**Lab# 4**

การสื่อสารผ่านพอร์ตอนุกรม

**ฟังก์ชั่นเกี่ยวกับการสื่อสารผ่านพอร์ตอนุกรม**

ใช้สำหรับสื่อสารข้อมูลระหว่างฮาร์ดแวร์ Arduino กับคอมพิวเตอร์ หรืออุปกรณ์อื่นๆ โดยจะแบ่งพอร์ตสำหรับเชื่อมต่อออกเป็น 2 ส่วน คือ ส่วนแรกติดต่อพอร์ตอนุกรมเสมือน (virtual COM port) จากการทำงานของส่วนเชื่อมต่อพอร์ต USB ฟังก์ชั่นที่ใช้ คือ Serial อีกส่วนหนึ่ง คือ ขาพอร์ตสื่อสารข้อมูลอนุกรมโดยใช้ขา 0 (RxD) และ 1 (TxD) ฟังก์ชั่นของ Arduino ที่ใช้ คือ Serial1 ดังนั้นเมื่อเลือกใช้งานเป็นขาพอร์ตสื่อสารข้อมูลอนุ กรมแล้ว จะไม่สามารถใช้ขาพอร์ต 0 และ 1 เป็นพอร์ตดิจิตอลได้

**การเรียกใช้งานฟังก์ชั่น**

**Serial.begin(int datarate)**

เป็นการกำหนดค่าอัตราบอดของการรับส่งข้อมูลอนุกรมในหน่วยบิตต่อวินาที (bits per second : bps) สามารถใช้ค่าต่อไปนี้ 300, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600 หรือ 115200

พารามิเตอร์

Int datarate อัตราการรับส่งในหน่วยบิตต่อวินาที (baud หรือ bps)

หมายเหตุ

อัตราบอด (Baud Rate) คือจำนวนสัญญาณดิจิตอลหรืออนาล็อก ที่ส่งผ่านไปในช่องทางสื่อสารภายใน 1 วินาที มีหน่วยเป็นบอดต่อวินาที (baud per second)

**int Serial.available()**

ใช้ในการแจ้งว่าได้รับข้อมูลตัวอักษร (characters) แล้ว และพร้อมสำหรับการอ่านไปใช้งาน

ค่าที่ส่งกลับ

จำนวนไบต์ที่พร้อมสำหรับการอ่านค่า โดยเก็บข้อมูลในบัฟเฟอร์ตัวรับ ถ้าไม่มีข้อมูลจะมีค่าเป็ น 0 ถ้ามีข้อมูลฟังก์ชั่นจะคืนค่าที่มากกว่า 0 โดยบัฟเฟอร์สามารถเก็บข้อมูลได้สูงสุด 128 ไบต์

**int Serial.read()**

ใช้อ่านค่าข้อมูลที่ได้รับจากพอร์ตอนุกรม

ค่าที่ส่งกลับ

เป็นเลข int ที่เป็นไบต์แรกของข้อมูลที่ได้รับ (หรือเป็น -1 ถ้าไม่มีข้อมูล) ใช้อ่านค่าข้อมูลที่ได้รับจากพอร์ตอนุกรม

**Serial.flush()**

ใช้เคลียค่าบัฟเฟอร์ตัวรับข้อมูลของพอร์ตอนุกรม

**Serial.print(data)**

ใช้ส่งข้อมูลออกทางพอร์ตอนุกรม

พารามิเตอร์

Data - เป็นข้อมูลเลขจำนวนเต็ม ได้แก่ char, int หรือเลขทศนิยมที่ตัดเศษออกเป็นเลขจำนวนเต็ม

รูปแบบฟังก์ชั่น

คำสั่งนี้สามารถเขียนได้หลายรูปแบบ

*Serial.print(b)*

เป็นการเขียนคำสั่งแบบไม่ได้ระบุรูปแบบจะพิมพ์ค่าตัวแปร b เป็นเลขฐานสิบ โดยพิมพ์ตัวอักษรตามรหัส ASCII

*Serial.print(b, DEC)*

เป็นคำสั่งพิมพ์ค่าตัวแปร b เป็นตัวเลขฐานสิบ โดยพิมพ์ตัวอักษรตามรหัส ASCII

*Serial.print(b, HEX)*

เป็นคำสั่งพิมพ์ค่าตัวแปร b เป็นตัวเลขฐานสิบหก โดยพิมพ์ตัวอักษรตามรหัส ASCII

*Serial.print(b, OCT)*

เป็นคำสั่งพิมพ์ค่าตัวแปร b เป็นตัวเลขฐานแปด โดยพิมพ์ ตัวอักษรตามรหั ส ASCII

Serial.print(b, BIN)

เป็นคำสั่งพิมพ์ค่าตัวแปร b เป็นตัวเลขฐานสอง โดยพิมพ์ตัวอักษรตามรหั ส ASCII

*Serial.print(b, BYTE)*

เป็นคำสั่งพิมพ์ค่าตัวแปร b ขนาด 1 ไบต์

*Serial.print(str)*

เป็นคำสั่งพิมพ์ค่าข้อความในวงเล็บ หรือข้อความที่เก็บในตัวแปร str

หมายเหตุ

Serial.print() จะตัดเศษเลขทศนิยมเหลือเป็นเลขจำนวนเต็ม ทางแก้ไขทางหนึ่งคือ คูณเลขทศนิยมด้วย 10, 100, 1000 ฯลฯ ขึ้นอยู่ กับจำนวนหลักของเลขทศนิยม เพื่อแปลงเลขทศนิยมเป็นจำนวนเต็มก่อน แล้วจึงส่งข้อมูลออกพอร์ตอนุกรม จากนั้นที่ฝั่งภาครับให้ทำการหารค่าที่รับได้ เพื่อแปลงกลับเป็นเลขทศนิยม

**Serial.println(data)**

เป็นฟังก์ชั่นพิมพ์ (หรือส่ง) ข้อมูลออกทางพอร์ตอนุกรมตามด้วยรหัส carriage return (รหัส ASCII หมายเลข 13 หรือ \r) และ linefeed (รหัส ASCII หมายเลข 10 หรือ \n) เพื่อให้เกิดการเลื่อนบรรทัด และขึ้นบรรทัดใหม่ หลังจากพิมพ์ข้อความ มีรูปแบบเหมือนคำสั่ง Serial.print()

รูปแบบฟังก์ชั่น

*Serial.println(b)*

เป็นคำสั่งพิมพ์ข้อมูลแบบไม่ได้ระบุรูปแบบ จะพิมพ์ค่าตัวแปรเป็นเลขฐานสิบ ตามด้วยรหัสอักษร carriage return และ linefeed

*Serial.println(b, DEC)*

เป็นคำสั่งพิมพ์ค่าตัวแปร b เป็นตัวเลขฐานสิบ ตามด้วยรหัสอักษร carriage return และ linefeed

*Serial.println(b, HEX)*

เป็นคำสั่งพิมพ์ค่าตัวแปร b เป็นตัวเลขฐานสิบหก ตามด้วยรหัสอักษร carriage return และ linefeed

*Serial.println(b, OCT)*

เป็นคำสั่งพิมพ์ค่าตัวแปร b เป็นตัวเลขฐานแปด ตามด้วยรหัสอักษร carriage return และ linefeed

*Serial.println(b, BIN)*

เป็นคำสั่งพิมพ์ค่าตัวแปร b เป็นตัวเลขฐานสอง ตามด้วยรหัสอักษร carriage return และ linefeed

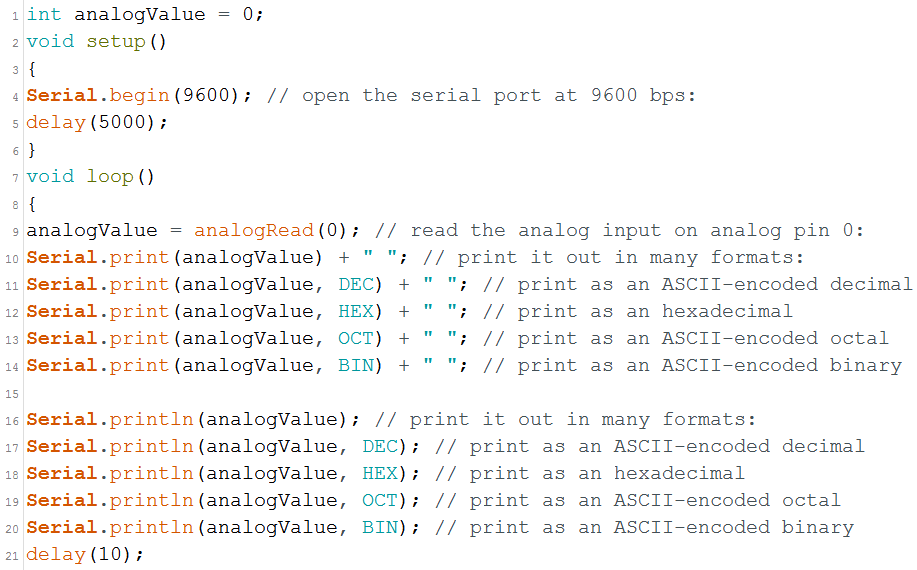
*Serial.println(str)*

พิมพ์ค่าในวงเล็บ หรือ ข้อความที่เก็บในตัวแปร str ตามด้วยรหัสอักษร carriage return และ linefeed

*Serial.println()*

เป็นคำสั่งพิมพ์รหัส carriage return และ linefeed

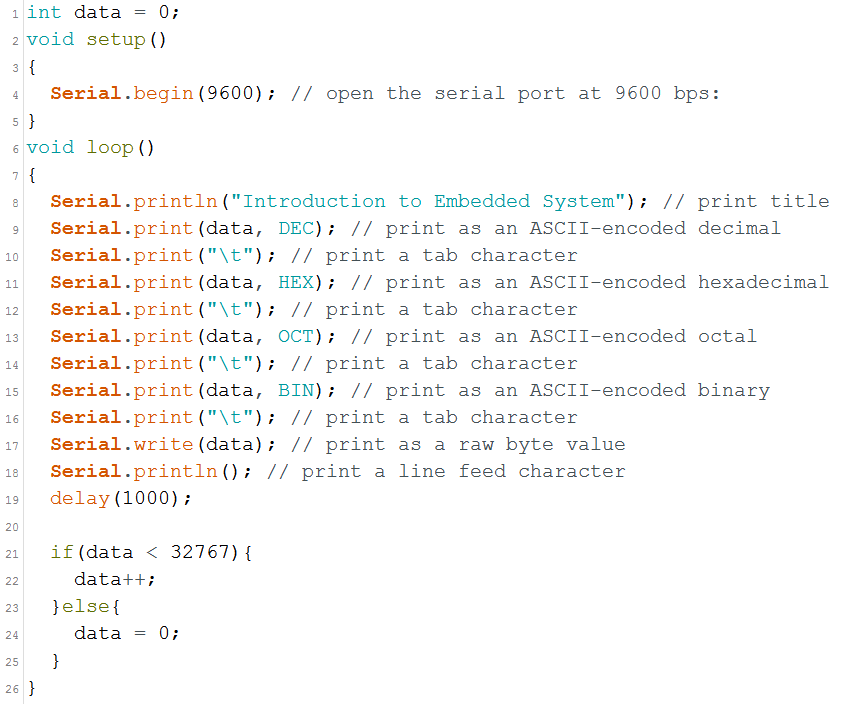
ตัวอย่าง



**การทดลองที่ 1 การส่งข้อมูลออกพอร์ตอนุกรม**

เริ่มต้นด้วยการใช้ฟังก์ชั่น Serial.begin() เพื่อสั่งเปิดพอร์ตอนุกรม และกำหนดอัตราถ่ายทอดข้อมูลที่ ใช้ในการสื่อสารข้อมูลหรืออัตราบอด ปกติค่าที่ใช้ คือ 9600 หลังจากเปิดพอร์ตเรียบร้อยแล้ว เมื่อต้องการส่งข้อมูลให้ใช้ฟังก์ชั่น Serial.print() หรือ Serial.println()

Source Code

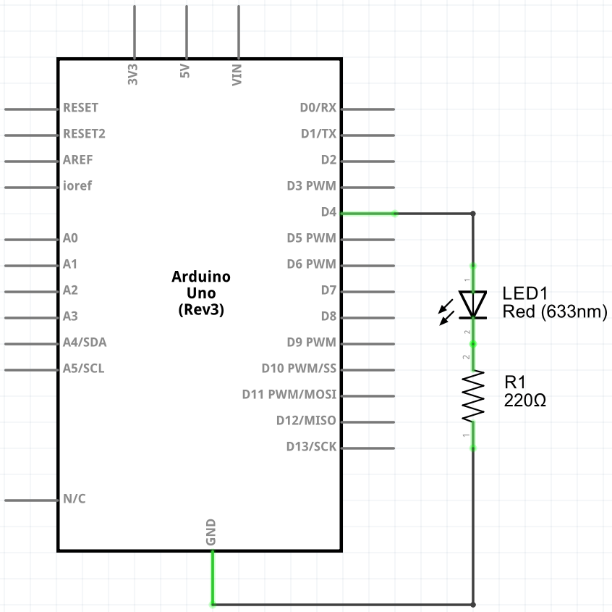


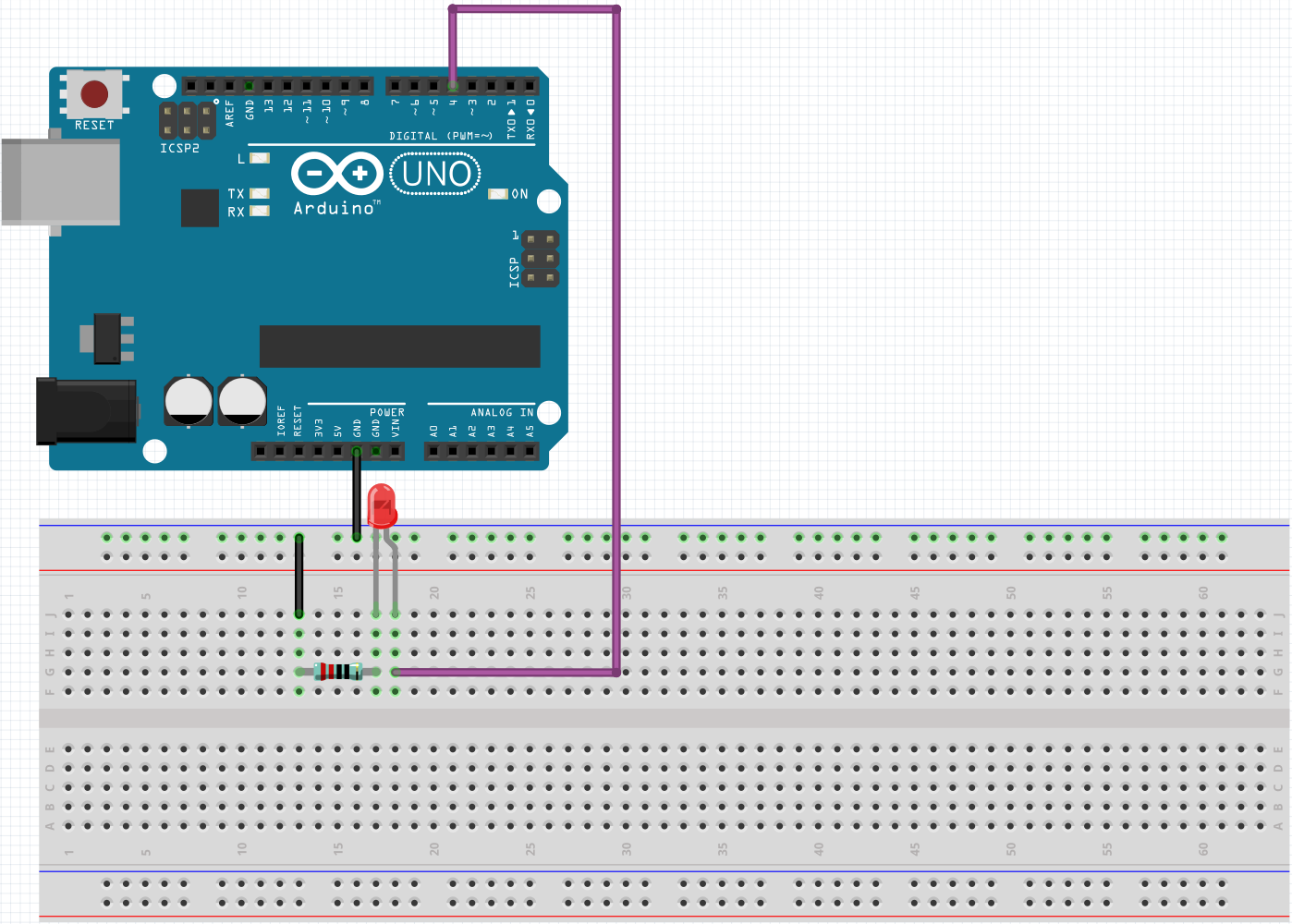
บันทึกผลการทดลอง

พิมพ์ค่าตั้งแต่ 0 – 32766 เป็นเลขฐาน 10 16 8 2 ตามลำดับ

**การทดลองที่ 2 การรับค่าจากพอร์ตอนุกรมเพื่อกำหนดความเร็วในการกระพริบของ LED**

ในการรับค่าจากพอร์ตอนุกรมจะใช้ฟังก์ชั่น 2 ตัวคือ Serial.avaliable() และ Serial.read() โดยเริ่มจากใช้ฟังก์ชั่น Serial.avaliable() เพื่อตรวจสอบว่ามีข้อมูลหรือไม่ ฟังก์ชั่นจะคืนค่าเป็นเลขจำนวนเต็มแสดงจำ นวนข้อมูลในบัฟเฟอร์ตัวรับของพอร์ตอนุกรม ถ้าอ่านค่าได้เท่ากับ 0 แสดงว่าไม่มีข้อมูล เมื่อทดสอบพบว่า ฟังก์ชั่น Serial.avaliable() คืนค่าไม่เท่ากับ 0 ถัดมาให้ใช้ฟังก์ชั่น Serial.read() เพื่ออ่านค่าจากบัฟเฟอร์ตัวรับ ฟังก์ชั่นคืนค่าเป็นเลขจำนวนเต็มที่เป็นไบต์แรกของข้อมูลที่ได้รับ (หรือเป็น -1 ถ้าไม่มีข้อมูล) ในการทดลองนี้เป็นการรับค่าจากพอร์ตอนุกรม เพื่อนำค่าที่รับได้ไปควบคุมอัตราการกระพริบของ LED

รูปแบบการต่อวงจร



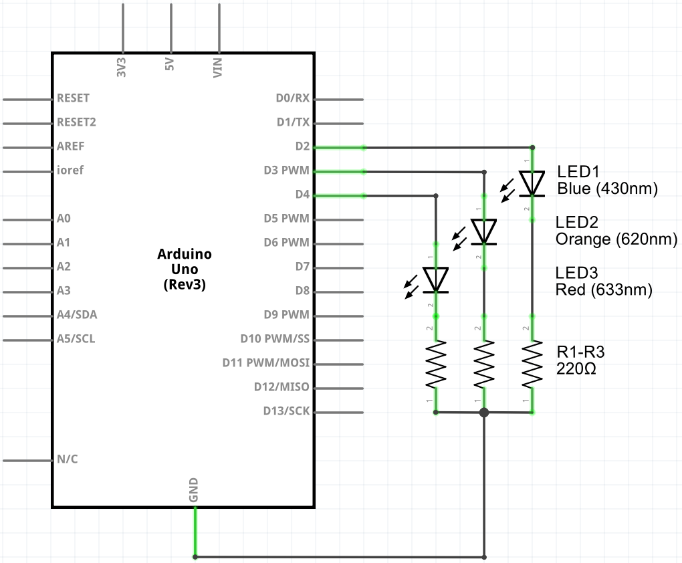
Source Code

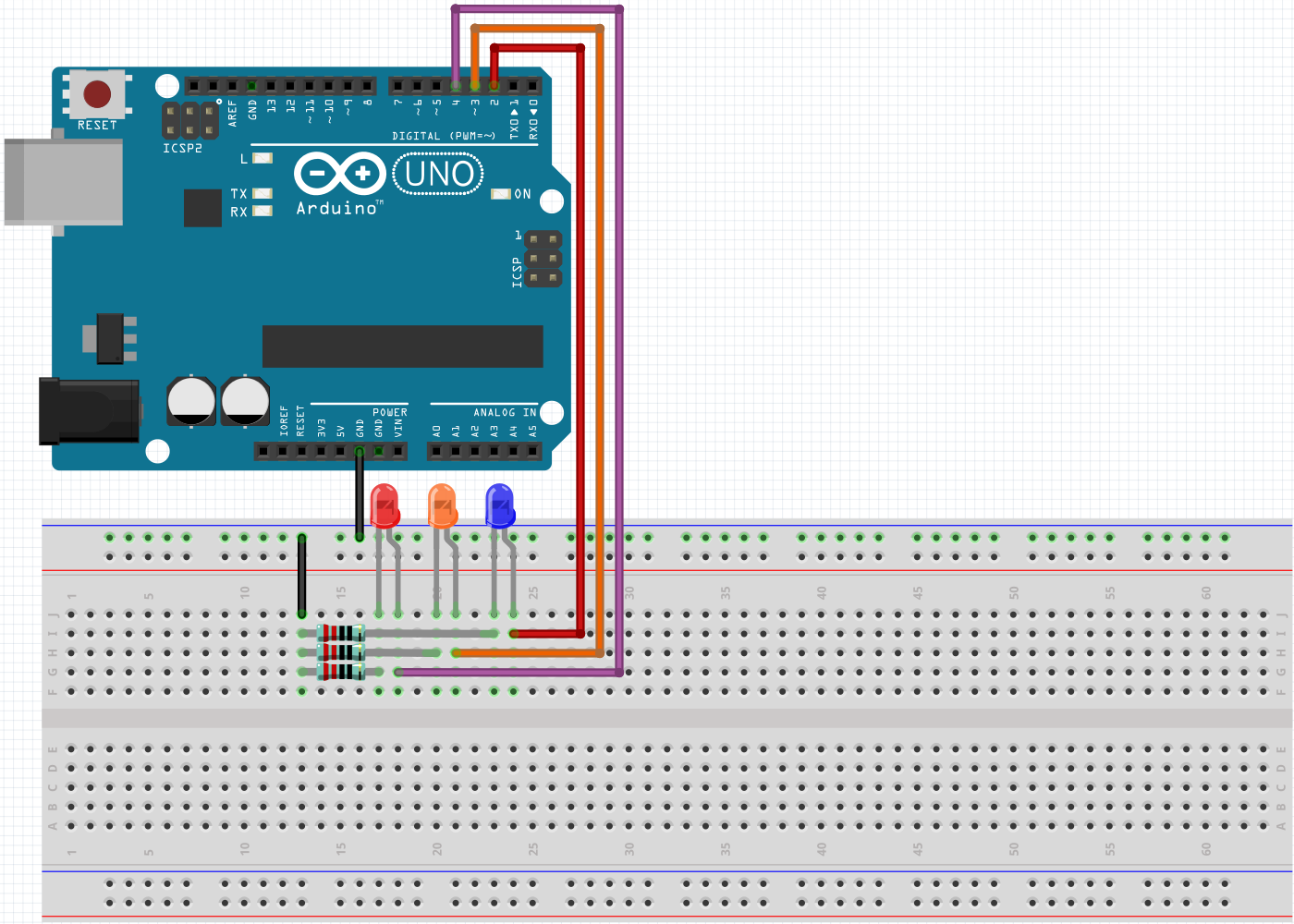


บันทึกผลการทดลอง

ป้อนค่าผ่าน Serial port เพื่อควบคุม delay ของไฟกระพริบ

ทดลองเปลี่ยนการต่อวงจรตามรูปแบบการต่อวงจรด้านล่าง และเขียนโปรแกรมเพื่อควบคุมความเร็วการวิ่งของหลอด LED 3 ดวง โดยรับค่าความเร็วจากพอร์ตอนุกรม

รูปแบบการต่อวงจร



Source Code

#define LED2\_PIN 3

#define LED1\_PIN 2

#define LED3\_PIN 4

int incomingByte = 0 ;

int delayTime = 100;

void setup() {

// put your setup code here, to run once:

pinMode(LED1\_PIN,OUTPUT);

pinMode(LED2\_PIN,OUTPUT);

pinMode(LED3\_PIN,OUTPUT);

Serial.begin(9600);

delay(5000);

}

void loop() {

// put your main code here, to run repeatedly:

if(Serial.available() > 0)

{

incomingByte = Serial.read();

if(incomingByte >= 49 && incomingByte <=53){

Serial.write(incomingByte);

Serial.println();

delayTime = (incomingByte-48)\*100;

}

Serial.print("Press 1-5 for control speed of LED : ");

}

digitalWrite(LED1\_PIN, HIGH);

delay(delayTime);

digitalWrite(LED2\_PIN, HIGH);

delay(delayTime);

digitalWrite(LED3\_PIN, HIGH);

delay(delayTime);

digitalWrite(LED1\_PIN, LOW);

delay(delayTime);

digitalWrite(LED2\_PIN, LOW);

delay(delayTime);

digitalWrite(LED3\_PIN, LOW);

delay(delayTime);

}

บันทึกผลการทดลอง

ป้อนค่าผ่าน Serial port เพื่อควบคุม delay ของไฟกระพริบ 3 ดวง 1 คือเร็วสุด 5 คือสุด