สายไฟเสียบลงรูเพื่อเชื่อมวงจรไฟฟ้าได้เช่นกัน ข้อดีของโปรโตบอร์ดคือ ไม่ต้องออกแบบแผงวงจรและไม่ต้องบัดกรี แต่มีข้อเสียคือ ใช้ทดลองวงจรที่ทำงานที่ความถี่สูง ๆ ไม่ได้เนื่องจากมีปัญหาเรื่องสัญญาณรบกวนในวงจร



ภาพที่ 3.4 บอร์ดทดลอง (Bread Board)

3.1 อุปกรณ์พื้นฐานในการควบคุมหลอด LED

3.1.5 สวิตช์

เป็นอุปกรณ์ควบคุมประเภทเปิดปิด On - off ทำหน้าที่ในการเชื่อมต่อวงจรให้กระแสไฟฟ้าไหลเข้าสู่วงจร หรือทำหน้าที่ในการตัดกระแสไฟไม่ให้เข้าสู่วงจร



ภาพที่ 3.5 สวิตช์

3.1 อุปกรณ์พื้นฐานในการควบคุมหลอด LED

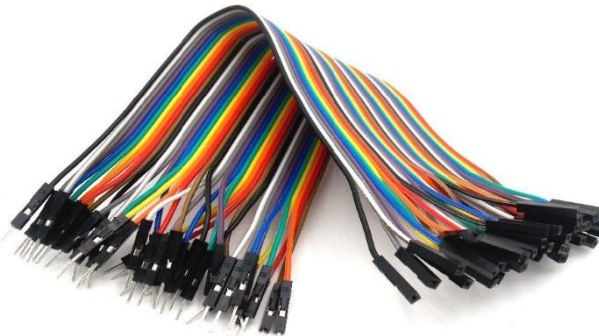
3.1.6 สายต่อ Jumper

ใช้สำหรับเสียบเข้ากับ Arduino และ บอร์ดทดลองเพื่อใช้ในการเชื่อมต่อวงจร ปลายหัวที่ใช้เสียบแบ่งออกเป็น ผู้ และ เมีย โดยผู้จะมีปลายแหลม ส่วนเมีย จะปลายเป็นรูเสียบ ซึ่งได้แบ่งสายจัมเปอร์ ออกเป็น 3 แบบ ดังนี้

ภาพที่ 3.6 สายจัมเปอร์ (Jumper)



ผู้ - ผู้



ผู้ - เมีย



เมีย - เมีย

Digital signal



สัญญาณไม่ต่อเนื่อง

Analog signal



สัญญาณต่อเนื่อง

3.2 ประเภทสัญญาณ Digital และ Analog

3.2 ประเภทสัญญาณ Digital และ Analog

3.2.1 คำสั่ง digitalWrite

เป็นคำสั่งที่ใช้กำหนดสัญญาณ HIGH LOW ของขาดิจิตอลของ Arduino
HIGH คือลอจิก 1 ปลั๊กไฟออกแรงดัน 5V
LOW คือลอจิก 0 กำหนดขานั้นให้เป็นกราว 0V

digitalWrite(PiN,Status)

PiN หมายถึง ขา Digital ของ Arduino ที่จะสั่งงาน ให้เป็น HIGH หรือ LOW
Status หมายถึง สถานะ HIGH หรือ LOW

ตัวอย่าง

digitalWrite(3, HIGH) //ให้ขาที่ 3 ปลั๊กไฟออกแรงดัน 5V

digitalWrite(5, LOW) //ให้ขาที่ 5 ปลั๊กไฟ 0V

3.2 ประเภทสัญญาณ Digital และ Analog

3.2.2 คำสั่ง analogWrite

การส่งสัญญาณ Analog OutPut ของ Arduino โดยใช้สัญญาณ PWM 0-5v มาควบคุมความสว่างของหลอดไฟ LED ขาของ Arduino UNO ที่ส่งสัญญาณ PWM ได้จะมีขา 3 5 6 9 จะมีตัวหนอนอยู่ใกล้ๆ

รูปแบบของคำสั่ง: analogWrite

analogWrite(Pin,value)

Pin คือ หมายเลขขาหรือตัวแปรที่แทนขาของ Arduino

value คือ ค่าระดับสัญญาณ Analog Output มีค่าความละเอียด 8 บิต หรือ 256 มีค่าระหว่าง 0-255
ค่าสูงสุดจะให้แรงดันไฟฟ้ามากที่สุด

ตัวอย่างคำสั่ง analogWrite

analogWrite(9,255)

ต้องการให้ขา 9 ของ Arduino ส่งสัญญาณ PWM 255 ออกมา

3.3 การรับค่าจากปุ่ม / สวิตช์



3.3 การรับค่าจาก ปุ่ม / สวิตช์

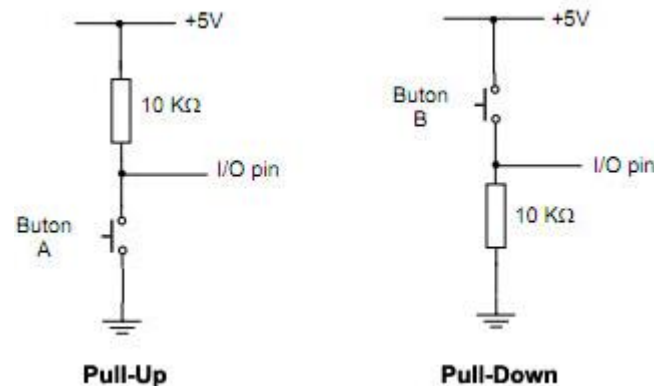
3.3.1 การใช้คำสั่ง digitalRead()

การรับค่าจากปุ่ม / สวิตช์ใช้งาน Arduino อ่านค่าการกดปุ่ม โดยใช้คำสั่ง digitalRead()

การต่อสวิตช์มี 2 แบบ คือ pull up โดยจ่ายสัญญาณ 1 ให้กับสวิตช์ อีกแบบคือ pull down โดยจ่ายสัญญาณ 0 ให้กับสวิตช์ โดยผ่านตัวต้านทานเพื่อใช้คงค่าสถานะ ป้องกันไฟ 5V และ 0V ชนกันลัดวงจร

การต่อสวิตช์มี 2 แบบคือ

1. Pull UP คือการคงค่าสัญญาณ 1 ให้กับขา Digital Arduino ที่มาต่อกับสวิตช์ เมื่อสวิตช์ถูกกดจะให้สัญญาณ 0
2. Pull Down คือ การคงค่าสัญญาณ 0 ให้กับขา Digital Arduino ที่มาต่อกับสวิตช์ เมื่อสวิตช์ถูกกดจะให้สัญญาณ 1



ภาพที่ 3.7 วงจร Pull-up และ Pull-down

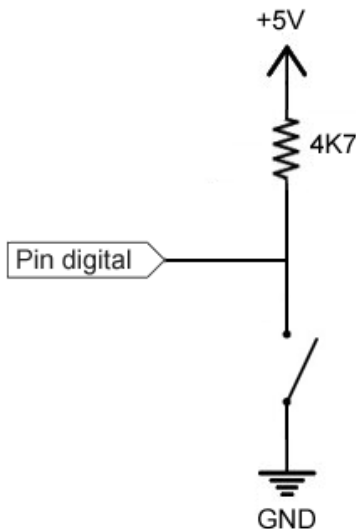
3.3 การรับค่าจาก ปุ่ม / สวิตช์

การต่อสวิตช์มี 2 แบบคือ

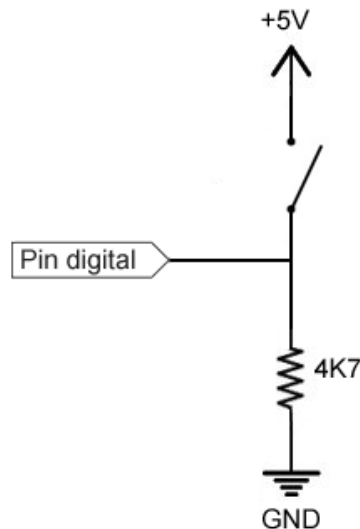
1. Pull UP คือการกำหนดสัญญาณ 1 ให้กับสวิตช์ตลอดเวลา เมื่อสวิตช์ถูกกดจะให้สัญญาณ 0
2. Pull Down คือ การกำหนดสัญญาณ 0 ให้กับสวิตช์ตลอดเวลา เมื่อสวิตช์ถูกกดจะให้สัญญาณ 1

ใช้ตัวต้านทานจ่ายไฟให้กับสวิตช์
ตัวอย่างดังรูป

RESISTENCIA PULL UP



RESISTENCIA PULL DOWN



ภาพที่ 3.8 การทำงานของวงจร Pull-up และ Pull-down

3.3 การรับค่าจาก ปุ่ม / สวิตช์

3.3.1 การใช้คำสั่ง digitalRead()

คำสั่ง Digital Read

เป็นคำสั่งที่ใช้อ่านค่าสถานะขาดิจิทัลอินพุตของ Arduino ว่าเป็น 5V หรือ 0V

HIGH คือลอจิก 1 ปล่อยไฟออกแรงดัน 5V

LOW คือลอจิก 0 กำหนดขานั้นให้เป็นกราวด์ 0V

digitalRead(PiN)

PiN หมายถึง ขา Digital ของ Arduino ที่ต้องการอ่าน

ตัวอย่างคำสั่ง Digital Read

ต้องการอ่านค่า สถานะขา 13 ว่าเป็น 1 หรือ 0 เก็บในตัวแปร Val

Val = digitalRead(13)

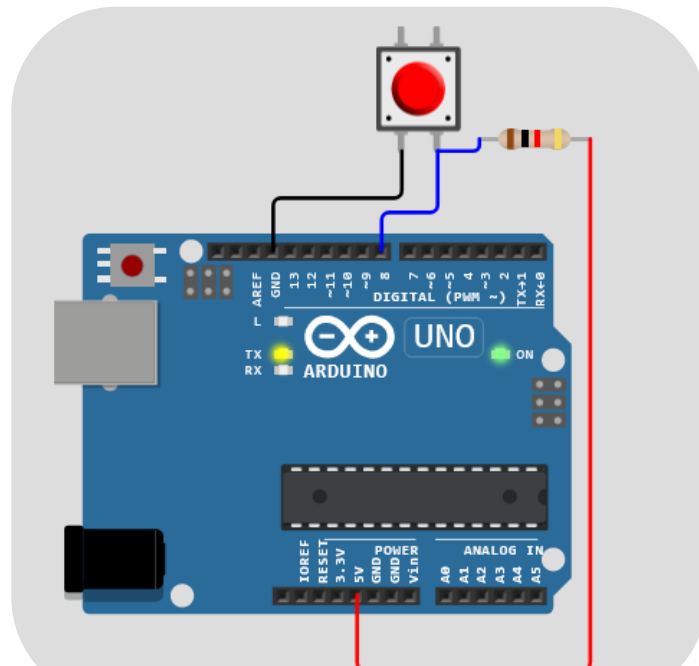
3.3 การรับค่าจาก ปุ่ม / สวิตช์

3.3.2 ตัวอย่างการใช้คำสั่ง digitalRead()

```
int sw = 8;
void setup() {
  Serial.begin(9600);
  pinMode(sw, INPUT);
}

void loop() {
  int val = digitalRead(sw);
  Serial.println(val);
}
```

<https://wokwi.com/projects/326676786507154003>



ภาพที่ 3.9 การต่อวงจรสำหรับทดสอบการทำงานของ digitalRead()

3.3 การรับค่าจาก ปุ่ม / สวิตช์

3.3.2 ตัวอย่างการใช้คำสั่ง digitalRead()

<https://wokwi.com/projects/326676786507154003>

```
int sw = 8;
void setup() {
  Serial.begin(9600);
  pinMode(sw, INPUT);
}
void loop() {
  int val = digitalRead(sw);
  Serial.println(val);
}
```

โค้ดนี้เขียนด้วยภาษาโปรแกรม Arduino และใช้งานกับบอร์ด Arduino หรือไมโครคอนโทรลเลอร์ที่เขียนโปรแกรมด้วย Arduino IDE ในที่นี้จะอธิบายโค้ดไปยังบรรทัดที่ละบรรทัด:

- `int sw = 8;`: ประกาศตัวแปร `sw` และกำหนดค่าให้เป็น 8 ซึ่งหมายถึงขาที่ต่อสวิตช์ (switch) ของ Arduino ที่พอร์ตดิจิทัล 8.
- `void setup() {...}`: เป็นฟังก์ชันที่ทำงานครั้งเดียวที่เริ่มต้นโปรแกรม ในที่นี้กำหนดให้ Serial communication เริ่มต้นที่ 9600 bps และกำหนดให้ขาที่เชื่อมกับสวิตช์ (`sw`) เป็นขา Input.
- `void loop() {...}`: เป็นฟังก์ชันที่ทำงานวนลูปตลอดเวลาหลังจากที่ `setup()` ทำงานเสร็จสิ้น ในลูปนี้มีการอ่านค่าจากสวิตช์ที่ต่อที่ขาที่กำหนดไว้ (`sw`) โดยใช้ `digitalRead()` และนำค่าที่ได้มาแสดงผลทาง Serial Monitor ด้วย `Serial.println()`.
- `int val = digitalRead(sw);`: อ่านค่าจากสวิตช์ที่ต่อที่ขาที่กำหนดไว้ (`sw`) และเก็บค่านั้นไว้ในตัวแปร `val`.
- `Serial.println(val);`: แสดงค่าที่อ่านได้จากสวิตช์ทาง Serial Monitor ซึ่งสามารถเรียกดูได้จาก Arduino IDE.

ดังนั้น, โค้ดนี้ทำหน้าที่อ่านค่าจากสวิตช์ที่ต่อที่ขา 8 แล้วแสดงผลค่านั้นทาง Serial Monitor ทุกครั้งที่โปรแกรมทำการวนลูปใน `loop()`

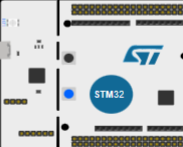
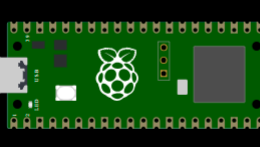
<https://wokwi.com/>



WOKWI
Simulate IoT Projects in Your Browser

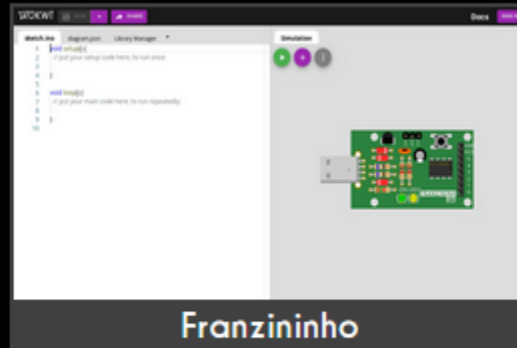
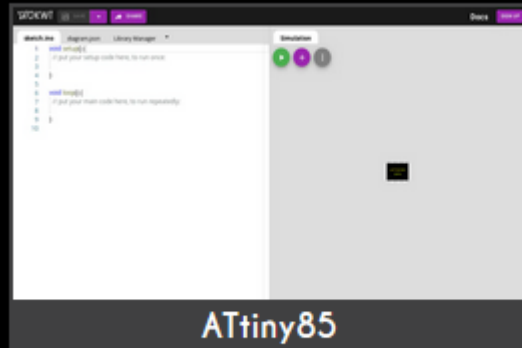
[Discord Community](#) [LinkedIn Group](#)

Simulate with Wokwi Online

 Arduino (Uno, Mega, Nano)	 ESP32
 STM32	 Pi Pico

Wokwi Pro

Start from Scratch



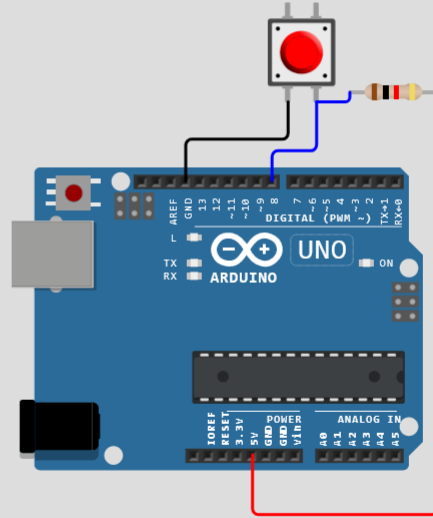
sketch.ino

diagram.json

Library Manager

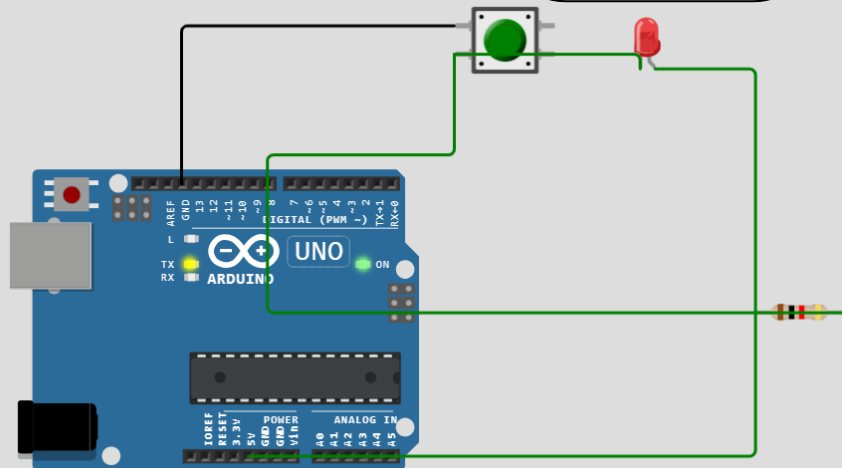
```
1 int sw = 8;
2 void setup() {
3   Serial.begin(9600);
4   pinMode(sw, INPUT);
5 }
6
7
8 void loop() {
9   int val = digitalRead(sw);
10  Serial.println(val);
11 }
12
```

Simulation

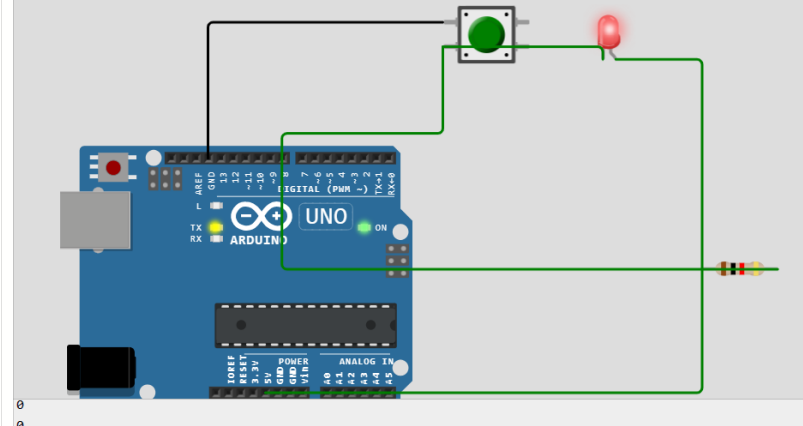
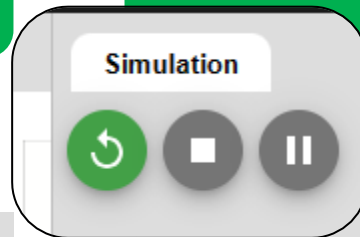


<https://wokwi.com/>

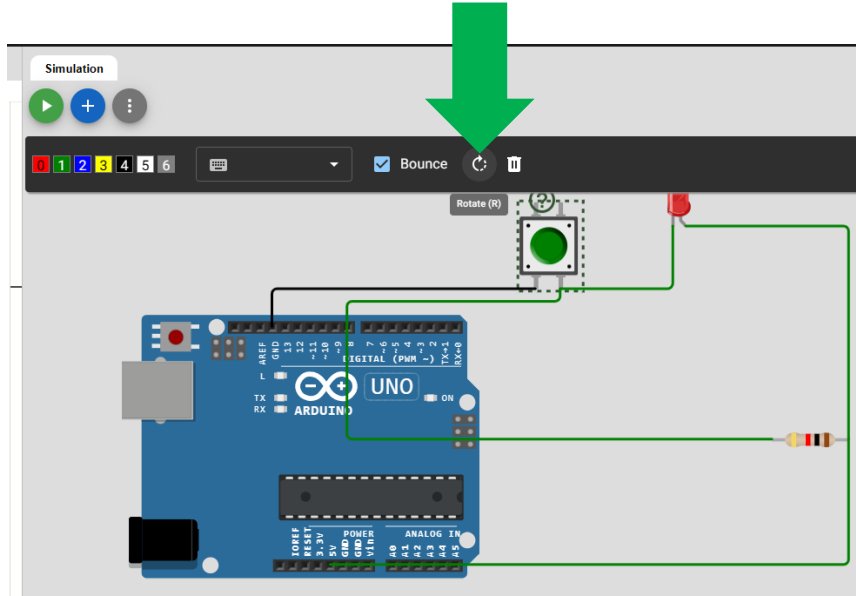
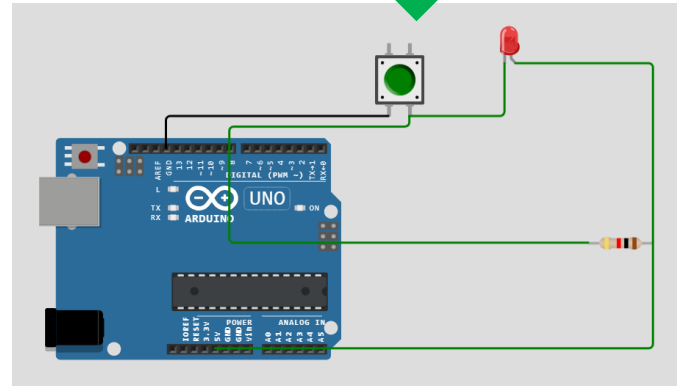
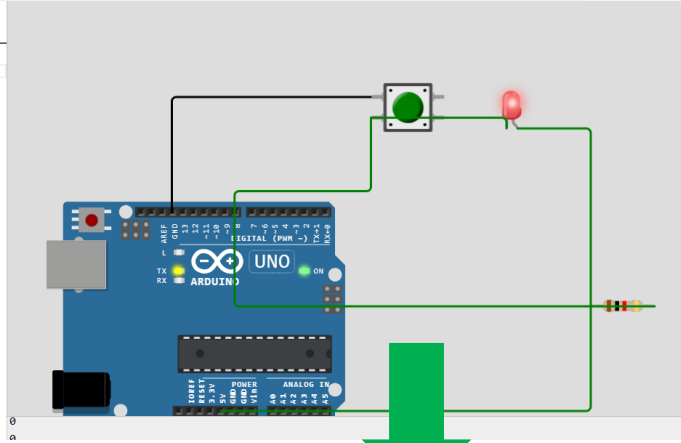
ไม่กดสวิตช์



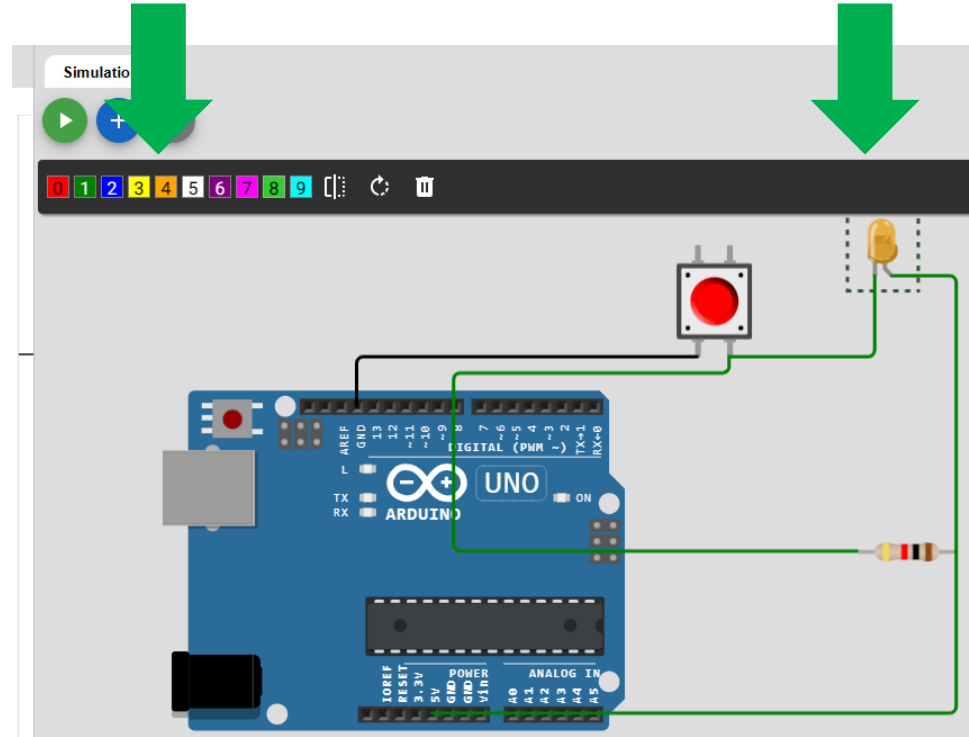
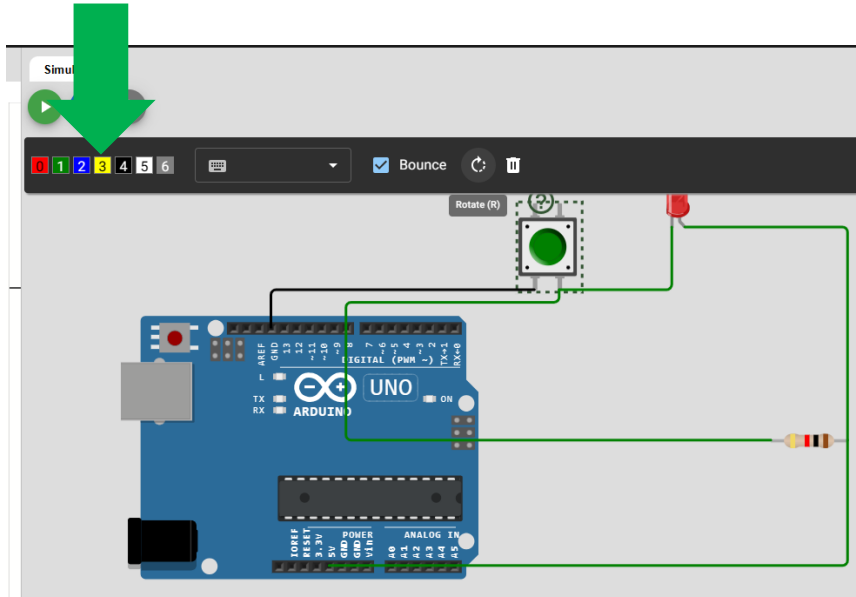
กดสวิตช์



Rotate : การหมุนตำแหน่งอุปกรณ์



การเลือกสีของอุปกรณ์

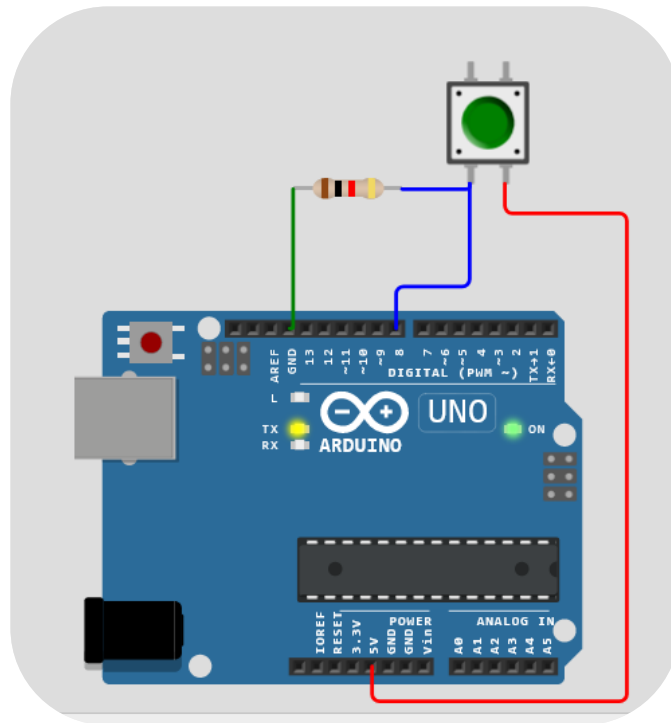


3.3 การรับค่าจาก ปุ่ม / สวิตช์

3.3.2 ตัวอย่างการใช้คำสั่ง digitalRead()

```
int sw = 8;  
void setup() {  
    Serial.begin(9600);  
    pinMode(sw, INPUT);  
}  
  
void loop() {  
    int val = digitalRead(sw);  
    Serial.println(val);  
}
```

<https://wokwi.com/projects/326677369267946067>

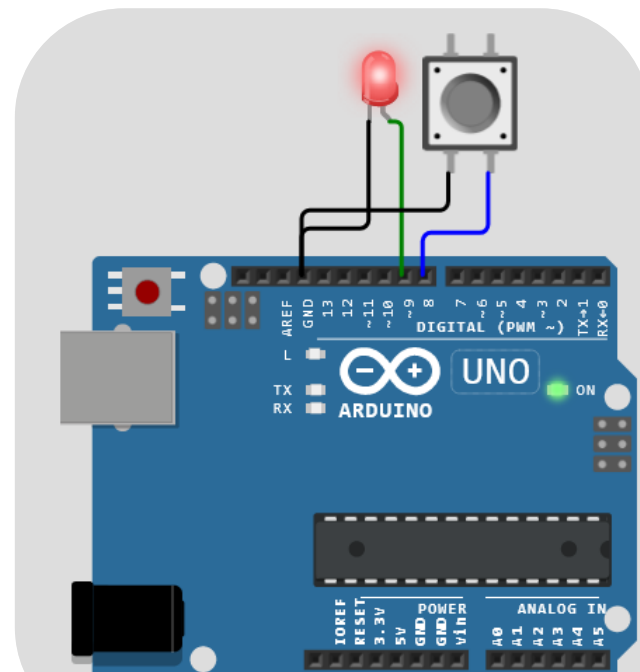


3.3 การรับค่าจาก ปุ่ม / สวิตช์

3.3.2 ตัวอย่างการใช้คำสั่ง digitalRead()

<https://wokwi.com/projects/326677824866878035>

```
int sw = 8;
int led = 9;
void setup() {
    Serial.begin(9600);
    pinMode(sw, INPUT_PULLUP);
    pinMode(led, OUTPUT);
}
void loop() {
    int val = digitalRead(sw);
    if(val==0){
        digitalWrite(led,1);
    }else{
        digitalWrite(led,0);
    }
    delay(500);
}
```



ภาพที่ 3.9 การต่อวงจรสำหรับทดสอบการทำงานของ digitalRead()

3.3 การรับค่าจาก ปุ่ม / สวิตช์

3.3.2 ตัวอย่างการใช้คำสั่ง digitalRead()

<https://wokwi.com/projects/326677824866878035>

```
int sw = 8;
int led = 9;
void setup() {
    Serial.begin(9600);
    pinMode(sw, INPUT_PULLUP);
    pinMode(led, OUTPUT);
}
void loop() {
    int val = digitalRead(sw);
    if(val==0){
        digitalWrite(led,1);
    }else{
        digitalWrite(led,0);
    }
    delay(500);
}
```

โค้ดนี้เป็นโค้ด Arduino ที่ใช้สวิตช์ (switch) ที่ต่อที่ขาที่ 8 เพื่อควบคุมไฟ LED ที่ต่อที่ขาที่ 9 โดยใช้การเขียนค่า HIGH หรือ LOW บนขาของ LED ตามสถานะของสวิตช์ โดยมีรายละเอียดดังนี้:

1. `int sw = 8;`: ประกาศตัวแปร `sw` และกำหนดค่าให้เป็น 8 ซึ่งหมายถึงขาที่ต่อสวิตช์ของ Arduino ที่พอร์ตดิจิทัล 8.
2. `int led = 9;`: ประกาศตัวแปร `led` และกำหนดค่าให้เป็น 9 ซึ่งหมายถึงขาที่ต่อ LED ของ Arduino ที่พอร์ตดิจิทัล 9.
3. `void setup() {...}`: เป็นฟังก์ชันที่ทำงานครั้งเดียวที่เริ่มต้นโปรแกรม ในที่นี้กำหนดให้ Serial communication เริ่มต้นที่ 9600 bps, กำหนดให้ขาที่ต่อสวิตช์เป็นขา Input และกำหนดให้ขาที่ต่อ LED เป็นขา Output.
4. `void loop() {...}`: เป็นฟังก์ชันที่ทำงานวนลูปตลอดเวลาหลังจากที่ `setup()` ทำงานเสร็จสิ้น ในลูปนี้มีการอ่านค่าจากสวิตช์ที่ต่อที่ขาที่กำหนดไว้ (`sw`) โดยใช้ `digitalRead()` และตรวจสอบค่า นั้น ถ้าค่าเป็น 0 (ต่อบวก), แสดงว่าสวิตช์ถูกกด จึงให้ไฟ LED ติดด้วยการเขียนค่า HIGH บนขาที่ต่อ LED (`led`) ถ้าค่าไม่ใช่ 0 แสดงว่าสวิตช์ไม่ถูกกด จึงให้ LED ดับด้วยการเขียนค่า LOW บนขาที่ต่อ LED. นอกจากนี้, มีการใส่ `delay(500);` เพื่อทำให้โปรแกรมหยุดทำงานเป็นเวลา 500 มิลลิวินาทีก่อนที่จะทำซ้ำลูปใหม่.



3.4 ตัวอย่างโปรแกรม เปิด-ปิด หลอดไฟ LED

3.4 ตัวอย่างโปรแกรม เปิด-ปิด หลอดไฟ LED

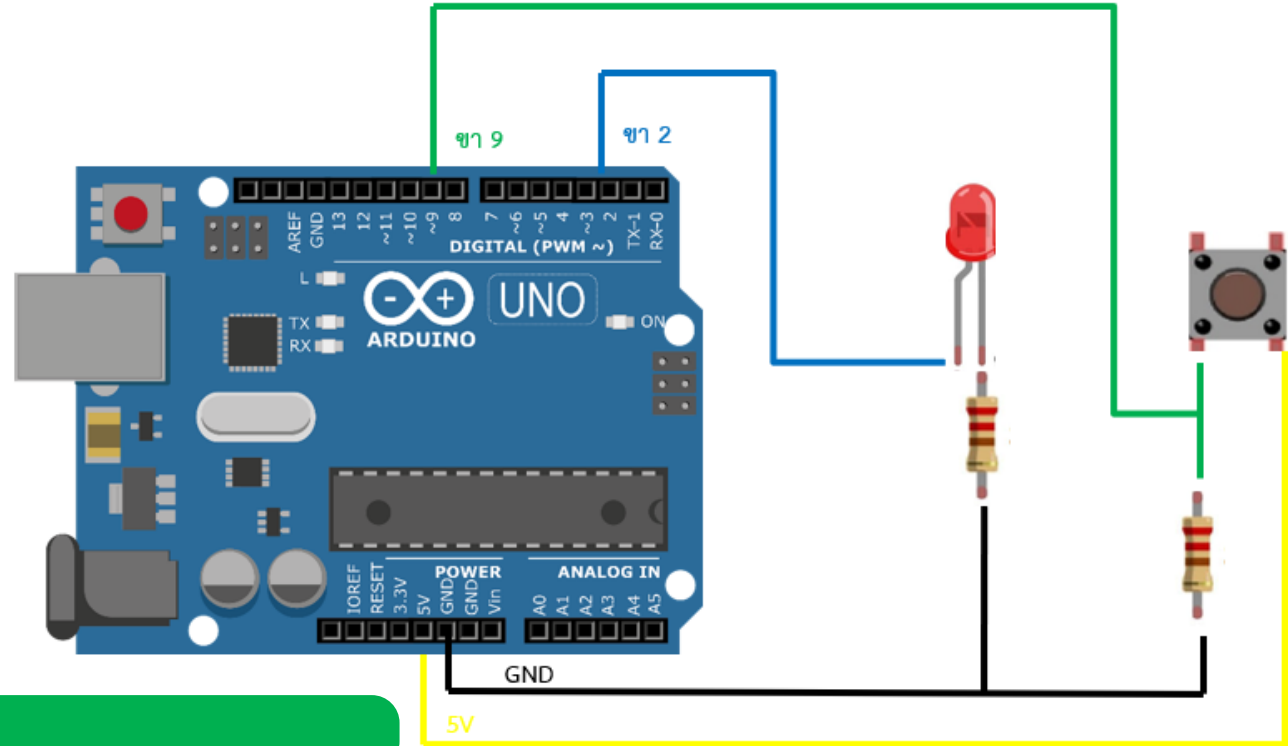
3.4.1 อุปกรณ์ที่ต้องใช้

- **Arduino UNO R3 พร้อม สาย USB**
- **LED ขนาด 5mm สีแดง 1 ดวง**
- **สวิตช์ เปิด - ปิด 1 อัน**
- **บอร์ดทดลอง Breadboard 830 Point**
- **Resistor ตัวต้านทาน 2 อัน**
- **สายไฟจัมเปอร์ ผุ้-ผุ้ 3 เส้น**

3.4 ตัวอย่างโปรแกรม เปิด-ปิด หลอดไฟ LED

3.4.2 วิธีการต่อวงจร

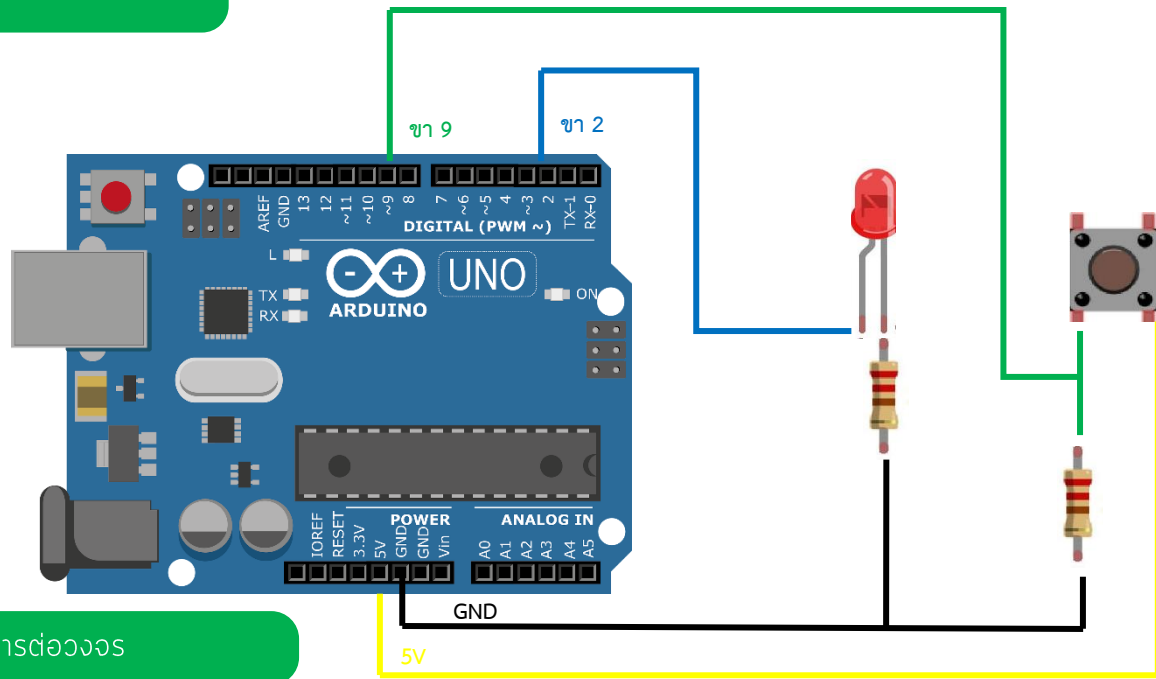
- ขา2 -> LED1
- ขา9 = ปุ่ม1



ภาพที่ 3.10 ภาพการต่อวงจร

3.4 ตัวอย่างโปรแกรม เปิด-ปิด หลอดไฟ LED

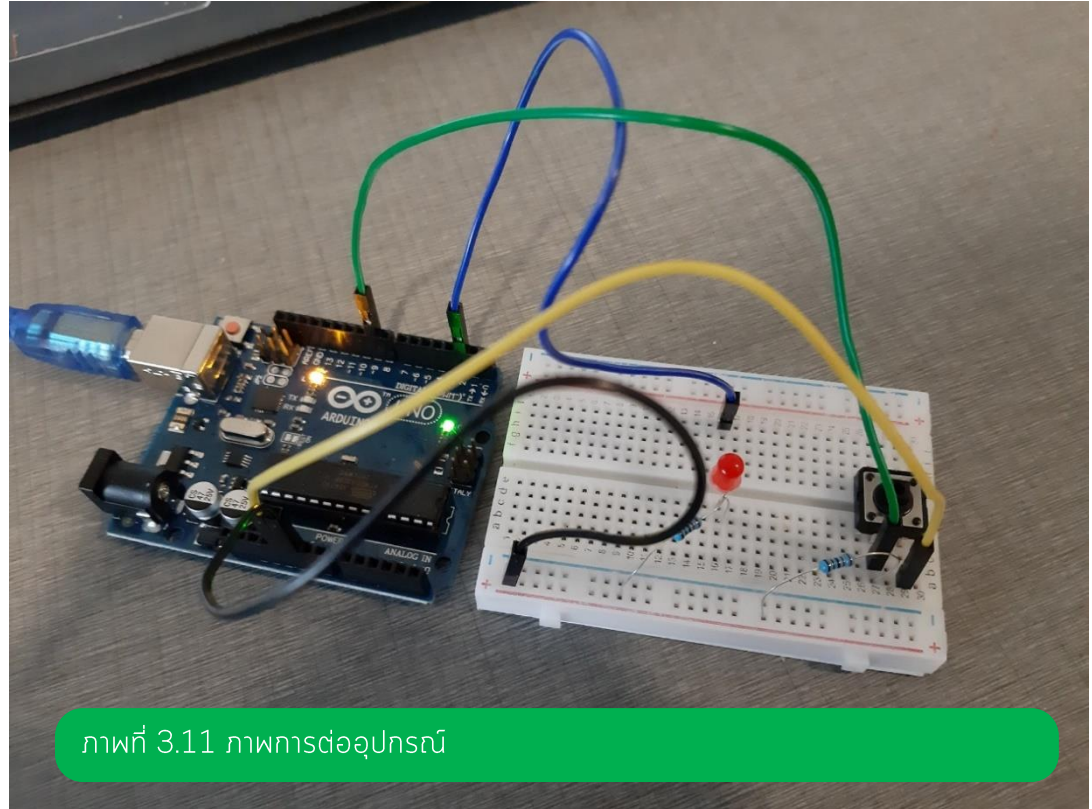
3.4.2 วิธีการต่อวงจร



ภาพที่ 3.10 ภาพการต่อวงจร

3.4 ตัวอย่างโปรแกรม เปิด-ปิด หลอดไฟ LED

3.4.3 ภาพการต่ออุปกรณ์



ภาพที่ 3.11 ภาพการต่ออุปกรณ์

3.4.1 ตัวอย่างโปรแกรม เปิด-ปิด หลอดไฟ LED

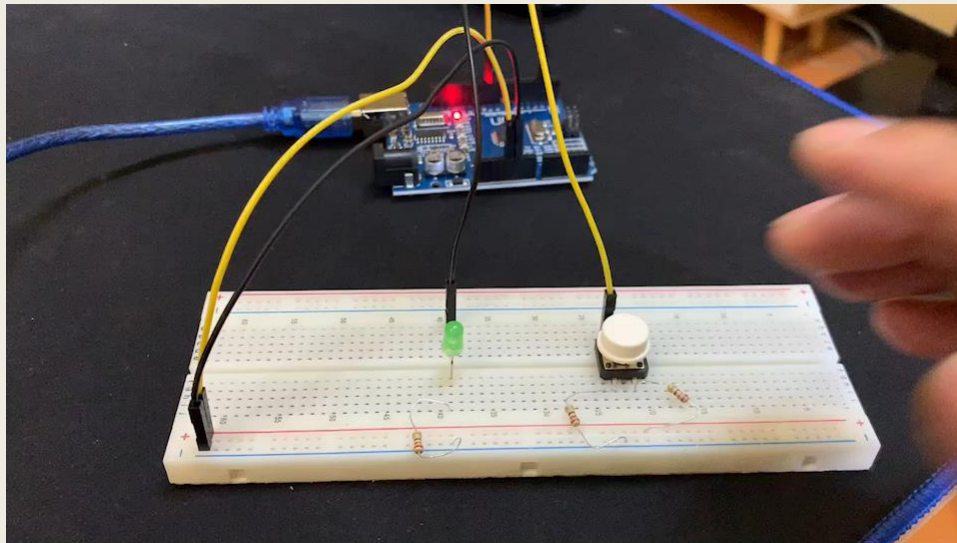
3.4.4 Code โปรแกรม

```
sketch_dec12a §
int led = 2;
int btnLight = 9;
int btnStatus = 0;
void setup() {
  pinMode(led, OUTPUT);          // ส่งค่าออกไปยังหลอด LED
  pinMode(btnStatus, INPUT);     // รับค่าปุ่มกดสวิตช์
}
void loop() {
  btnStatus = digitalRead(btnLight);
  if (btnStatus == HIGH) {
    digitalWrite(led, HIGH);
  } else {
    digitalWrite(led, LOW);
  }
}
```

```
int led = 2;
int btnLight = 9;
int btnStatus = 0;
void setup() {
  pinMode(led, OUTPUT);    // ส่งค่าออกไปยังหลอด LED
  pinMode(btnStatus, INPUT); // รับค่าปุ่มกดสวิตช์
}
void loop() {
  btnStatus = digitalRead(btnLight);
  if (btnStatus == HIGH){
    digitalWrite(led, HIGH);
  } else {
    digitalWrite(led, LOW);
  }
}
```

3.4.1 ตัวอย่างโปรแกรม เปิด-ปิด หลอดไฟ LED

3.4.5 หน้าจอการเขียนโปรแกรม



ภาพที่ 3.12 ภาพการเขียนโปรแกรมด้วย ArduinoIDE

```
test_led_03 | Arduino 1.8.12
File Edit Sketch Tools Help

test_led_03

int led = 2;
int btnLight = 9;
int btnStatus = 0;

void setup() {
  pinMode(led, OUTPUT);
  pinMode(btnStatus, INPUT);
}

void loop() {
  btnStatus = digitalRead(btnLight);
  if (btnStatus == HIGH) {
    digitalWrite(led, HIGH);
  } else {
    digitalWrite(led, LOW);
  }
}
```

Done uploading.

Sketch uses 940 bytes (2%) of program storage space. Maximum available is 32256 bytes. Global variables use 11 bytes (0%) of dynamic memory, leaving 2048 bytes free. Maximum allowed is 2048 bytes.

17 Arduino Uno on COM4

3.5 ตัวอย่างโปรแกรม เปิด-ปิด หลอดไฟ LED 3 ตำแหน่ง

3.5 ตัวอย่างโปรแกรม เปิด-ปิด หลอดไฟ LED 3 ตำแหน่ง

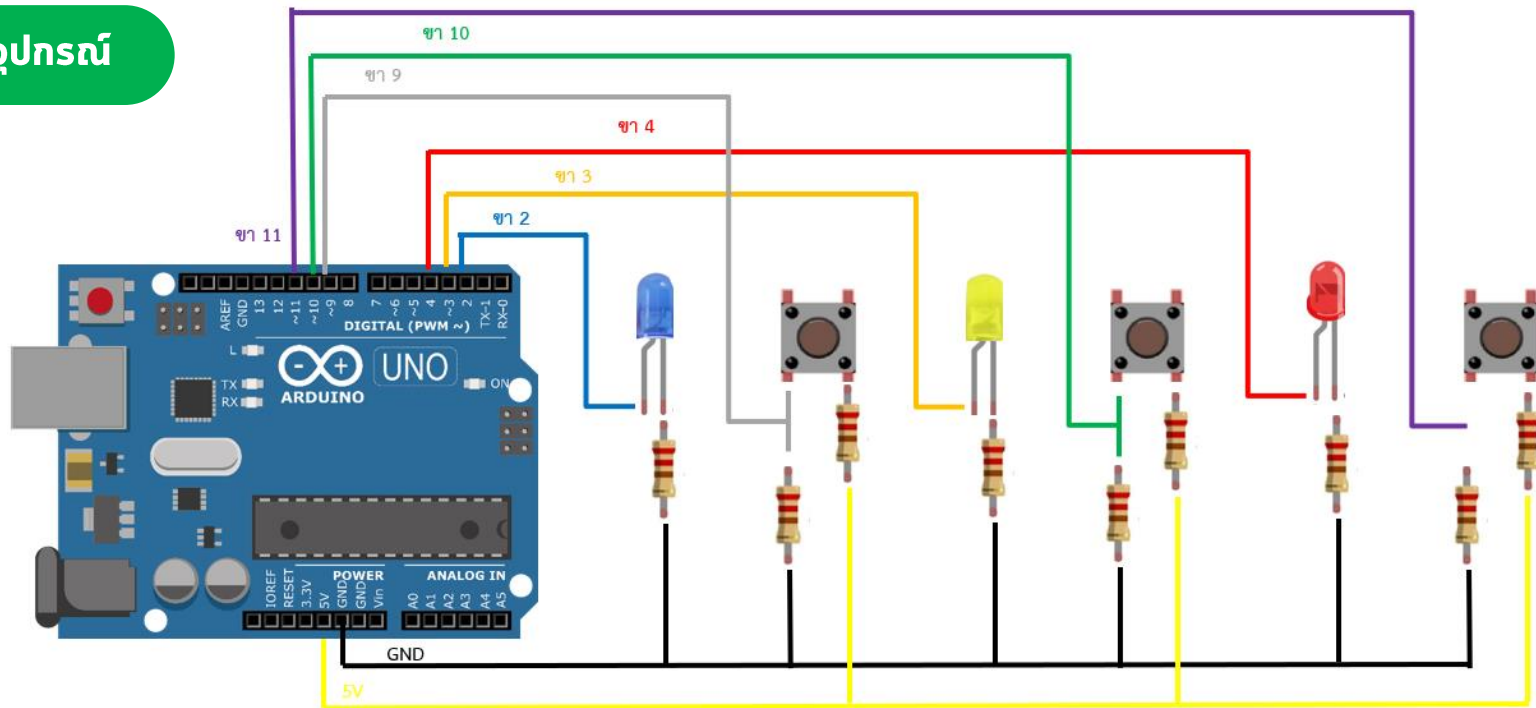
3.5.1 อุปกรณ์ที่ต้องใช้

- Arduino UNO R3 พร้อม สายUSB
- LED ขนาด 5mm สีน้ำเงิน สีเหลือง สีแดง อย่างละ 1 ดวง รวม 3 ดวง
- สวิตช์ เปิด - ปิด 3 อัน
- บอร์ดทดลอง Breadboard 830 Point
- Resistor ตัวต้านทาน 9 อัน
- สายไฟจัมเปอร์ ผุ้-ผุ้ 8 เส้น

3.5 ตัวอย่างโปรแกรม เปิด-ปิด หลอดไฟ LED 3 ตำแหน่ง

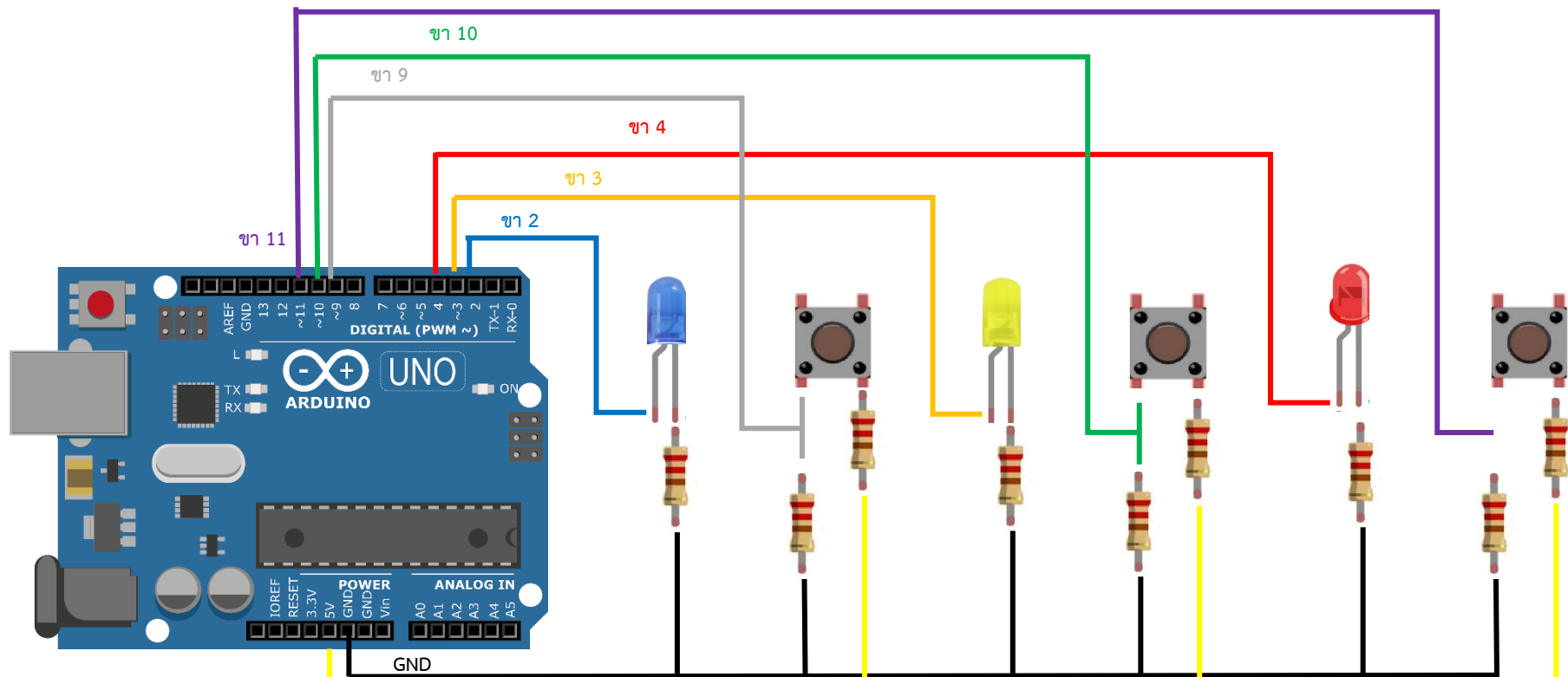
3.5.2 วิธีการต่ออุปกรณ์

- ขา2 = LED1
- ขา3 = LED2
- ขา4 = LED3
- ขา9 = ปุ่ม1
- ขา10 = ปุ่ม2
- ขา11 = ปุ่ม3



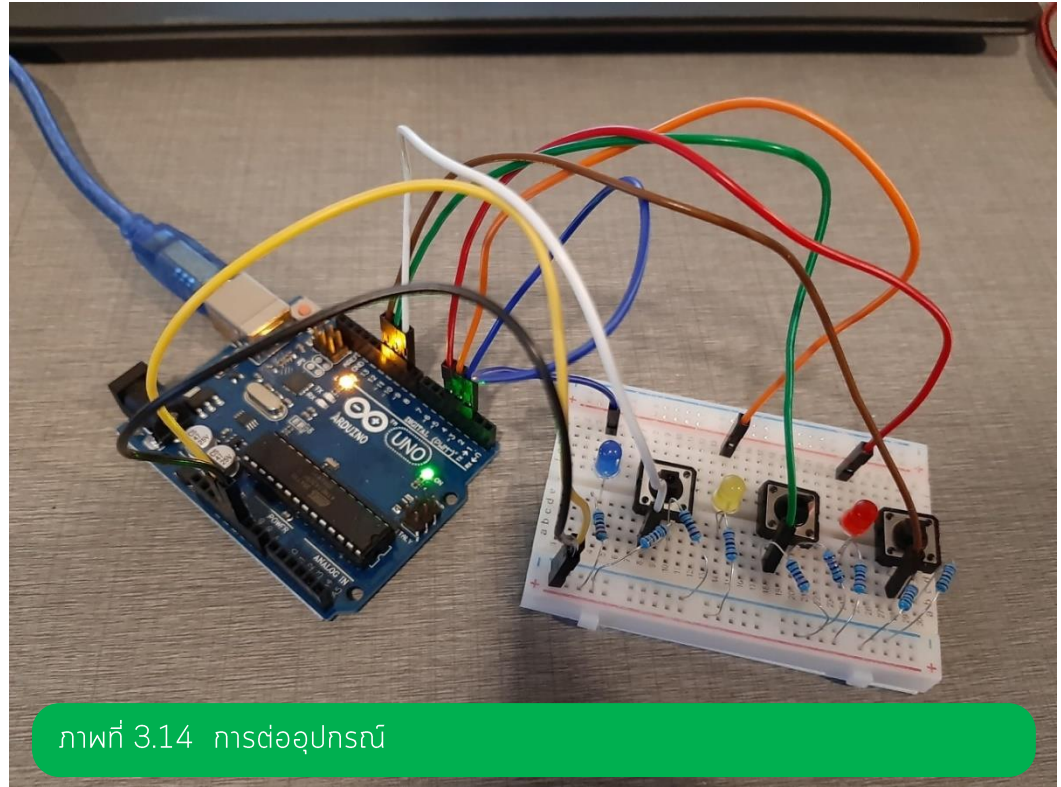
ภาพที่ 3.13 การต่อวงจร

3.5 ตัวอย่างโปรแกรม เปิด-ปิด หลอดไฟ LED 3 ตำแหน่ง



3.5 ตัวอย่างโปรแกรม เปิด-ปิด หลอดไฟ LED 3 ตำแหน่ง

3.5.3 ภาพการต่อวงจร



ภาพที่ 3.14 การต่ออุปกรณ์

3.5 ตัวอย่างโปรแกรม เปิด-ปิด หลอดไฟ LED 3 ตำแหน่ง

1

```
int ledBlue = 2;
int ledYellow = 3;
int ledRed = 4;
int btnLightBlue = 9;
int btnLightYellow = 10;
int btnLightRed = 11;
int btnStatusBlue = 0;
int btnStatusYellow = 0;
int btnStatusRed = 0;
void setup()
{
  pinMode(btnLightBlue, INPUT);
  pinMode(btnLightYellow, INPUT);
  pinMode(btnLightRed, INPUT);
  pinMode(ledBlue, OUTPUT);
  pinMode(ledYellow, OUTPUT);
  pinMode(ledRed, OUTPUT);
}
```

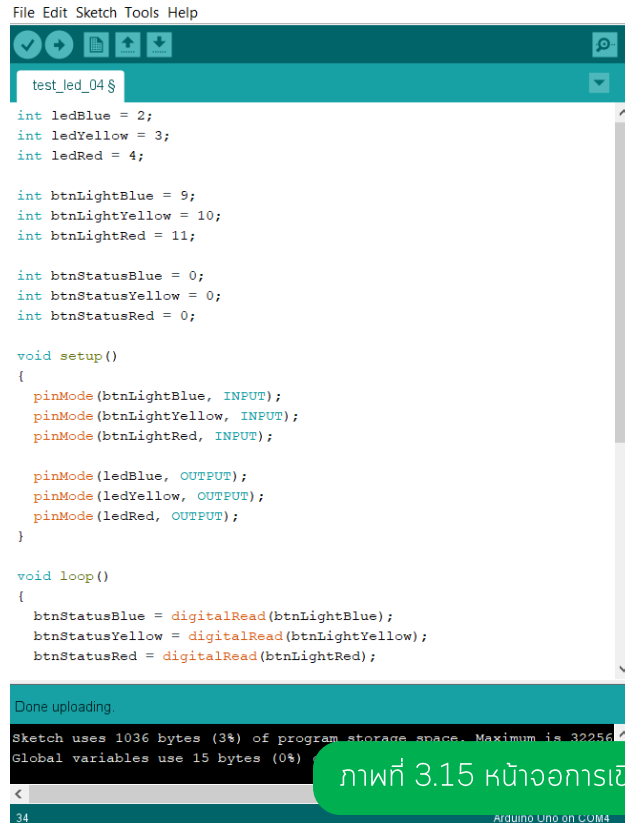
3.5.4 Code โปรแกรม

2

```
void loop()
{
  btnStatusBlue = digitalRead(btnLightBlue);
  btnStatusYellow = digitalRead(btnLightYellow);
  btnStatusRed = digitalRead(btnLightRed);
  if (btnStatusBlue == HIGH) {
    digitalWrite(ledBlue,HIGH);
  } else {
    digitalWrite(ledBlue,LOW);
  }
  if (btnStatusYellow == HIGH) {
    digitalWrite(ledYellow,HIGH);
  } else {
    digitalWrite(ledYellow,LOW);
  }
  if (btnStatusRed == HIGH) {
    digitalWrite(ledRed,HIGH);
  } else {
    digitalWrite(ledRed,LOW);
  }
}
```

3.5 ตัวอย่างโปรแกรม เปิด-ปิด หลอดไฟ LED 3 ตำแหน่ง

3.5.5 หน้าจอ การเขียน โปรแกรมด้วย Arduino IDE



```
File Edit Sketch Tools Help
test_led_04 §
int ledBlue = 2;
int ledYellow = 3;
int ledRed = 4;

int btnLightBlue = 9;
int btnLightYellow = 10;
int btnLightRed = 11;

int btnStatusBlue = 0;
int btnStatusYellow = 0;
int btnStatusRed = 0;

void setup()
{
  pinMode(btnLightBlue, INPUT);
  pinMode(btnLightYellow, INPUT);
  pinMode(btnLightRed, INPUT);

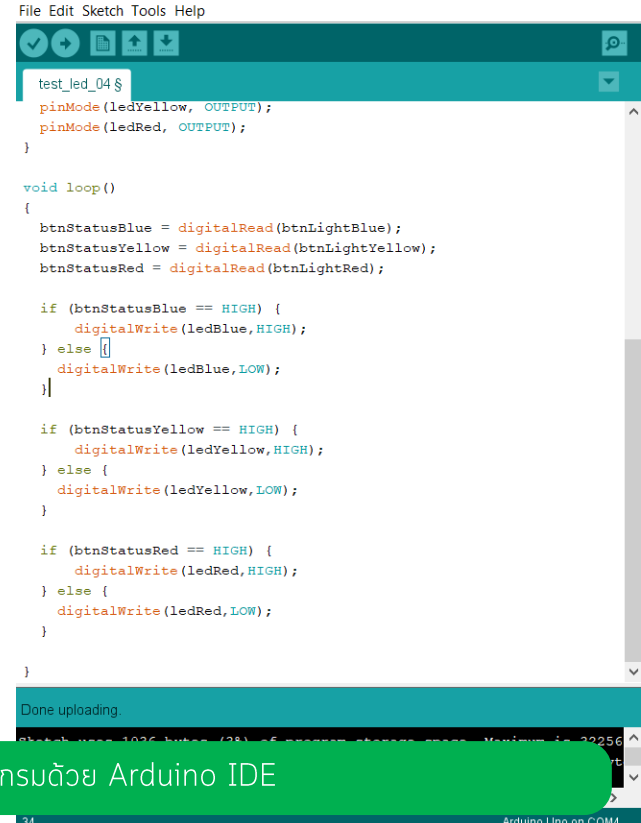
  pinMode(ledBlue, OUTPUT);
  pinMode(ledYellow, OUTPUT);
  pinMode(ledRed, OUTPUT);
}

void loop()
{
  btnStatusBlue = digitalRead(btnLightBlue);
  btnStatusYellow = digitalRead(btnLightYellow);
  btnStatusRed = digitalRead(btnLightRed);
}
```

Done uploading.

Sketch uses 1036 bytes (3%) of program storage space. Maximum is 32256.
Global variables use 15 bytes (0%).

34 Arduino Uno on COM4



```
File Edit Sketch Tools Help
test_led_04 §
pinMode(ledYellow, OUTPUT);
pinMode(ledRed, OUTPUT);
}

void loop()
{
  btnStatusBlue = digitalRead(btnLightBlue);
  btnStatusYellow = digitalRead(btnLightYellow);
  btnStatusRed = digitalRead(btnLightRed);

  if (btnStatusBlue == HIGH) {
    digitalWrite(ledBlue, HIGH);
  } else {
    digitalWrite(ledBlue, LOW);
  }

  if (btnStatusYellow == HIGH) {
    digitalWrite(ledYellow, HIGH);
  } else {
    digitalWrite(ledYellow, LOW);
  }

  if (btnStatusRed == HIGH) {
    digitalWrite(ledRed, HIGH);
  } else {
    digitalWrite(ledRed, LOW);
  }
}
```

Done uploading.

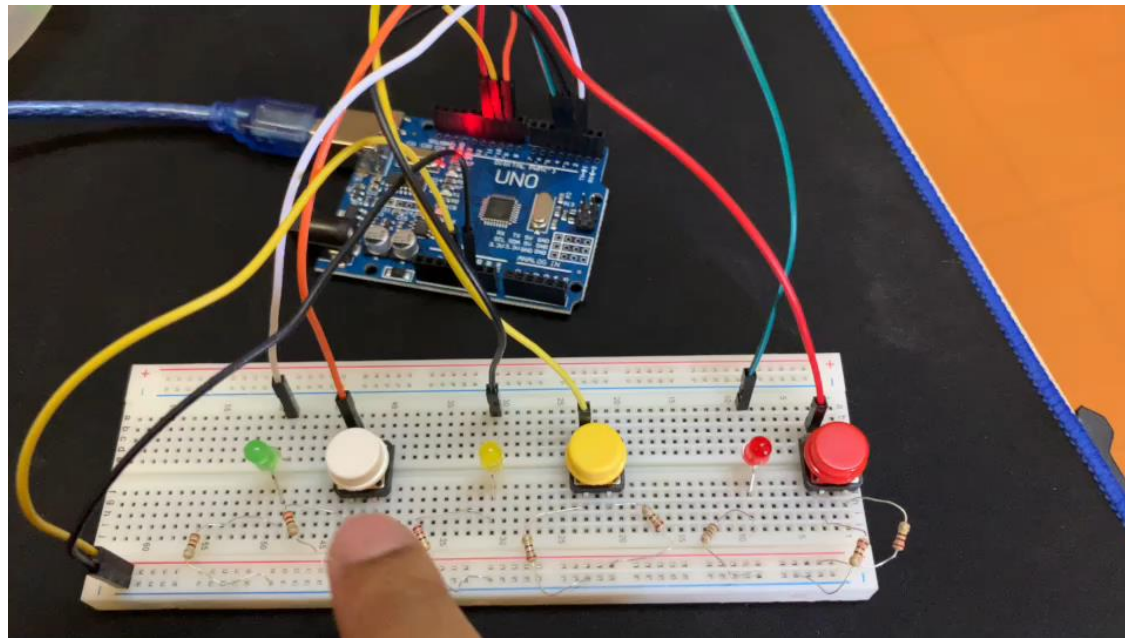
Sketch uses 1036 bytes (3%) of program storage space. Maximum is 32256.
Global variables use 15 bytes (0%).

34 Arduino Uno on COM4

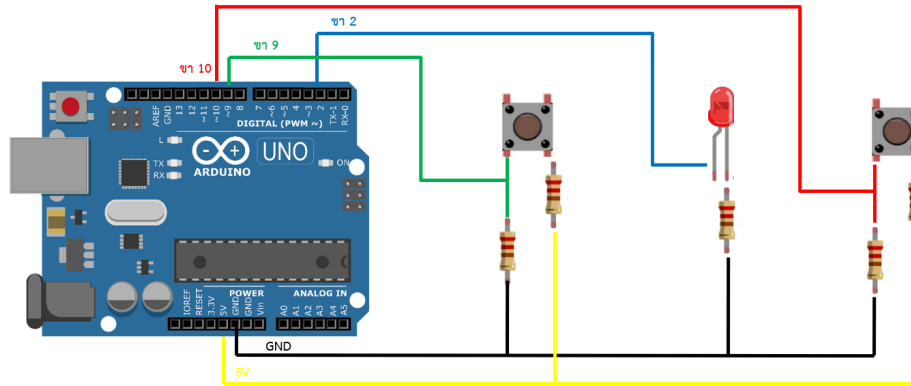
ภาพที่ 3.15 หน้าจอการเขียนโปรแกรมด้วย Arduino IDE

3.5 ตัวอย่างโปรแกรม เปิด-ปิด หลอดไฟ LED 3 ตำแหน่ง

3.5.5 หน้าจอ
การเขียน
โปรแกรมด้วย
Arduino IDE



ภาพที่ 3.15 วิธีควบคุมการทำงานของสวิตช์ทั้ง 3 ตัว



3.6 ตัวอย่างโปรแกรม เปิด-ปิด หลอดไฟ LED ด้วยสวิตช์ 2 ทิศทาง

3.6 ตัวอย่างโปรแกรม เปิด-ปิด หลอดไฟ LED ด้วยสวิตช์ 2 ทิศทาง

3.6.1 อุปกรณ์ที่ต้องใช้

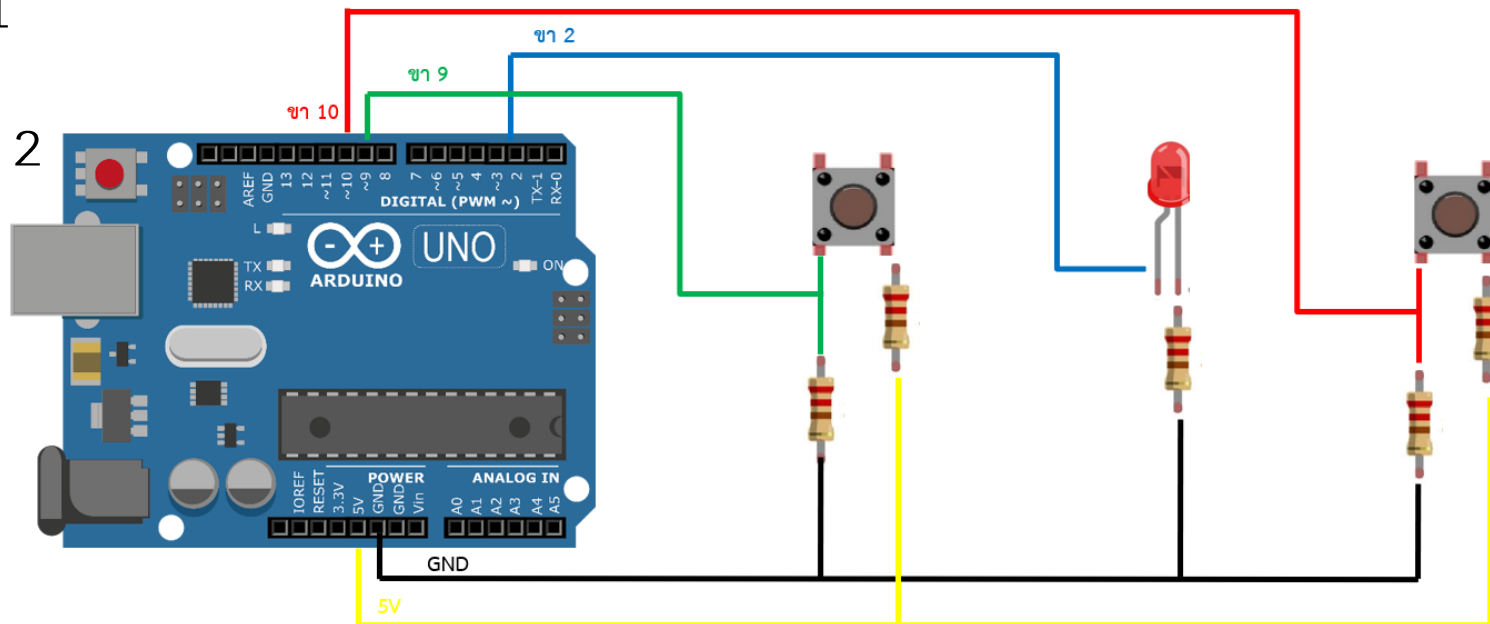
- **Arduino UNO R3 พร้อม สายUSB**
- **LED ขนาด 5mm สีแดง 1 ดวง**
- **สวิตช์ เปิด - ปิด 2 อัน**
- **บอร์ดทดลอง Breadboard 830 Point**
- **Resistor ตัวต้านทาน 5 อัน**
- **สายไฟจัมเปอร์ ผุ้-ผุ้ 5 เส้น**

3.6 ตัวอย่างโปรแกรม เปิด-ปิด หลอดไฟ LED ด้วยสวิตช์ 2 ทิศทาง

3.6.2 วิธีการต่อวงจร

ภาพที่ 3.16 วิธีการต่อวงจร

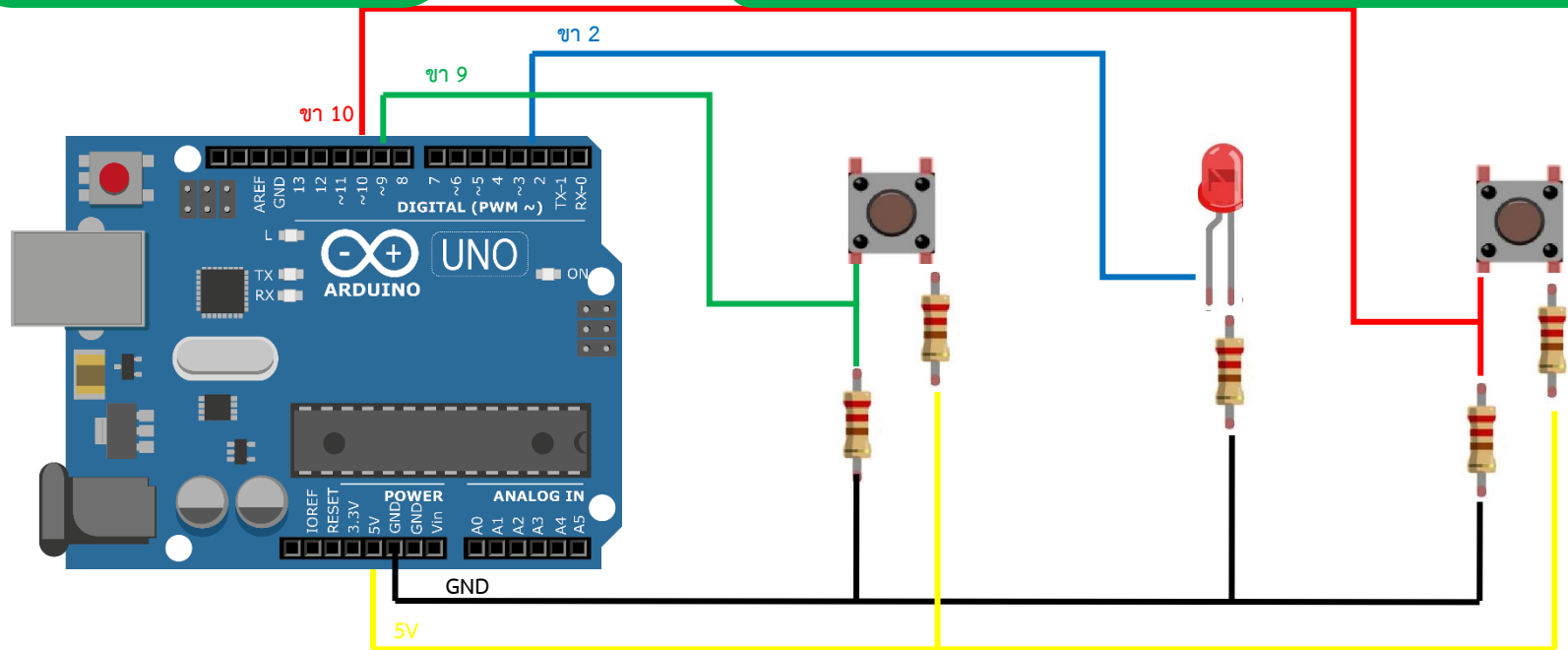
- ขา 2 = LED1
- ขา 9 = ปุ่ม 1
- ขา 10 = ปุ่ม 2



3.6 ตัวอย่างโปรแกรม เปิด-ปิด หลอดไฟ LED ด้วยสวิตช์ 2 ทิศทาง

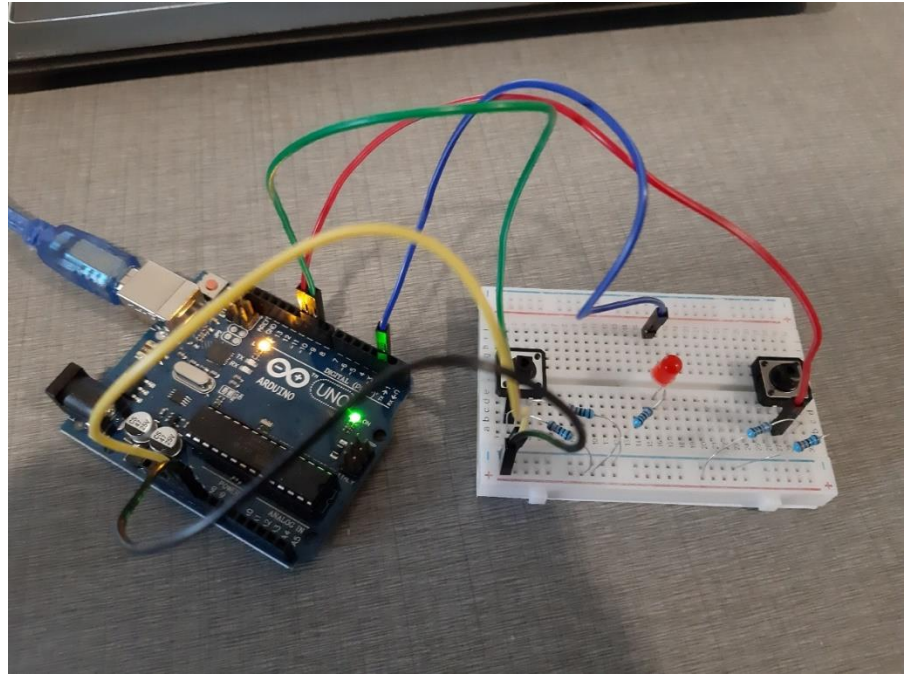
3.6.2 วิธีการต่อวงจร

ภาพที่ 3.16 วิธีการต่อวงจร



3.6 ตัวอย่างโปรแกรม เปิด-ปิด หลอดไฟ LED ด้วยสวิตช์ 2 ทิศทาง

3.6.3 ภาพการต่ออุปกรณ์



ภาพที่ 3.17 ภาพการต่อวงจร

3.6 ตัวอย่างโปรแกรม เปิด-ปิด หลอดไฟ LED ด้วยสวิตช์ 2 ทิศทาง

1

```
int led1 = 2;
int btnLight1 = 9;
int btnLight2 = 10;
int btnStatus1 = 0;
int btnStatus2 = 0;
void setup()
{
  pinMode(btnLight1, INPUT);
  pinMode(btnLight2, INPUT);
  pinMode(led1, OUTPUT);
}
```

2

```
void loop()
{
  btnStatus1 = digitalRead(btnLight1);
  btnStatus2 = digitalRead(btnLight2);
  if (btnStatus1 == HIGH) {
    digitalWrite(led1,HIGH);
  } else if (btnStatus2 == HIGH) {
    digitalWrite(led1,HIGH);
  } else {
    digitalWrite(led1,LOW);
  }
}
```

3.6.4 Code โปรแกรม

3.6 ตัวอย่างโปรแกรม เปิด-ปิด หลอดไฟ LED ด้วยสวิตช์ 2 ทิศทาง

3.6.5 หน้าจอการเขียนโปรแกรมด้วย Arduino IDE

ภาพที่ 3.18 หน้าจอการเขียนโปรแกรมด้วย Arduino IDE

ผศ.ดร. นัฐพงศ์ ส่งเนียม (<http://www.siam2dev.com>)



```
test_led_02 $
int led1 = 2;
int btnLight1 = 9;
int btnLight2 = 10;
int btnStatus1 = 0;
int btnStatus2 = 0;

void setup()
{
  pinMode(btnLight1, INPUT);
  pinMode(btnLight2, INPUT);
  pinMode(led1, OUTPUT);
}

void loop()
{
  btnStatus1 = digitalRead(btnLight1);
  btnStatus2 = digitalRead(btnLight2);
  if (btnStatus1 == HIGH) {
    digitalWrite(led1, HIGH);
  } else if (btnStatus2 == HIGH) {
    digitalWrite(led1, HIGH);
  } else {
    digitalWrite(led1, LOW);
  }
}
```

Done uploading.

Sketch uses 956 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 11 bytes (0%) of dynamic memory, leaving 2037 bytes free.

3.7 ตัวอย่างโปรแกรม การเปิดไฟวิ่งจาก ขวา-ซ้าย, ซ้าย-ขวา

3.7 ตัวอย่างโปรแกรม การเปิดไฟวิ่งจาก ขวา-ซ้าย, ซ้าย-ขวา

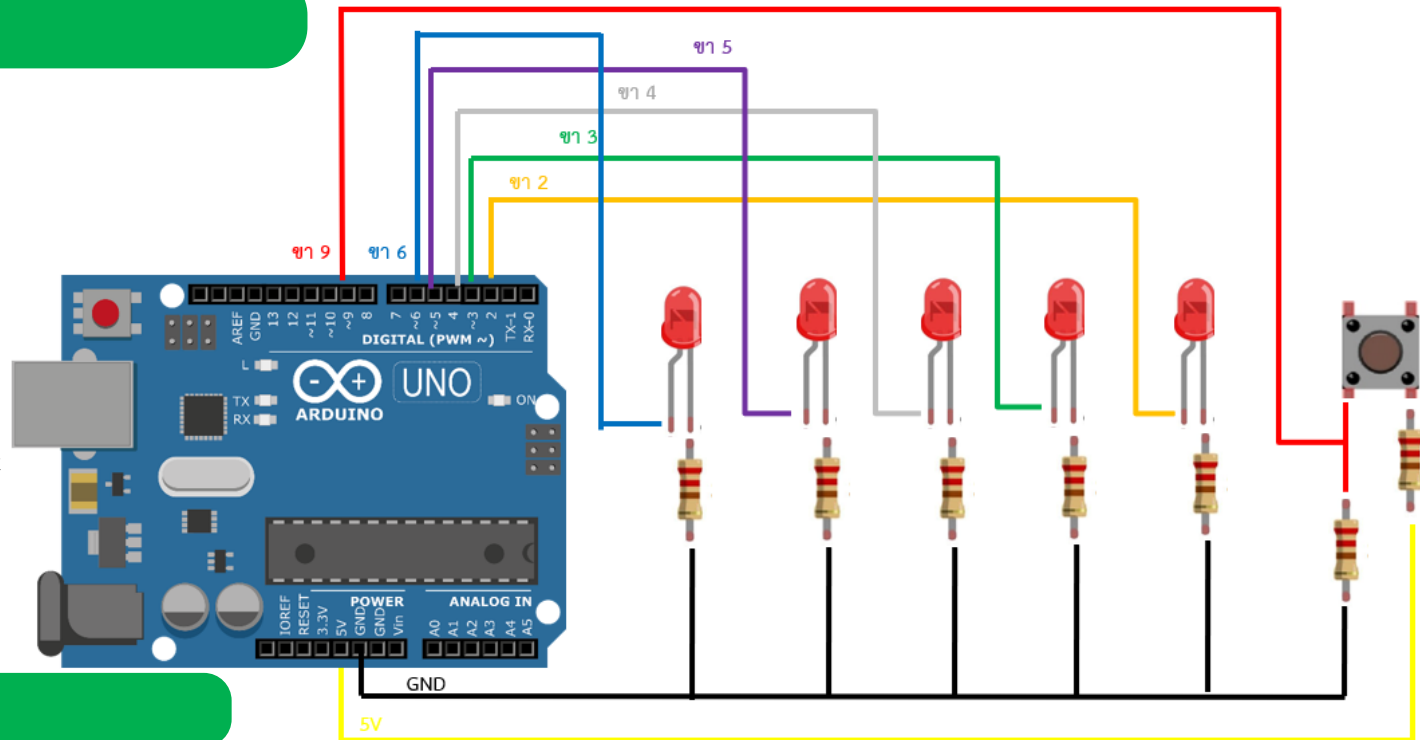
3.7.1 อุปกรณ์ที่ต้องใช้

- **Arduino UNO R3 พร้อม สายUSB**
- **LED ขนาด 5mm สีแดง 5 ดวง**
- **สวิตช์ เปิด - ปิด 1 อัน**
- **บอร์ดทดลอง Breadboard 830 Point**
- **Resistor ตัวต้านทาน 6 อัน**
- **สายไฟจัมเปอร์ ผุ้-ผุ้ 8 เส้น**

3.7 ตัวอย่างโปรแกรม การเปิดไฟวิ่งจาก ขวา-ซ้าย, ซ้าย-ขวา

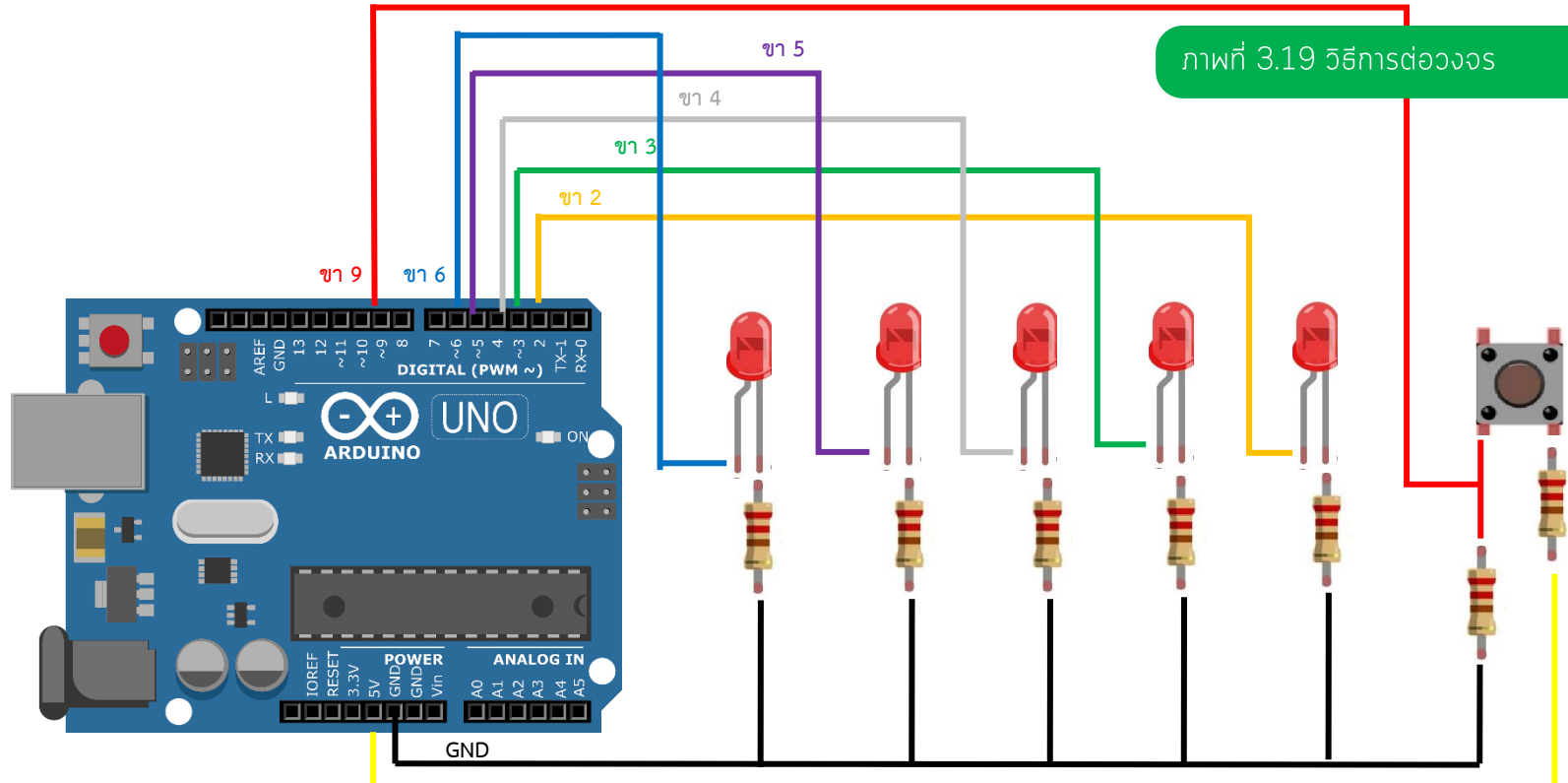
3.7.2 วิธีการต่อวงจร

- ขา2 = LED1
- ขา3 = LED2
- ขา4 = LED3
- ขา5 = LED4
- ขา6 = LED5
- ขา9 = ปุ่มสวิตช์



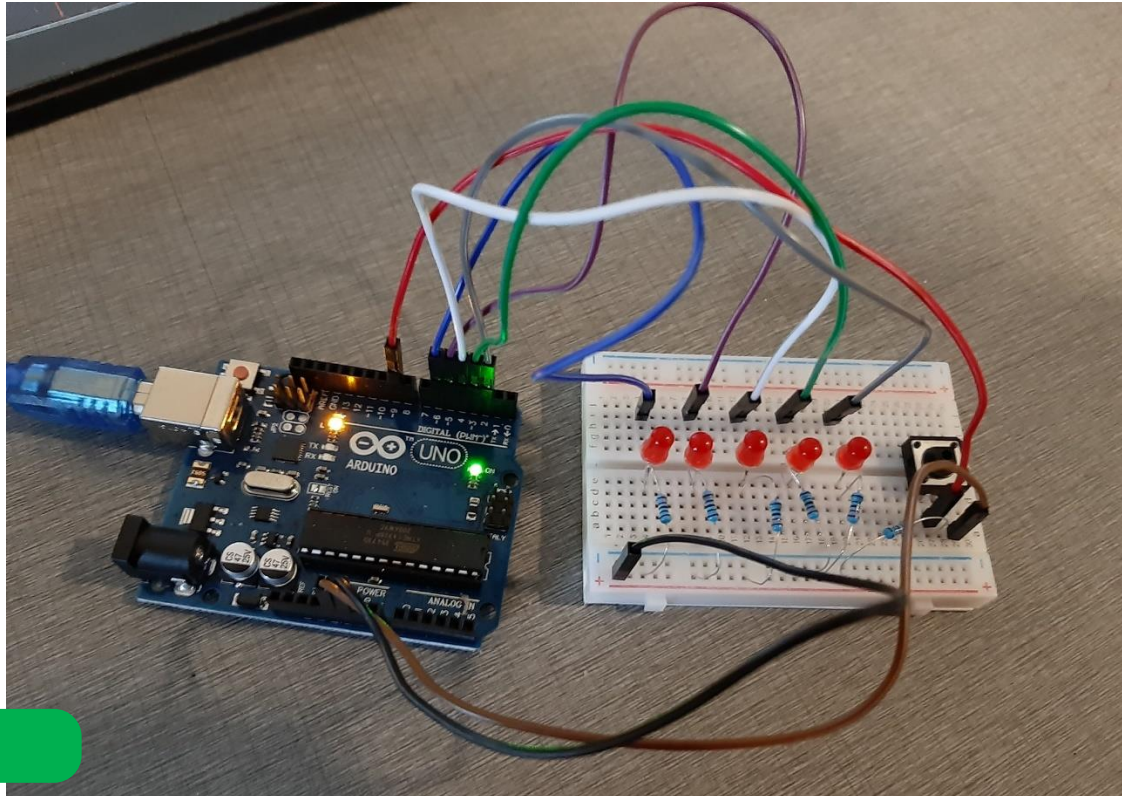
ภาพที่ 3.19 วิธีการต่อวงจร

3.7 ตัวอย่างโปรแกรม การเปิดไฟวิ่งจาก ขวา-ซ้าย, ซ้าย-ขวา



3.7 ตัวอย่างโปรแกรม การเปิดไฟวิ่งจาก ขวา-ซ้าย, ซ้าย-ขวา

3.7.3 ภาพการต่อวงจร



ภาพที่ 3.20 ภาพการต่อวงจร

3.7 ตัวอย่างโปรแกรม การเปิดไฟวิ่งจาก ขวา-ซ้าย, ซ้าย-ขวา

1

```
int btnLight = 9;
int btnStatus = 0;

void setup()
{
  pinMode(btnLight, INPUT);
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  digitalWrite(2,LOW);
  digitalWrite(3,LOW);
  digitalWrite(4,LOW);
  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
}
```

2

```
void loop() {
  btnStatus = digitalRead(btnLight);
  if (btnStatus == HIGH) {
    int x = 2;

    while (x<=6){
      digitalWrite(x,HIGH);
      delay(150);
      x++;
    }
    while (x>=2){
      digitalWrite(x,LOW);
      delay(150);
      x--;
    }
  }
}
```

3.7.4 Code โปรแกรม

3.7 ตัวอย่างโปรแกรม การเปิดไฟวิ่งจาก ขวา-ซ้าย, ซ้าย-ขวา

```
while (x<=6){  
    digitalWrite(x,HIGH);  
    delay(150);  
    x++;  
}  
x = 2;  
while (x<=6){  
    digitalWrite(x,LOW);  
    delay(150);  
    x++;  
}
```

3

```
while (x>=2){  
    digitalWrite(x,HIGH);  
    delay(150);  
    x--;  
}  
x = 6;  
while (x>=2){  
    digitalWrite(x,LOW);  
    delay(150);  
    x--;  
}  
else {  
    digitalWrite(2,LOW);  
    digitalWrite(3,LOW);  
    digitalWrite(4,LOW);  
    digitalWrite(5,LOW);  
    digitalWrite(6,LOW);  
}  
}
```

4

3.7.4 Code โปรแกรม (ต่อ)

3.7 ตัวอย่างโปรแกรม การเปิดไฟวิ่งจาก ขวา-ซ้าย, ซ้าย-ขวา

3.7.5 หน้าจอการเขียนโปรแกรม ด้วย Arduino IDE



```
test_led_01 | Arduino 1.8.12
File Edit Sketch Tools Help
test_led_01
int btnLight = 9;
int btnStatus = 0;

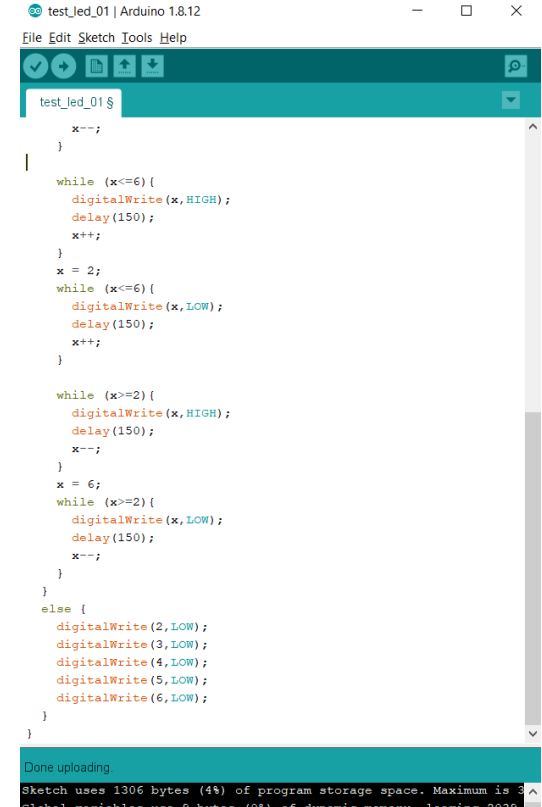
void setup()
{
  pinMode(btnLight, INPUT);
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  digitalWrite(2,LOW);
  digitalWrite(3,LOW);
  digitalWrite(4,LOW);
  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
}

void loop() {
  btnStatus = digitalRead(btnLight);
  if (btnStatus == HIGH) {
    int x = 2;

    while (x<=6){
      digitalWrite(x,HIGH);
      delay(150);
      x++;
    }
    while (x>=2){
      digitalWrite(x,LOW);
      delay(150);
      x--;
    }
  }
}
```

Done uploading.

Sketch uses 1306 bytes (4%) of program storage space. Maximum is 32768 bytes. Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes free.



```
test_led_01 | Arduino 1.8.12
File Edit Sketch Tools Help
test_led_01 $
x--;
}

while (x<=6){
  digitalWrite(x,HIGH);
  delay(150);
  x++;
}
x = 2;
while (x<=6){
  digitalWrite(x,LOW);
  delay(150);
  x++;
}

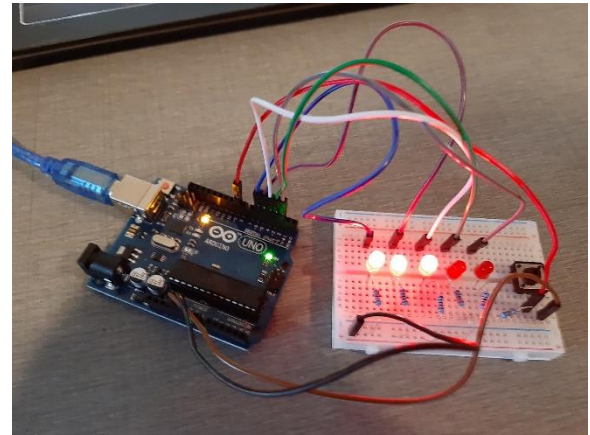
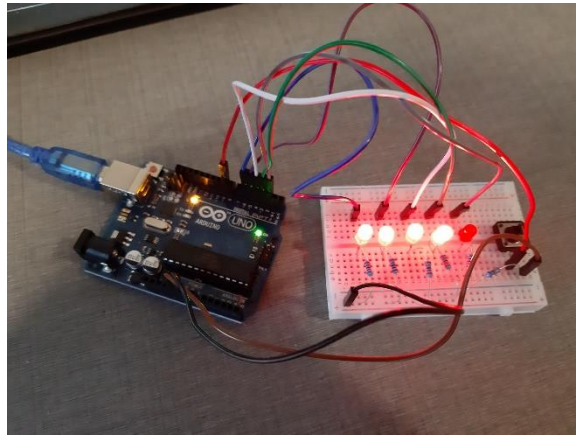
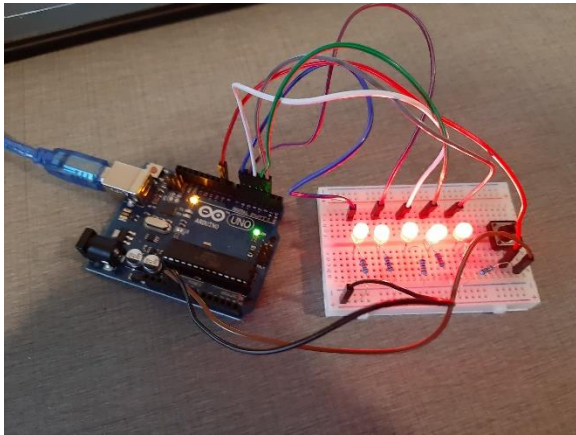
while (x>=2){
  digitalWrite(x,HIGH);
  delay(150);
  x--;
}
x = 6;
while (x>=2){
  digitalWrite(x,LOW);
  delay(150);
  x--;
}
else {
  digitalWrite(2,LOW);
  digitalWrite(3,LOW);
  digitalWrite(4,LOW);
  digitalWrite(5,LOW);
  digitalWrite(6,LOW);
}
}
```

Done uploading.

Sketch uses 1306 bytes (4%) of program storage space. Maximum is 32768 bytes. Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes free.

3.7 ตัวอย่างโปรแกรม การเปิดไฟวิ่งจาก ขวา-ซ้าย, ซ้าย-ขวา

3.7.6 ผลการทำงาน



3.8 ตัวอย่างโปรแกรม การไฟจราจร Traffic Light

3.8.1 Code

```
void setup() {  
  // put your setup code here, to run once:  
  
  pinMode(8, OUTPUT);  
  pinMode(9, OUTPUT);  
  pinMode(10, OUTPUT);  
  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
  
  digitalWrite(10, LOW);  
  digitalWrite(9, LOW);  
  digitalWrite(8, HIGH);  
  delay(4000);
```

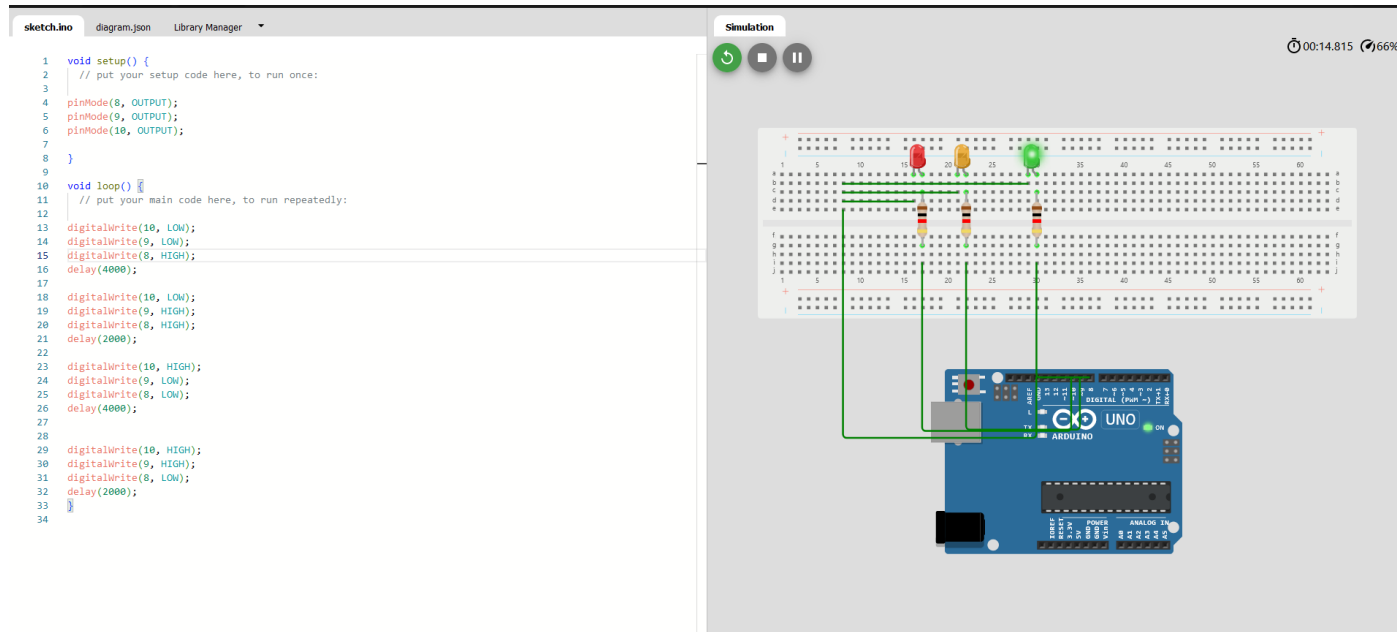
```
    digitalWrite(10, LOW);  
    digitalWrite(9, HIGH);  
    digitalWrite(8, HIGH);  
    delay(2000);
```

```
    digitalWrite(10, HIGH);  
    digitalWrite(9, LOW);  
    digitalWrite(8, LOW);  
    delay(4000);
```

```
    digitalWrite(10, HIGH);  
    digitalWrite(9, HIGH);  
    digitalWrite(8, LOW);  
    delay(2000);  
  }
```

3.8 ตัวอย่างโปรแกรม การไฟจราจร Traffic Light

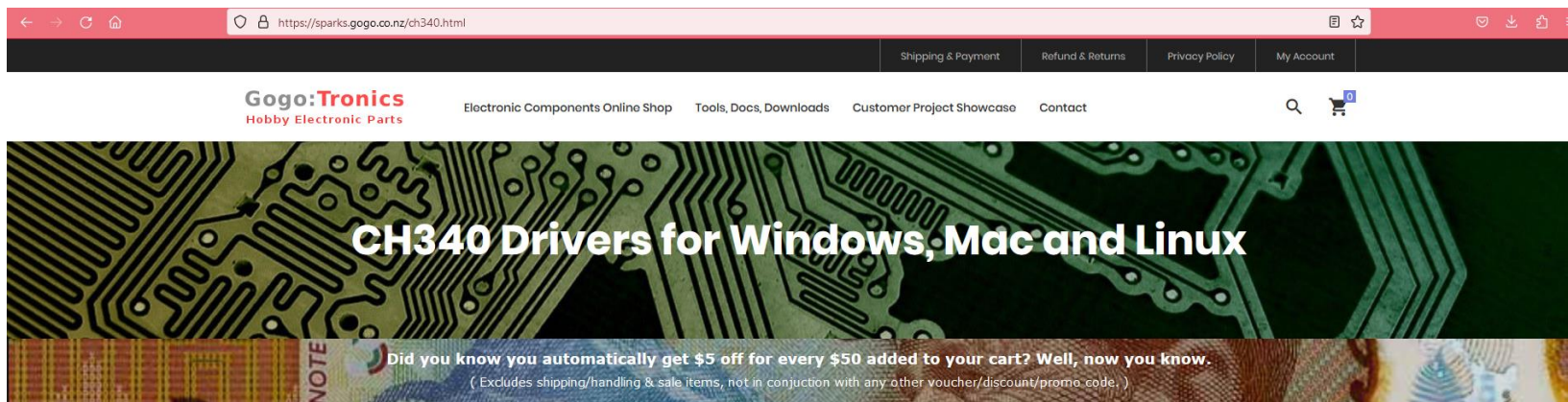
3.8.2 ผลการทำงาน



อ้างอิง

- <http://www.siam2dev.com>
- <http://www.myarduino.net>
- <https://www.allnewstep.com/article/135/สอน-arduino-เรียน-arduino-สำหรับผู้เริ่มต้น-คอร์ส-arduino-ที่ฟรีและดีที่สุด>
- <https://www.allnewstep.com/article/94/6-สอน-arduino-tutorial-arduino-ควบคุมขา-digital-input>

การติดตั้ง Driver



The CH340 chip is used by a number of Arduino compatible boards to provide USB connectivity, you may need to install a driver, don't panic, it's easier than falling off a log, and much less painful.

Windows

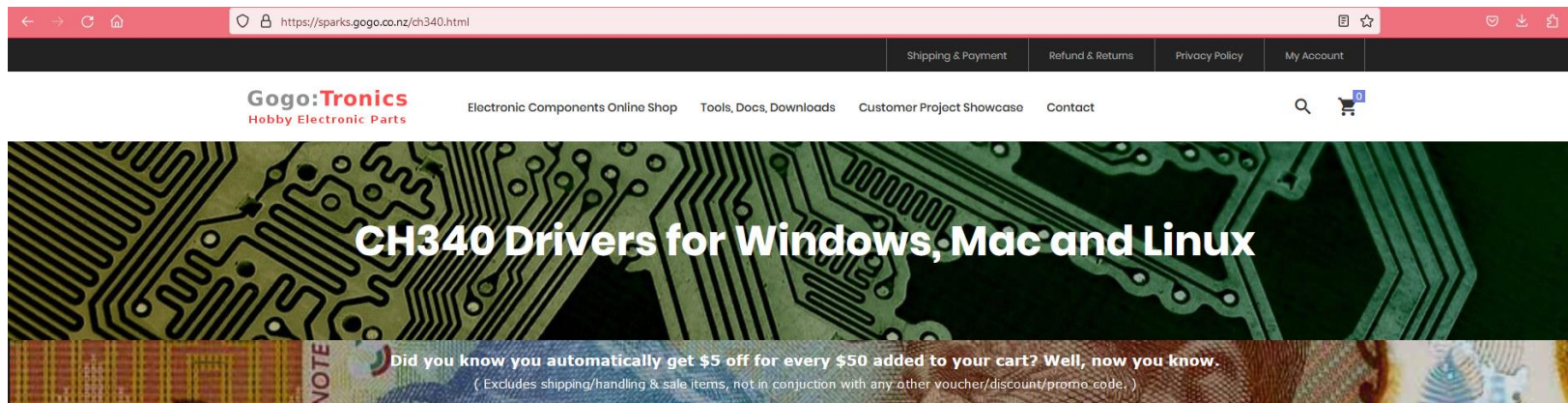
([Manufacturer's Chinese Info Link](#))

- Download the [Windows CH340 Driver](#)
- Unzip the file
- Run the installer which you unzipped
- In the Arduino IDE when the CH340 is connected you will see a COM Port in the Tools > Serial Port menu, the COM number for your device may vary depending on your system.

Older Windows Driver Version and Instructions

[Download Older Windows Driver](#)

Driver :: CH340



The CH340 chip is used by a number of Arduino compatible boards to provide USB connectivity, you may need to install a driver, don't panic, it's easier than falling off a log, and much less painful.

Windows

[\(Manufacturer's Chinese Info Link\)](#)

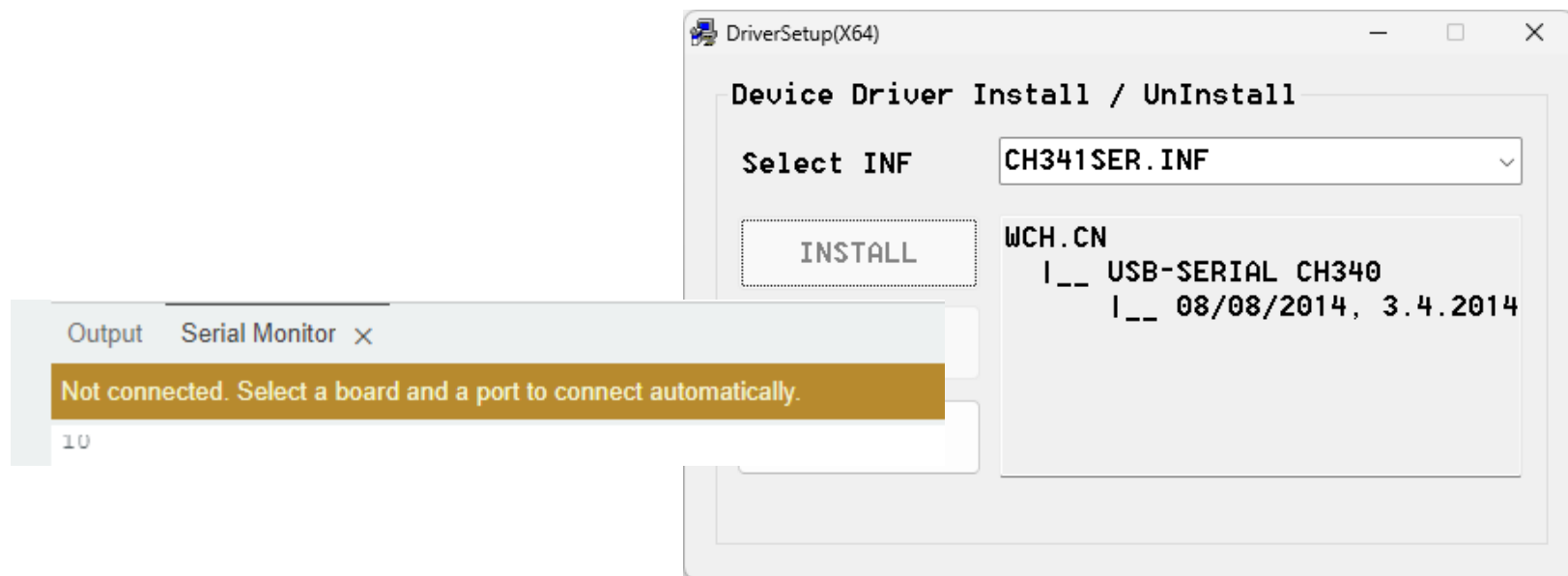
- Download the [Windows CH340 Driver](#)
- Unzip the file
- Run the installer which you unzipped
- In the Arduino IDE when the CH340 is connected you will see a COM Port in the Tools > Serial Port menu, the COM number for your device may vary depending on your system.

Older Windows Driver Version and Instructions

[Download Older Windows Driver](#)

Port Monitor error: command 'open failed: Invalid serial port. Could not connect to COM3 serial port. Q:

อธิบาย



CH34x_Install_Windows_v3_4. X

+

+

New

✂

📄

📄

📄

📄

🗑

↕

Sort

☰

View

📂

Extract all

⋮

← → ▾ ⬆

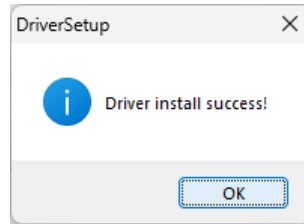
📂 > This PC > Local Disk (C:) > Users > User > Downloads > CH34x_Install_Windows_v3_4.zip

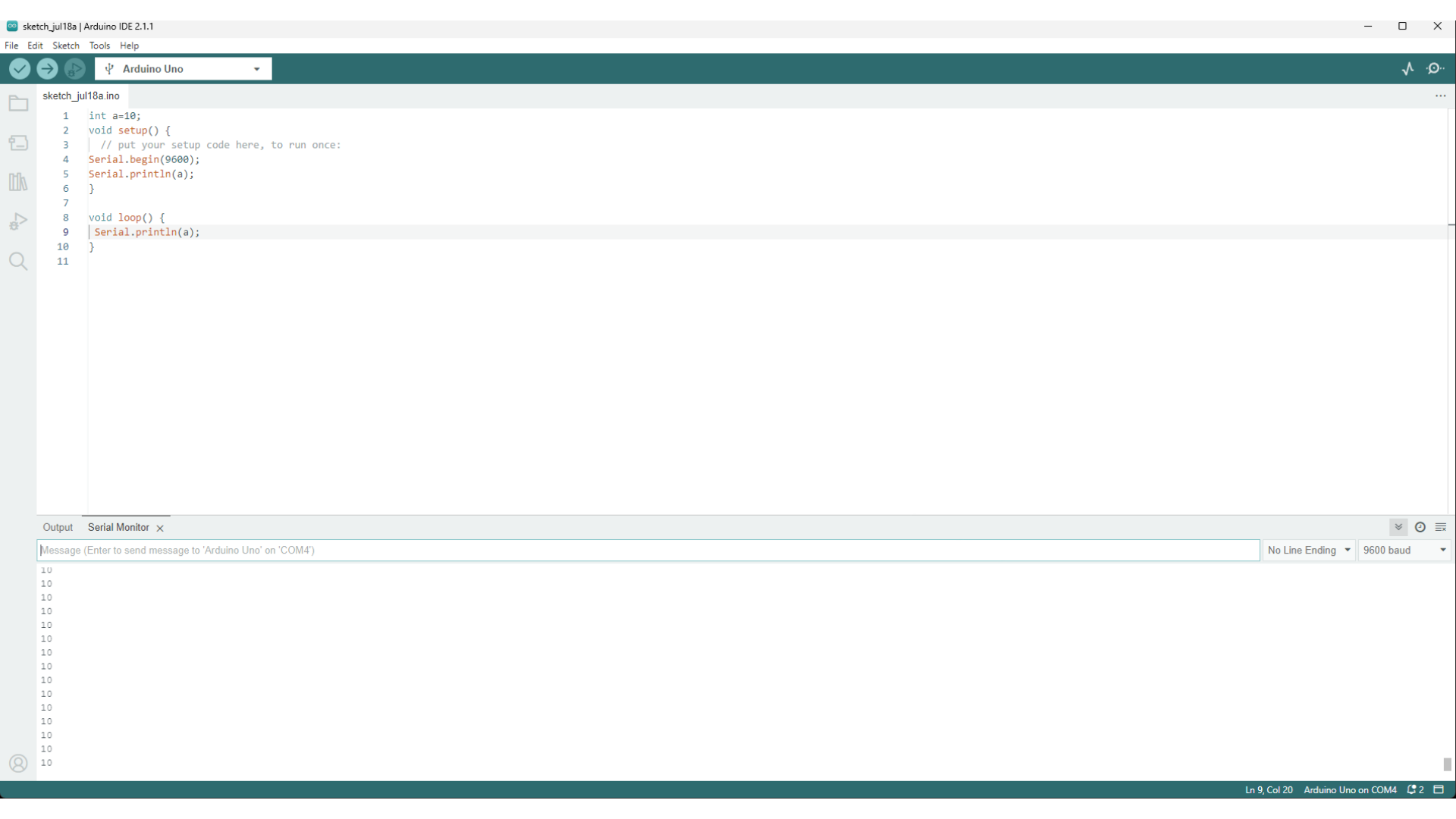
🏠 Home

> ☁ OneDrive - Persona

Name	Type	Compressed size	Password ...	Size	Ratio	Date modified
📄 CH34x_Install_Windows_v3_4.EXE	Application	189 KB	No	238 KB	21%	24/1/2560 13:17

ติดตั้งสำเร็จ





Thank you