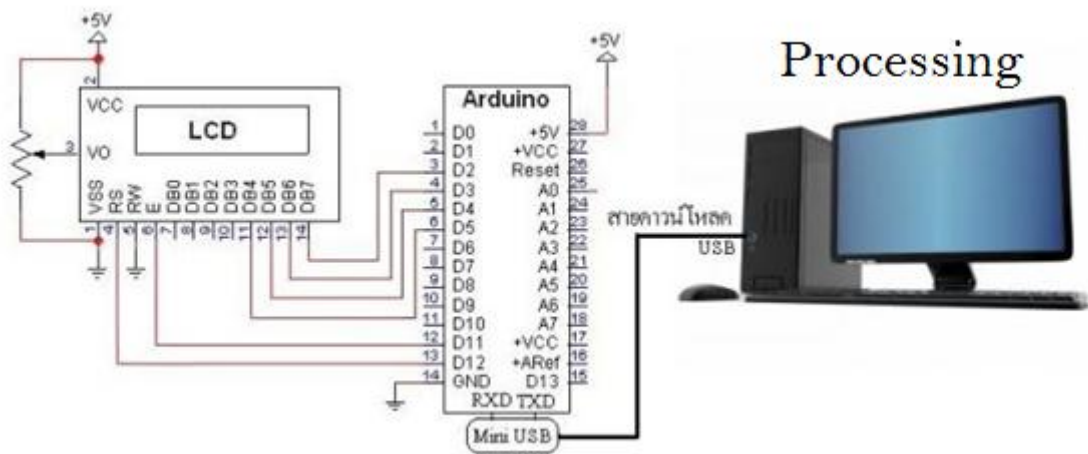


การใช้ Processing กับ Aduino



สารบัญ

1	การสื่อสารระหว่าง Arduino กับ Processing.....	2
2	การต่อวงจร.....	2
3	ทดสอบการทำงานของ Arduino ผ่านทางพอร์ตอนุกรม	3
4	การใช้งานพอร์ตอนุกรมบน โปรแกรม Processing.....	4
5	การทดลองควบคุมบอร์ด Arduino ด้วย Processing ผ่านทางพอร์ตอนุกรม	5
6	การทดลองควบคุม Arduino ด้วย Processing แบบที่ 2.....	5
7	โปรแกรม ใช้สวิทช์แบบ กดติด ปุ่มกด คำนวณการติดดับของ LED บน Arduino	6
8	โปรแกรมที่ 3 รับคำสั่งจากบอร์ด Arduino มาที่ processing.....	7
9	งานมอบหมาย.....	9

1 การสื่อสารระหว่าง Arduino กับ Processing

สายคาร์บอน โหลดของอาร์ดูโนเป็นพอร์ทอนุกรม สามารถใช้สายนี้สื่อสารกับ Processing ที่เขียนอยู่บน คอมพิวเตอร์ที่มีพอร์ทอนุกรมได้ การทดลองนี้จะใช้พอร์ทอนุกรมของอาร์ดูโน สื่อสารกับเครื่องพีซี โดยที่พีซีจะใช้โปรแกรม Processing ส่วน Arduino จะต่อกับ อุปกรณ์ IO เช่น push button switch และ หลอด LED ใช้ Processing เป็นเหมือนกับ GUI (Graphics User Interface ของระบบควบคุม)

หมายเหตุ เนื่องจาก พอร์ทอนุกรมของ Arduino ใช้ขาสัญญาณเพื่อการติดต่ออยู่ 2 ขาคือ

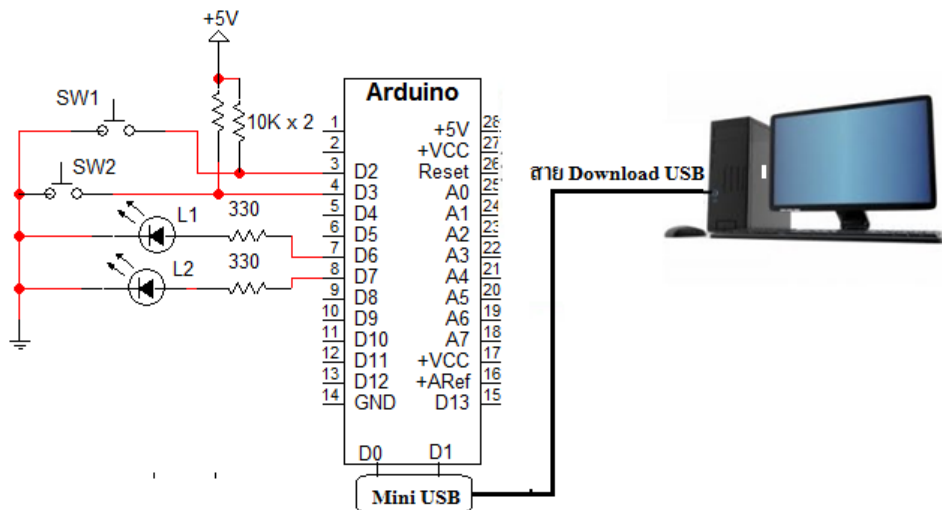
ขาข้อมูล RxD ใช้ขา D0

ขาส่งข้อมูล TxD ใช้ขา D1

ดังนั้นเมื่อใช้พอร์ทอนุกรม ขา D0 และ D1 จึงใช้เป็นสัญญาณ Digital I/O ไม่ได้ และขณะ Download ลงบอร์ด จะใช้ขาทั้งสองนี้ ดังนั้นขณะ Download โปรแกรมขาทั้งสองต้องไม่มีอุปกรณ์อื่นๆต่ออยู่

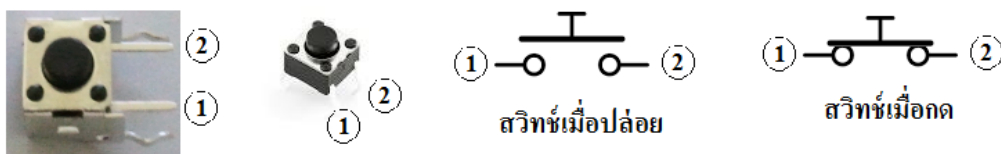
2 การต่อวงจร

ประกอบวงจรตามรูปที่ 1



รูปที่ 1

push button switch หรือที่เรียกว่า สวิตช์แบบกดติดปลายนิ้ว มีการทำงานแบบเมื่อกดสวิตช์ ขั้วทั้งสองของสวิตช์จะต่อถึงกันกระแสไฟฟ้าสามารถไหลผ่านได้ เมื่อปล่อยปุ่มกด ขั้วทั้งสองจะขาดจากกันกระแสไฟฟ้าผ่านไม่ได้ รูปร่างและสัญลักษณ์ของสวิตช์แสดงอยู่ในรูปที่ 2



รูปที่ 2

3 ทดสอบการสั่งงาน Arduino ผ่านทางพอร์ตอนุกรม

การอ่านข้อมูลจากพอร์ตอนุกรมใช้ serialEvent และ Serial.read();

การส่งข้อมูลออกทางพอร์ตอนุกรมใช้คำสั่ง Serial.println

- serialEvent เมื่อมีข้อมูลเข้ามาที่พอร์ตอนุกรม ซีพียูจะมาทำฟังก์ชันนี้
- Serial.read() อ่านข้อมูลจากพอร์ตอนุกรม 1 ไบต์ ได้ค่าเป็น int
- Serial.write ส่งข้อมูลเป็นเลขไบนารีออกทางพอร์ตอนุกรม ข้อมูลอาจมีเพียงไบต์เดียว หรือเป็นหลายๆไบต์ก็ได้ โดยมีรูปแบบดังนี้
Serial.write(val) หรือ Serial.write(str) หรือ Serial.write(buf, len)
- Serial.println ส่งข้อมูลที่เป็นข้อความออกทางพอร์ตอนุกรม พร้อมกับรหัส CR LF

- 1) ต่อบอร์ดตามรูปที่ 1
- 2) เขียนโปรแกรมบน Arduino ดังนี้

โปรแกรมที่ 1 โปรแกรมควบคุมการทำงานของ Arduino ผ่านพอร์ตอนุกรม

```
#define L1_OFF digitalWrite(6,LOW)
#define L1_ON  digitalWrite(6,HIGH)
#define L2_OFF digitalWrite(7,LOW)
#define L2_ON  digitalWrite(7,HIGH)
void setup() {
  Serial.begin(9600); // initialize serial:
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  digitalWrite(6,LOW);
  digitalWrite(7,LOW);
}

void loop() {

}

void serialEvent() {
  char inChar = (char)Serial.read();
  switch (inChar) {
    case '0':
      L1_OFF;
      break;
    case '1':
      L1_ON;
      break;
    case '2':
      L2_OFF;
      break;
    case '3':
      L2_ON;
      break;
    default:
      break;
  }
}
```

สังเกตการณ์เรียกใช้คำสั่ง `char inChar = (char)Serial.read();` เป็นการอ่านข้อมูลจากพอร์ทอนุกรมเข้ามา 1 ไบต์ แล้วเปลี่ยน data type ให้เป็นชนิดอักขระ

- 3) แพลและ Upload โปรแกรมลง Arduino
- 4) เปิด Serial Monitor ทดลองพิมพ์เลข 3 แล้วกด enter สังเกตดูว่า LED ติดดับอย่างไร ทดลองเปลี่ยนเป็นพิมพ์เลขอื่นๆ ตั้งแต่ 0 ถึง 3

4 การใช้งานพอร์ทอนุกรมบนโปรแกรม Processing

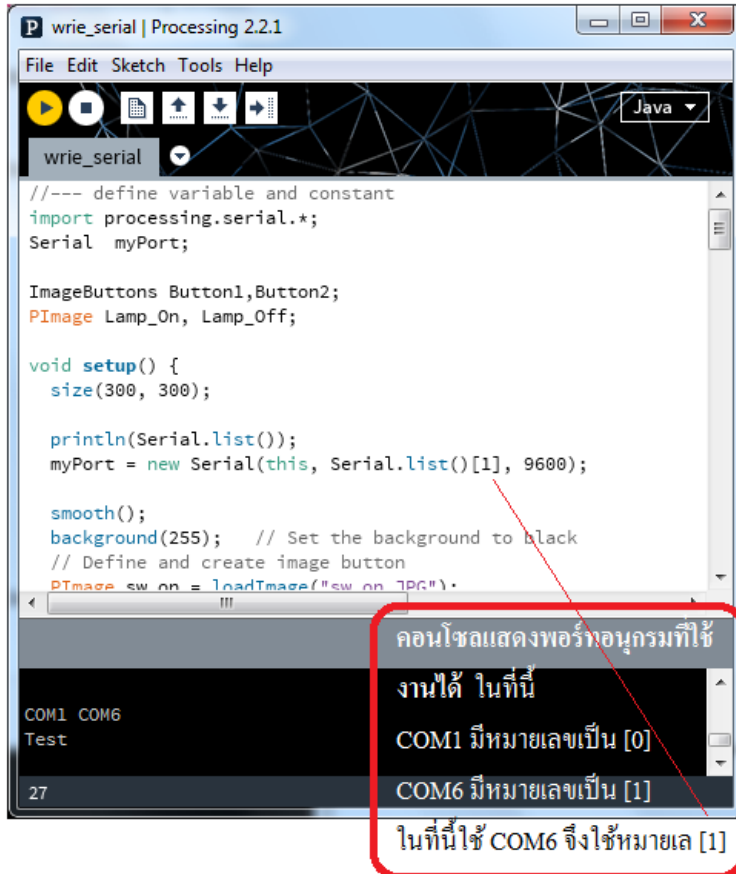
คำสั่งเกี่ยวกับ Serial ของ Processing

- `import processing.serial.*;` เรียกใช้ ไลบรารี serial (พอร์ทอนุกรม)
- `Serial.list()` ให้แสดงรายชื่อพอร์ทอนุกรมที่เห็น

เริ่มต้นเราต้องใช้คำสั่งตรวจสอบว่า Processing มองเห็นพอร์ทอนุกรมเป็นหมายเลขอะไรโดยใช้คำสั่ง

`Serial.list()`

```
import processing.serial.*; //เรียกใช้ ไลบรารี serial (พอร์ทอนุกรม)
void setup() {
  println(Serial.list()); //ให้แสดงรายชื่อพอร์ทอนุกรมที่เห็น
}
void draw() {
}
```



- serial.write(src) ส่งข้อมูลออกทางพอร์ทอนุกรม ข้อมูลอาจเป็น ไบต์ เลขจำนวนเต็ม string
- serial.read() อ่านข้อมูลจากพอร์ทอนุกรม ข้อมูลเป็นได้หลายแบบ

5 การทดลองควบคุมบอร์ด Arduino ด้วย Processing ผ่านทางพอร์ทอนุกรม

จากการทดลองควบคุมบอร์ด Arduino ผ่านทางพอร์ทอนุกรม โดยทำให้ LED ติดแล้วดับ เรียงกันไปตลอด

- 1) ตัวอย่างตามรูปที่ 1
- 2) โปรแกรม Arduino ยังเป็นตามโปรแกรมที่ 1
- 3) เขียนโปรแกรมบน Processing ตามโปรแกรมที่ 2

โปรแกรมที่ 2

```
import processing.serial.*;
Serial myPort;

void setup() {
  size(350, 300);

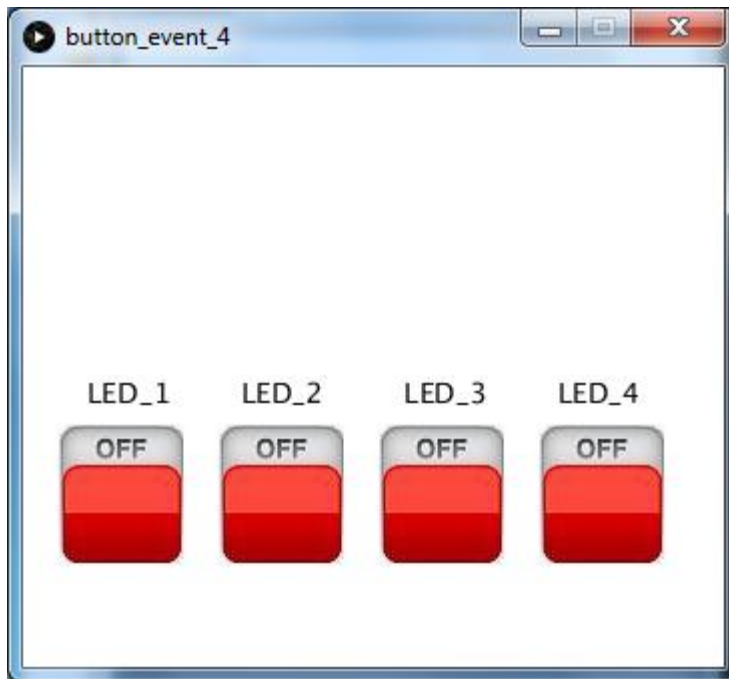
  println(Serial.list());
  myPort = new Serial(this, Serial.list()[1], 9600); // [1]
  background(255); // Set the background to black
  println("Test");
}
void draw()
{
  myPort.write("1"); //ส่งคำสั่งไป arduino
  println("1"); //Print to console
  delay(500);
  myPort.write("0");
  println("0");
  delay(500);
  myPort.write("3");
  println("3");
  delay(500);
  myPort.write("2");
  println("2");
  delay(500);
}
```

- 4) Run Processing แล้วสังเกตการณ์ติดดับของ LED
- 5) ทำความเข้าใจโปรแกรม

6 การทดลองควบคุม Arduino ด้วย Processing แบบที่ 2

ตัวอย่างเป็นการคลิกภาพสวิทซ์ที่จอ Processing แล้วให้ LED ที่บอร์ด Arduino ติดและดับตามการคลิก

- 1) ให้ Download โปรแกรม button_event_4.zip นำมาขยายและ Copy ไว้ที่โฟลเดอร์ sketch ของ Processing
- 2) Run โปรแกรม button_event_4 จะได้ภาพหน้าจอดังนี้



- 3) ให้ใช้เมาส์คลิกที่สวิทช์ LED_1 และ LED_2 ตรวจสอบการทำงานที่ LED ทั้ง 2 ต้องติดและดับตามการคลิกสวิทช์

โปรแกรมที่ 2 กดสวิทช์ที่บอร์ด arduino แล้วให้แสดงข้อความที่หน้าจอ processing ว่าสวิทช์ใดกด

7 โปรแกรมใช้สวิทช์แบบ กดติด ปล่อยดับ ควบคุมการติดดับของ LED บน Arduino

ใช้สวิทช์แบบกดติดปล่อยดับควบคุมการติดดับของ LED บนบอร์ด Arduino

โปรแกรมที่ 3 โปรแกรมบน Arduino

```
#define Read_sw1 digitalRead(2)
#define Read_sw2 digitalRead(3)
#define L1_OFF digitalWrite(6,LOW)
#define L1_ON digitalWrite(6,HIGH)
#define L2_OFF digitalWrite(7,LOW)
#define L2_ON digitalWrite(7,HIGH)
boolean L1 = false; //สถานะของ LED1 ดับ
boolean L2 = false; //สถานะของ LED2 ดับ
void setup() {
  Serial.begin(9600); // initialize serial:
  pinMode(2, INPUT);
  pinMode(3, INPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
}
```

```

digitalWrite(6,LOW);
digitalWrite(7,LOW);
}

void loop() {
  if(Read_sw1== 0)
  {
    if(L1)                //ถ้า L1 ดัด ให้ L1 ดับ
    {
      L1_OFF;
      L1 = false;
    }
    else                  //ถ้า L1 ดับ ให้ L1 ดัด
    {
      L1_ON;
      L1 = true;         //สถานะของ LED1 ดัด
    }
    delay(200);          //หน่วงเวลา 200 ms
  }

  if(Read_sw2== 0)
  {
    if(L2)
    {
      L2_OFF;
      L2 = false;
    }
    else
    {
      L2_ON;
      L2 = true;        //สถานะของ LED2 ดัด
    }
    delay(200);          //หน่วงเวลา 200 ms
  }
}

```

8 โปรแกรมที่ 3 รับคำสั่งจากบอร์ด Arduino มาที่ processing

จากข้อ 7 โปรแกรมใช้สวิทช์แบบกดติดปล่อยดับควบคุมการติดดับของ LED บนบอร์ด Arduino ให้เพิ่มการส่งสถานะการณ้ติดดับของ LED ไปแสดงบนคอนโซลของ Processing

โปรแกรมที่ 4 นี้ต้องเอาโปรแกรม Arduino โปรแกรมที่ 3 ในข้อ 7 มาดัดแปลงเล็กน้อยให้ส่งข้อมูลออกทางพอร์ตอนุกรมด้วย ในที่นี้จะแสดงเฉพาะโปรแกรมส่วนที่ดัดแปลงเท่านั้น ส่วนอื่นๆของโปรแกรมายังเหมือนเดิมทุกอย่าง

```

void loop() {
  if(Read_sw1== 0)
  {
    if(L1)

```

```

{
  L1_OFF;
  L1 = false;
  Serial.println("L1 OFF");          //เพิ่มคำสั่งนี้
}
else
{
  L1_ON;
  L1 = true;
  Serial.println("L1 ON");          //เพิ่มคำสั่งนี้
}
delay(200);
}

if(Read_sw2== 0)
{
  if(L2)
  {
    L2_OFF;
    L2 = false;
    Serial.println("L2 OFF");        //เพิ่มคำสั่งนี้
  }
  else
  {
    L2_ON;
    L2 = true;
    Serial.println("L2 ON");        //เพิ่มคำสั่งนี้
  }
  delay(200);
}
}

```

ส่วนโปรแกรม Processing มีดังนี้

```

//--- define variable and constant
import processing.serial.*;
Serial myPort;

void setup() {
  size(300, 300);

  println(Serial.list());
  myPort = new Serial(this, Serial.list()[1], 9600);
  smooth();
  background(255); // Set the background to black
  stroke(255);    // Set stroke color to white
  println("Test");
}
void draw()
{
}
void serialEvent(Serial myPort) {
  // read String from the serial port:

```



```
String inString = myPort.readString(); //รับข้อมูล String มาเก็บ  
println(inString);  
}
```

9 งานมอบหมาย

- 1) จากข้อ 6 ให้เปลี่ยนเป็นควบคุม LED บน Arduino 4 ดวง
- 2) จากข้อ 8 ให้เปลี่ยนจากการแสดงข้อความบน Console ของ Processing มาเป็นแสดงเป็นรูปหลอดไฟ ติด-ดับ บน ตามสวิตช์บน Arduino