

## 01236255 INTRODUCTION TO INTERNET OF THINGS

### โครงสร้างของการเขียนโปรแกรม

การเขียนโปรแกรมเพื่อสั่งงาน Arduino นั้นจะใช้ภาษาซี โดยโครงสร้างของการเขียนโปรแกรมจะคล้ายๆกับการเขียนโปรแกรมบนไมโครคอนโทรลเลอร์ทั่วๆไป แต่จะมีความง่ายกว่า เพราะ Arduino ได้มีการรวมคำสั่งและ Library ต่างๆไว้ให้ผู้ใช้สามารถเรียกใช้งานได้เลย ถ้าจะกล่าวถึงการเขียนโปรแกรมด้วยภาษาซีนั้น ฟังชันหลักที่จะหาดไม่ได้เลยนั้นคือฟังก์ชัน Main แต่การเขียนโปรแกรมกับ Arduino จะเป็นฟังก์ชัน setup และฟังก์ชัน loop ตามรูป

```
void setup() {  
    // put your setup code here, to run once:  
  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
  
}
```

รูปโครงสร้างการเขียนโปรแกรมขั้นต่ำของ Arduino

การทำงานของฟังก์ชันทั้งสองมีความหมายดังนี้

การทำงานของ void setup() เป็นฟังก์ชันที่ใช้สำหรับกำหนดค่าเริ่มต้นต่างๆให้กับบอร์ด เมื่อเริ่มต้นการทำงาน Arduino จะทำงานคำสั่งต่างๆที่อยู่ใน void setup() ก่อน 1 รอบ หลังจากนั้นจะเข้าสู่ void loop() ต่อไป การกำหนดรูปแบบ ใหม่การทำงานของสัญญาณในแต่ละขาที่จะใช้งานของไมโครคอนโทรลเลอร์ว่ามีหน้าที่อะไรนั้นแบ่งออกเป็นสองรูปแบบคือ สัญญาณขาเข้าหรือเรียกว่าสัญญาณอินพุต (input) และสัญญาณขาออกหรือเรียกว่าสัญญาณเอาท์พุต (output) ซึ่งเราต้องกำหนด ใหม่การทำงานของมันเสียก่อนด้วยคำสั่ง pinMode(pin, mode) โดย pin เป็นหมายเลขของบอร์ดไมโครคอนโทรลเลอร์ และ ในส่วนของ mode เป็นการเลือกใหม่การการทำงานของขาที่เป็นชนิดใดชนิดหนึ่ง

INPUT	กำหนดให้ขาที่เป็น input
OUTPUT	กำหนดให้ขาที่เป็น output
INPUT_PULLUP	กำหนดให้ขาที่เป็น input ที่มีการต่อ internal resistor แบบ pull-up

การทำงานของ void loop() เป็นฟังก์ชันสำหรับสั่งให้ Arduino ทำงานคำสั่งต่างๆที่เราเขียนไว้วนรอบซ้ำกันไป โดย จะเริ่มต้นทำงานเมื่อผ่านจาก void setup() มาแล้ว

การทดลองจะเขียนโปรแกรมควบคุมโดยใช้ Arduino ต่อ กับ Protoboard แล้วทำการเชื่อมต่ออุปกรณ์ที่ใช้ในการ ทดลองลงบนโปรแกรมบอร์ดังรูป และให้ขาดิจิตอลที่ D2 ของ Arduino ส่งค่าสัญญาณเอาท์พุตแบบดิจิตอลออกมา ด้วยคำสั่ง digitalWrite(pin,value) โดยที่ค่า value ของสัญญาณดิจิตอลที่ได้มีอยู่ 2 รูปแบบคือ สัญญาณ HIGH และ LOW

เมื่อออยู่ในสถานะ “HIGH” ของ Arduino Nano จะส่งแรงดันไฟฟ้าขนาด 5 โวลต์ออกมานา

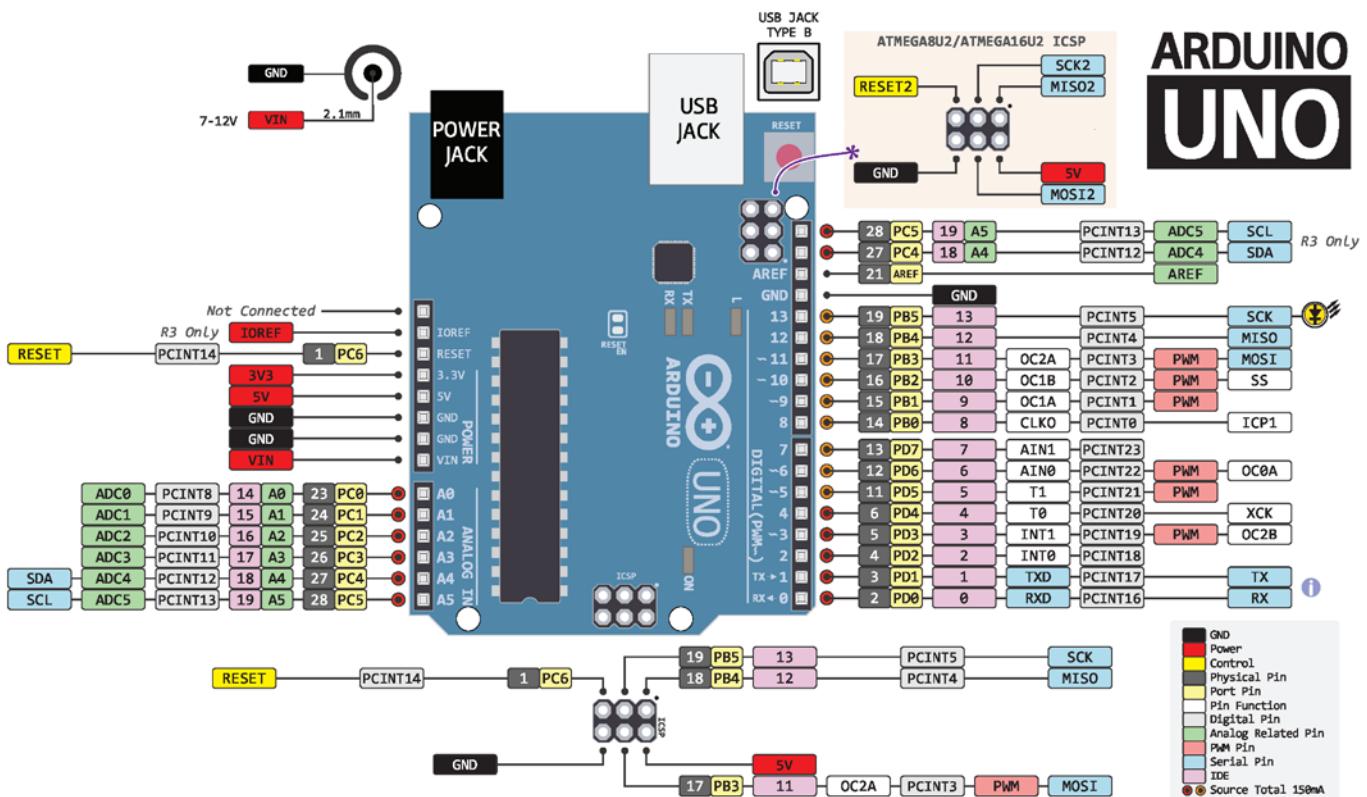
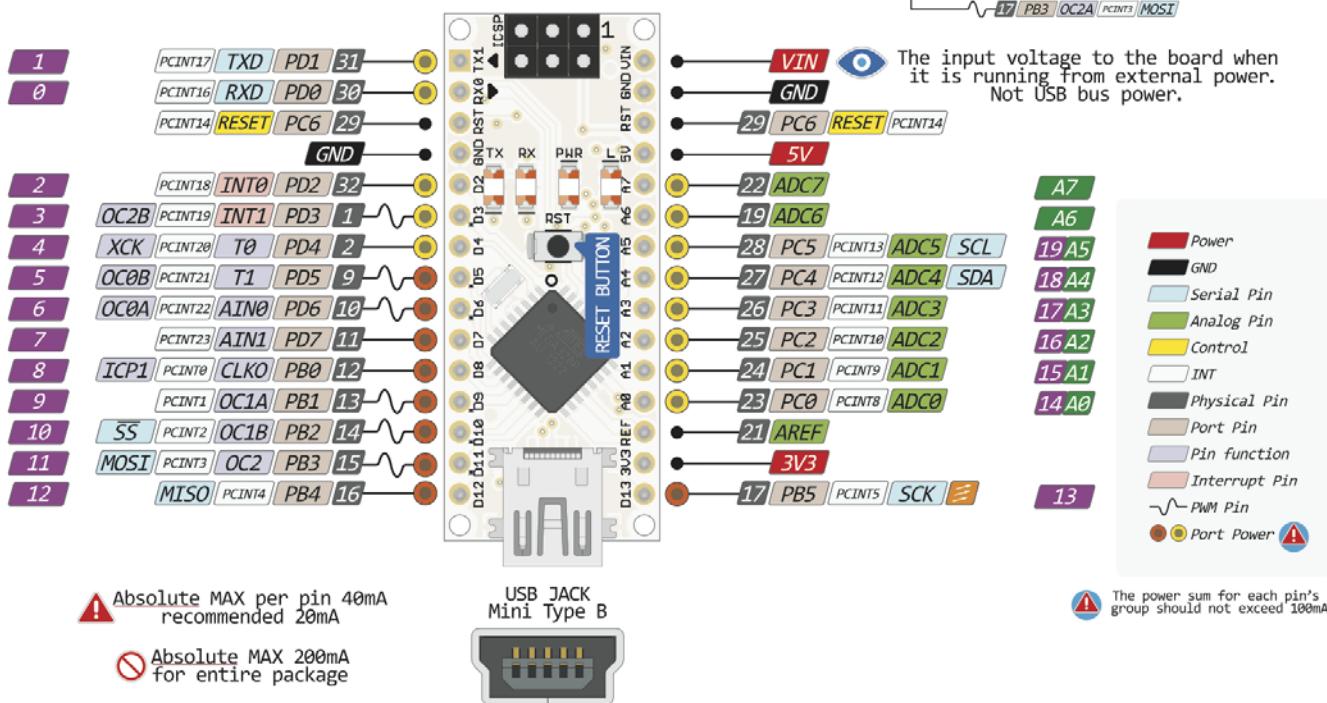
เมื่อออยู่ในสถานะ “LOW” ของ Arduino Nano จะซ่อนต่อกับ Ground (GND)

แรงดันไฟฟ้าที่ไมโครคอนโทรลเลอร์ส่งออกมานา เมื่อเรารสั่ง HIGH นั้นขึ้นอยู่กับรุ่นของบอร์ดไมโครคอนโทรลเลอร์ที่ เราใช้งาน หากเป็นบอร์ดไมโครคอนโทรลเลอร์ที่ใช้แหล่งจ่ายไฟ 3.3 โวลต์ การสั่ง HIGH จะเป็นการสร้างสัญญาณ 3.3 โวลต์ ออกมานาที่ขาที่ 5 และหากเป็นไมโครคอนโทรลเลอร์ที่ใช้ไฟ 5 โวลต์ สัญญาณ HIGH ที่ออกมานาก็จะเป็น 5 โวลต์

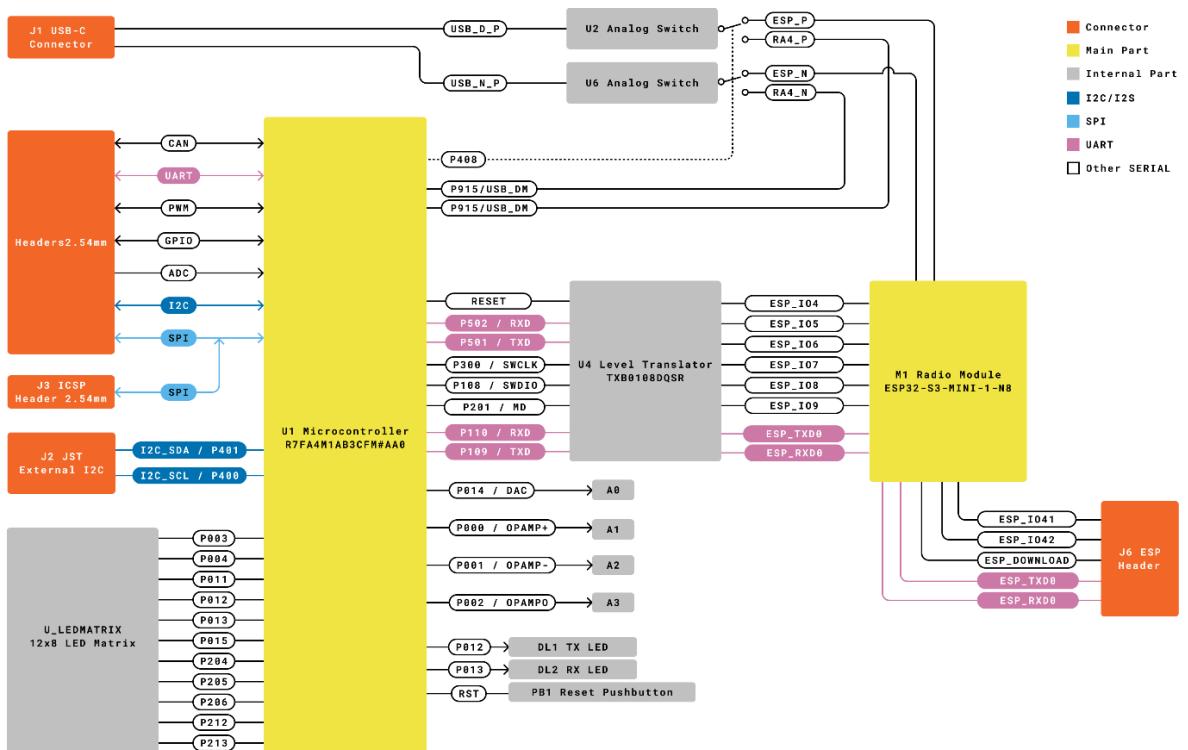
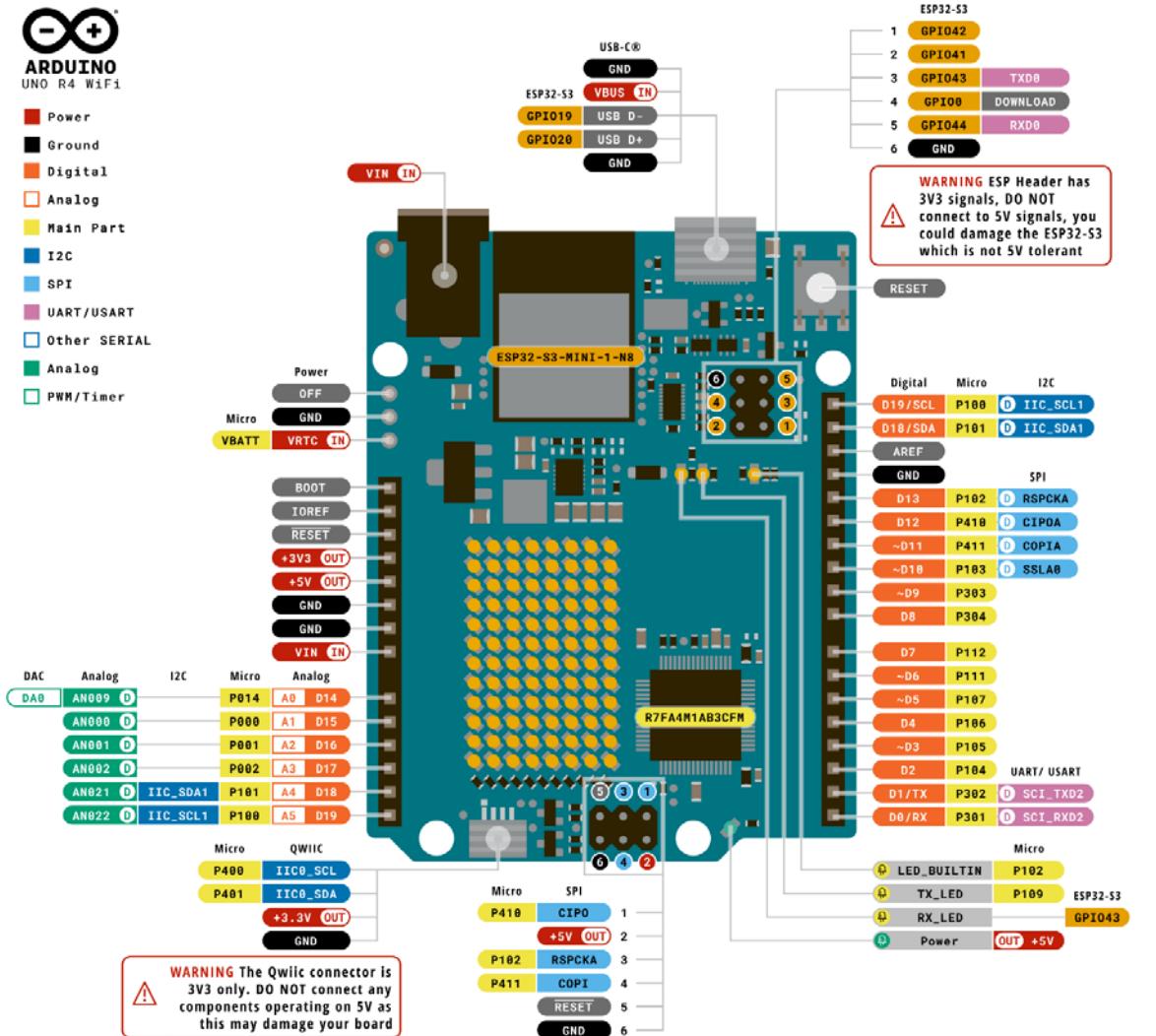
วงจรและตำแหน่งขาต่างๆของบอร์ด Arduino ที่ใช้ในการทดลองเป็นดังนี้

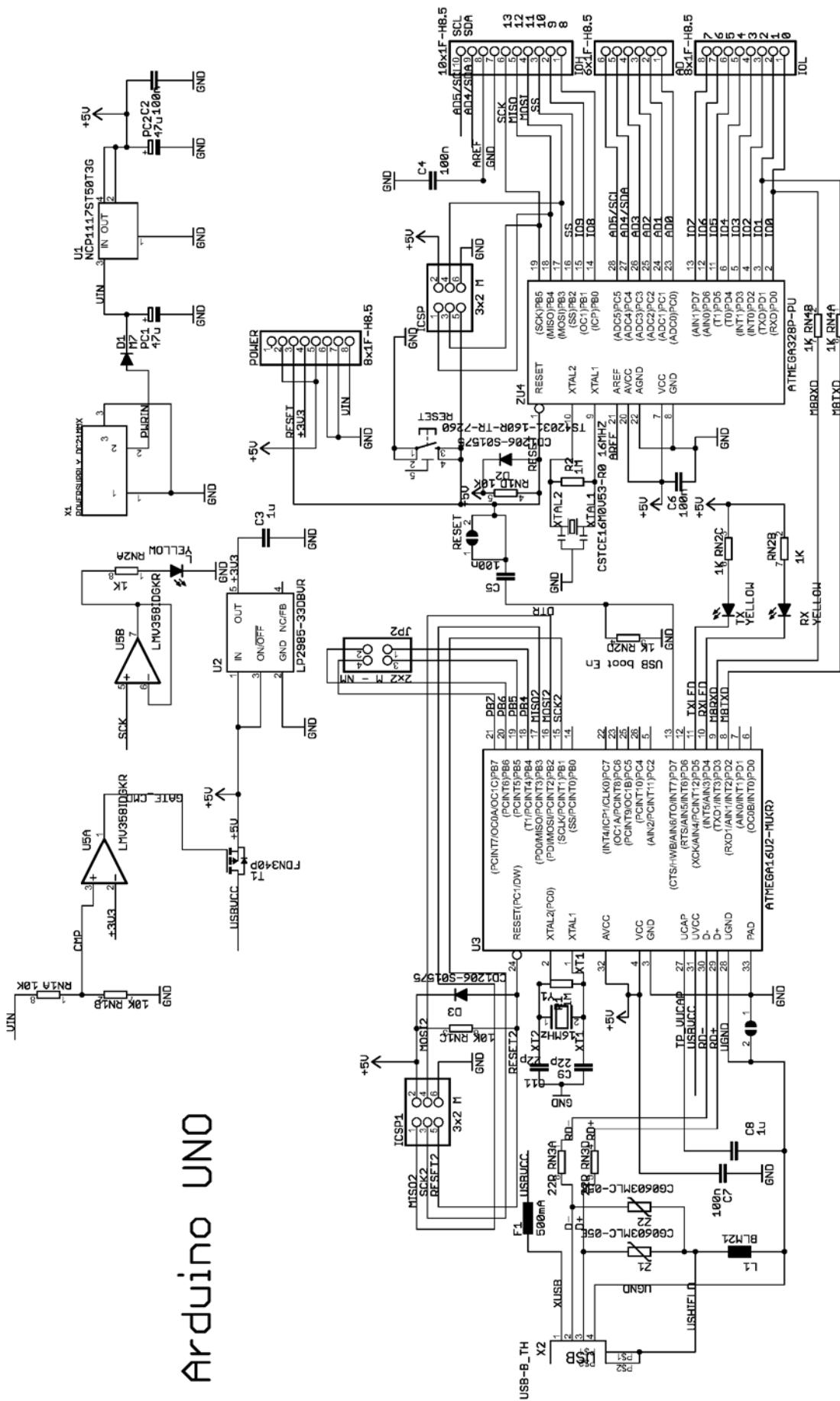
# 01236255 INTRODUCTION TO INTERNET OF THINGS

## NANO PINOUT



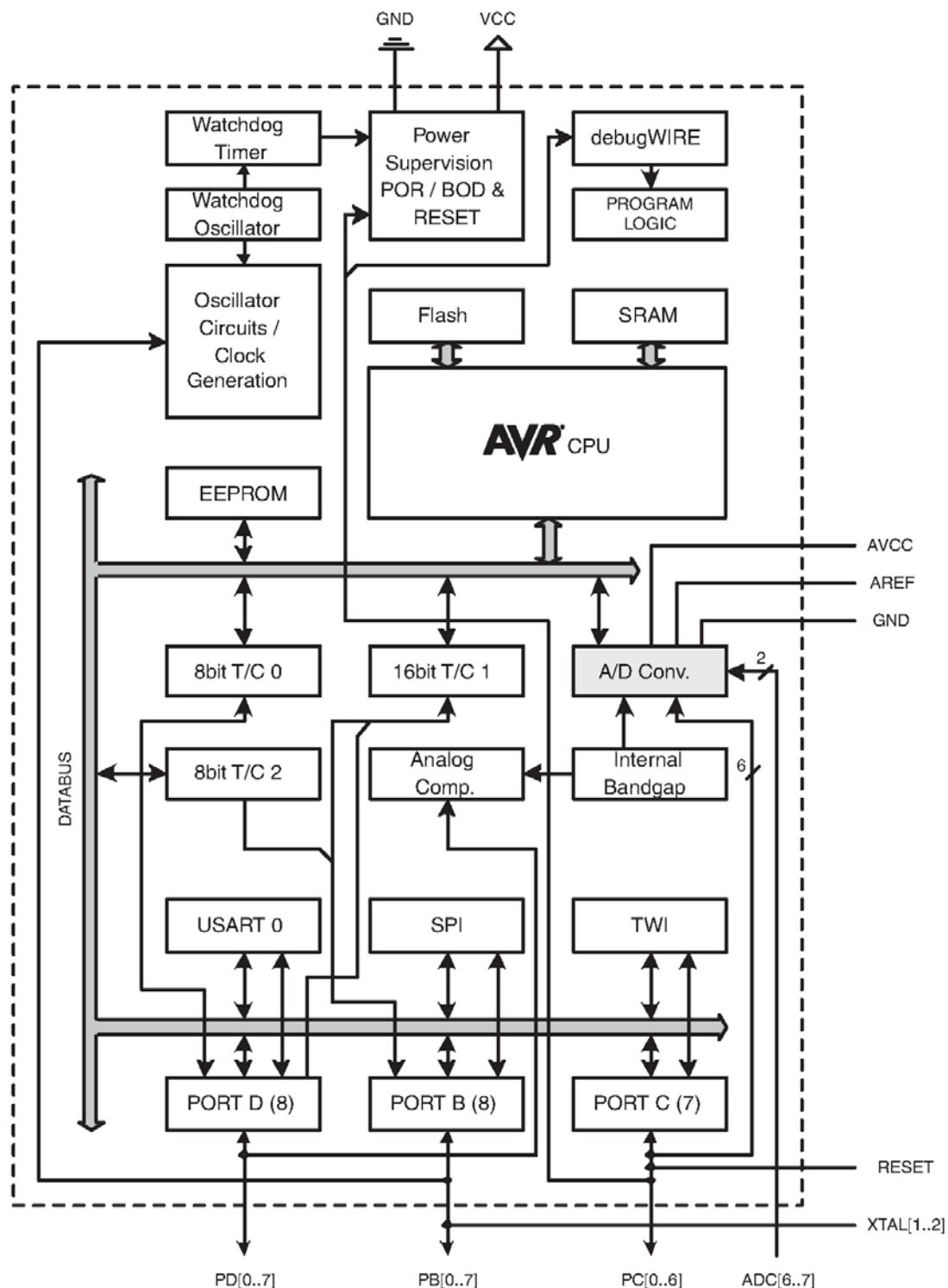
01236255 INTRODUCTION TO INTERNET OF THINGS





ภายในบอร์ด Arduino ที่ใช้ในการทดลองจะใช้ตัวประมวลผลเป็น Microcontroller AVR ขนาด 8 bit เป็นชิป ATmega328P ใช้สถาปัตยกรรม (Architecture) แบบ Reduced Instruction Set Computer (RISC) ที่มีชุดคำสั่งที่สั้นและมีจำนวนคำสั่งไม่มากนัก สามารถกระทำการตามคำสั่งได้อย่างรวดเร็ว ตรงข้ามกับ Complex Instruction Set Computer (CISC) รูปด้านล่างจะเป็น Block Diagram ภายในของไมโครคอนโทรลเลอร์นี้ รายละเอียดต่างๆเพิ่มเติมให้เปิดคู่ได้จากเอกสาร ATMEGA328 DataSheet

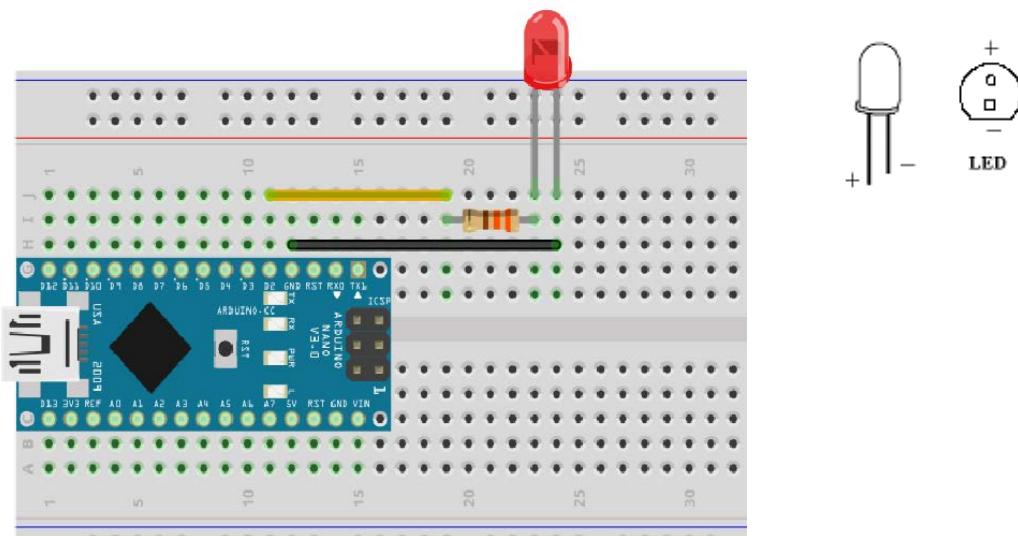
Block Diagram ของ ATmega328P



## 01236255 INTRODUCTION TO INTERNET OF THINGS

การใช้คำสั่ง digitalWrite กับขาที่เลือกโหมดเป็น input จะเป็นการปิด-ปิด การต่อ pull-up ภายในวงจรของไมโครคอนโทรลเลอร์ โดย HIGH เป็นการปิดโหมดการต่อ Pull-up และ LOW เป็นการปิดโหมด pull-up

ในกรณีการใช้คำสั่ง digitalWrite เพื่อสั่งให้ LED สว่าง โดยที่ไม่ได้กำหนดโหมดการทำงานของขาด้วยคำสั่ง pinMode จะส่งผลให้ LED ที่ต่ออยู่กับขาบันทึกไม่สว่างเท่าที่ควร เพราะการไม่ใช้คำสั่ง pinMode จะเปิดการทำงานโหมด pull-up ซึ่งจะทำให้กระแสไฟบางส่วนไหลผ่านตัวด้านทานภายในโดยไม่ผ่าน LED



รูปการเชื่อมต่ออุปกรณ์การทดลองลงบนโปรโตบอร์ด

- ให้เชื่อมต่อสาย USB ของบอร์ดกับคอมพิวเตอร์ เปิดโปรแกรม Arduino ที่ได้ติดตั้งในคอมพิวเตอร์ จากนั้นทำการเพียงโปรแกรมที่ทำหน้าที่สั่งงานให้ LED ที่อยู่บนบอร์ดไมโครคอนโทรลเลอร์กระพริบทุก 1 วินาที จากนั้นทำการคอมไฟล์แล้วทำการ Upload โปรแกรมที่ได้ลงบนบอร์ด Arduino

```
int led = 13; // LED connected to digital pin 13

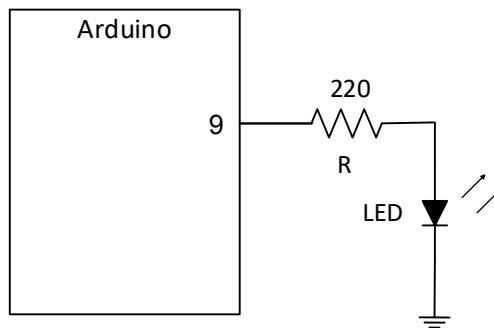
void setup()
{
    pinMode(led, OUTPUT); // initialize the digital pin as an output
}

void loop()
{
    digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000); // wait for a second (1000 milliseconds)
    digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
    delay(1000); // wait for a second (1000 milliseconds)
}
```

คำสั่งที่ใช้มีความหมายดังนี้

int led = 13;	ใช้ในการกำหนดขาที่ต่อ LED ภายในบอร์ด Arduino ว่าต้องยังที่ขา 13
pinMode(led, OUTPUT);	กำหนดให้ขาที่ต่อ LED เป็นขาเอาท์พุท
digitalWrite(led, HIGH);	ให้สั่งค่าลอจิก 1 ออกไปทางที่ต่อ กับ LED
digitalWrite(led, LOW);	ให้สั่งค่าลอจิก 0 ออกไปทางที่ต่อ กับ LED
delay(1000);	ให้ทำการหน่วงเวลา 1000 ms (Milliseconds)

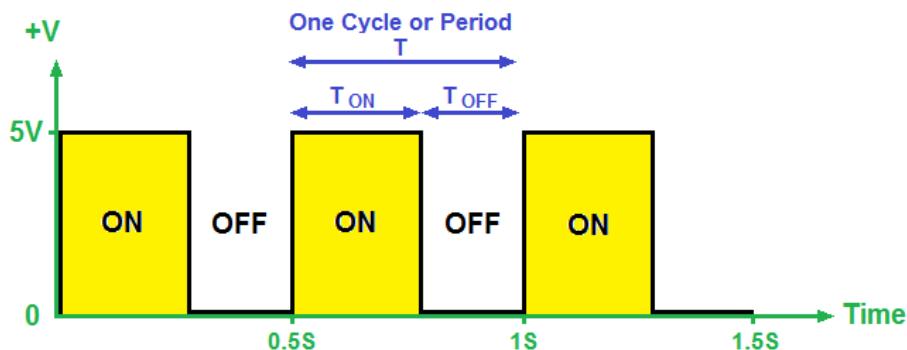
2. ให้ทำการข้ามขา LED ของโปรแกรมที่ ต่ออยู่ขาที่ 13 ไปเป็นขาที่ 9 และให้ต่อ LED อนุกรมกับตัวความค้านทาน 220 Ω เข้ากันขาที่ 9 แล้วลงกราวด์



3. จากข้อ 1 ให้แก้ไขโปรแกรมให้ LED กระพริบเป็นความถี่ 10 Hz

โดยที่ความถี่ (Frequency) เป็นจำนวนรอบที่แสดงว่าคลื่นเคลื่อนที่ไปได้กี่รอบในหนึ่งวินาที (Second) มีหน่วยเป็น รอบต่อวินาทีหรือเอร็ตซ์ (Hz) ใช้แทนสัญลักษณ์ด้วย  $f$

ค่าเวลากา (Period) คือ เวลาที่ใช้ในการเคลื่อนที่ครบ 1 รอบ (One Cycle) มีหน่วยเป็นวินาที (Second) ใช้แทนสัญลักษณ์ด้วย  $T$



จากรูปเมื่อเวลาผ่านไปหนึ่งวินาที คลื่นเคลื่อนที่ได้สองถูก แสดงว่าคลื่นนี้มีความถี่ 2 Hz

หรือถ้าพิจารณาจากความเวลาจะเห็นว่าใน 1 รอบจะใช้เวลา  $T = T_{on} + T_{off} = 0.5 \text{ Sec}$

ดังนั้นจะได้ความสัมพันธ์ระหว่างความถี่ ( $f$ ) และความ ( $T$ ) ตามสมการ

$$f = \frac{1}{T}$$

$$f = \frac{1}{0.5}$$

$$f = 2 \text{ Hz}$$

$$f = \frac{1}{T}$$

$$10 = \frac{1}{T}$$

$$T = \frac{1}{10} = 0.1 \text{ s} = 0.1 \times 10^3 \text{ ms} \\ = 100 \text{ ms}$$

$$T = T_{on} + T_{off}$$

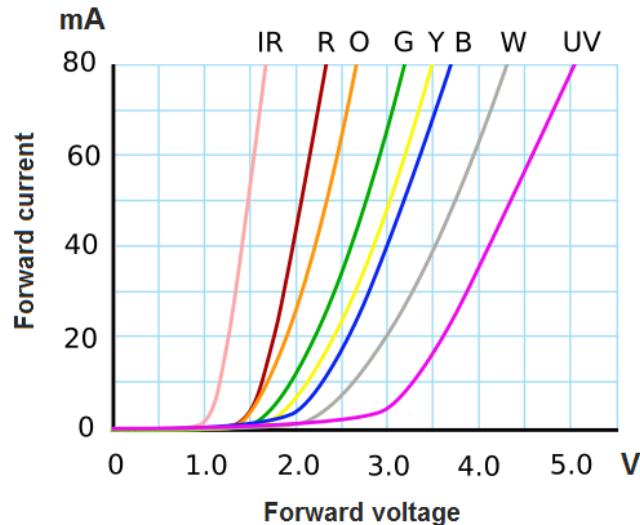
$$100 = T_{on} + T_{off}$$

$$\therefore T_{on}, T_{off} = 50 \text{ ms}$$

4. ให้วัดแรงดันไฟฟ้าต่อกว่าอมด้า LED ในช่วงที่ LED กำลังทำงาน (LED ON) โดยที่ LED จะต้องต่ออนุกรมกับตัวความค้านทาน 220 Ω

LED	Forward Voltage
infrared (IR)	1.19 V
Red	2.03 V
Green	2.05 V
Blue	2.82 V
Ultraviolet (UV)	3.09 V

ค่าแรงดันไฟฟ้าที่ต่อกลาง LED ขณะป้อนแรงดันไฟฟ้าแบบตรงตามข้อ (Forward) จะแปรผันตามกระแสที่ไฟหล่อผ่าน และจะขึ้นอยู่กับค่าความขาวคลื่นของแสงที่ส่องสว่างออกมาจาก LED ด้วย ดังตัวอย่างจะเป็นกราฟแสดงค่า Characteristic ของ LED แต่ละตัว



5. ให้คำนวณหากระแสที่ไฟหล่อผ่าน LED ที่ใช้ทุกดalon มา 2 ตัวในช่วงขณะที่ LED กำลังทำงาน (LED ON) โดยใช้กฎของโอล์ม

$$V = IR$$

$$I = \frac{V}{R}$$

$$I = \frac{5 - V_{LED}}{220}$$

$$I = \frac{5 - 2.03}{220}$$

$$I = 0.0135 \text{ A}$$

(Red)

$$V = IR$$

$$I = \frac{V}{R}$$

$$I = \frac{5 - V_{LED}}{220}$$

$$I = \frac{5 - 2.82}{220}$$

$$I = 0.01 \text{ A}$$

(Blue)

กฎของโอห์ม (Ohm's Law) กล่าวไว้ว่ากระแสไฟฟ้าที่ไหลในตัวนำไฟฟ้าจะแปรผันตามแรงดันที่ต่อกลางตัวนำนั้น และจะแปรผกผันกับค่าความต้านทานของตัวนำนั้น ดังสมการ

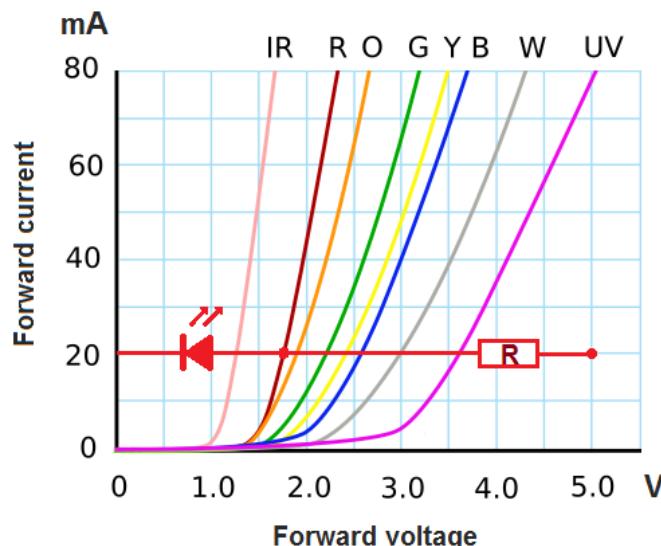
$$I = V / R$$

เมื่อ  $I$  = กระแสไฟฟ้ามีหน่วยเป็นแอมป์เรียร์ (A)

$V$  = แรงดันไฟฟ้ามีหน่วยเป็นโวลต์ (V)

$R$  = ความต้านทานมีหน่วยเป็นโอห์ม ( $\Omega$ )

โดยที่แรงดันต่อกลางตัวนำความต้านทานได้จากแรงดันของแหล่งจ่ายไฟบวกด้วยแรงดันต่อกลาง LED มีความสัมพันธ์ดังรูป



6. ให้ทดลองทำการเปลี่ยนค่าความต้านทานจาก  $220\ \Omega$  ไปเป็น  $1K\ \Omega$  และให้อธิบายผลที่ได้เป็นอย่างไร

**แสงส่องแดงของ LED จะสว่างมากขึ้นกว่าเดิมและ LED คงอยู่**

7. ให้แสดงวิธีการคำนวณหาค่าความต้านทานที่เหมาะสม เมื่อกำหนดให้ Forward Current ของ LED เท่ากับ  $20\ mA$

$$V = IR$$

$$2.97 = 90 \times 10^{-3} R$$

$$V_{LED} = 9.03\ V \quad (\text{Red})$$

$$V_{Th} - V_{LED} = IR$$

$$9 - 9.03 = 90 \times 10^{-3} R$$

$$5 - 9.03 = (90 \times 10^{-3}) R$$

$$R = 148.5\ \Omega$$

8. ให้ทำการแก้ไขโปรแกรมโดยการเปลี่ยนค่า `delay()` เพื่อให้ LED ติดสว่าง  $0.5$  วินาที และดับ  $1.5$  วินาที จากนั้นให้ LED กระพริบเร็วขึ้นเรื่อยๆ ตามลำดับจนกว่าเราจะไม่เห็นการกระพริบ โดยใช้คำสั่ง `for (....)`

9. จากข้อ 8 ค่าความถี่ในขณะที่เริ่มจะไม่เห็น LED กระพริบคือความถี่เท่าไร

$$T_{on} + T_{off} = T = 12\ ms \quad ; \quad f = \frac{1}{T} = \frac{1}{12 \times 10^{-3}} = \frac{1000}{12} = 83.3\ Hz$$

10. จากข้อ 8 ค่าความสว่างของ LED ในขณะที่เริ่มจะไม่เห็น LED กระพริบ ความสว่างนี้เท่าเดิมหรือน้อยลง และให้เหตุผลว่าทำไมจึงเป็นเช่นนั้น

```

1 //P
2 int led = 13;
3
4 void setup()
5 {
6     pinMode(led, OUTPUT);
7 }
8
9 void loop()
10 {
11     for (int i = 0; i < 500; i++)
12     {
13         digitalWrite(led, HIGH); // 0 -> T_on
14         digitalWrite(led, LOW); // 0 -> T_off
15         delay(1500 - 1*i); // ไม่แน่นอน
16         // T_on + T_off = 3 + 9 = 12ms
17         // T_on = 12ms เป็นเวลาที่ต้องกระโดดในเวลาระยะหนึ่ง ซึ่งถูกใช้สำหรับเพิ่มเวลาเป็น 12ms
18         // อาจเป็นสาเหตุที่ทำให้ลดลง
19     }
20 }

```

สูญเสียสัญญาณสีแดงของ LED หายไป

## ความล่าสุดที่ทำ成..... เนื่องจากใช้แบตเตอรี่ไฟฟ้าและกำเนิดเท่าเดิม.....

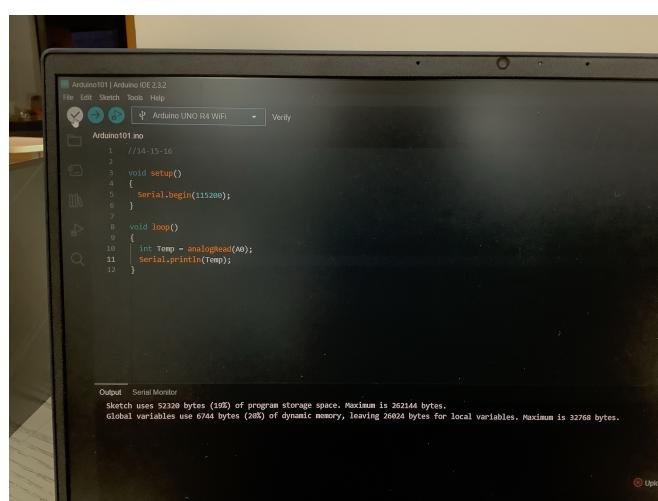
11. ให้ต่อ LED หลอดที่ 2 อนุกรมกับตัวความด้านทาน  $220\ \Omega$  เชื่อมเข้ากับขาที่ 10 แล้วลงกราฟ์  $\frac{\text{on } 0.5s}{\text{off } 0.5s} > T_3 = \frac{1}{f} = \frac{1}{1\ Hz}$   $\frac{\text{on } 0.5s}{\text{off } 0.25s} > T_{10} = \frac{1}{f} = \frac{1}{2\ Hz}$
12. ให้เขียนโปรแกรมให้ LED ขาที่ 9 กระพริบเป็นความถี่  $1\ Hz$  และให้ LED ขาที่ 10 กระพริบเป็นความถี่  $2\ Hz$
13. ให้ต่อ LED อนุกรมกับตัวความด้านทาน  $220\ \Omega$  เพิ่มอีกเป็นจำนวน 5 หลอด แล้วให้เขียนโปรแกรมควบคุมให้หลอดไฟ LED กระพริบໄล่จากขวาไปซ้าย แล้วกระพริบໄล่จากซ้ายสุดและขวาลับกันไปมา โดยใช้คำสั่ง `for(....)`

การสื่อสารกับอุปกรณ์ภายนอกของบอร์ด Arduino จะใช้พอร์ตที่เรียกว่าพอร์ตอนุกรม (Serial Port) ในการเชื่อมต่อ กับอุปกรณ์อื่น หรือสื่อสารระหว่างไมโครคอนโทรลเลอร์กับคอมพิวเตอร์ การสื่อสารนี้เรียกว่า **UART** โดยจะใช้ขาหมายเลข 0 (RX) ในการรับค่า และขาหมายเลข 1 (TX) ในการส่งค่า คำสั่งต่างๆที่จำเป็นมีดังนี้

- ✓ `void serial.begin(rate)` เป็นการกำหนดอัตราของการรับส่งข้อมูล หน่วยเป็น bits per second (baud rate)
- `int serial.available()` ใช้ตรวจสอบว่าบอร์ดมีข้อมูลใหม่เข้ามายังพอร์ตอนุกรมของไมโครคอนโทรลเลอร์
- `int serial.read()` อ่านค่าข้อมูลที่ถูกส่งเข้ามายังพอร์ตอนุกรมของไมโครคอนโทรลเลอร์
- `void Serial.flush()` เคลียร์บัฟเฟอร์ของพอร์ตอนุกรมให้ว่าง
- ✓ `void Serial.print()` พิมพ์ข้อมูล ออกทางพอร์ตอนุกรม
- ✓ `void Serial.println()` พิมพ์ข้อมูล ออกทางพอร์ตอนุกรมและขึ้นบรรทัดใหม่

14. ให้เพิ่มคำสั่ง `Serial.begin(115200); // initialize serial communication at 115200 bits per second` ลงใน `void setup()` เพื่อใช้กำหนดอัตราความเร็วในการรับส่งข้อมูลผ่าน Serial Monitor มีค่าเท่ากับ  $115200\ bps$
15. ให้เพิ่มคำสั่ง `int Temp = analogRead(A0); // read the input on analog pin 0` ลงใน `void loop()` เพื่อใช้รับค่าสัญญาณ온าล็อกจากขา A0 ของบอร์ด Arduino และแปลงค่าที่ได้ไปเป็นสัญญาณดิจิตอลขนาด 10 บิต แล้วเก็บไว้ที่ตัวแปร Temp ซึ่งค่าที่ได้จะอยู่ระหว่าง 0 ถึง 1023 (จำนวนได้จาก  $2^{10}$ )
16. ให้เพิ่มคำสั่ง `Serial.println(Temp); // print out the value` ต่อจากคำสั่งในข้อ 15 เพื่อให้พิมพ์ผลลัพธ์ค่าข้อมูลตัวแปร Temp ส่งออกไปทาง Serial Monitor

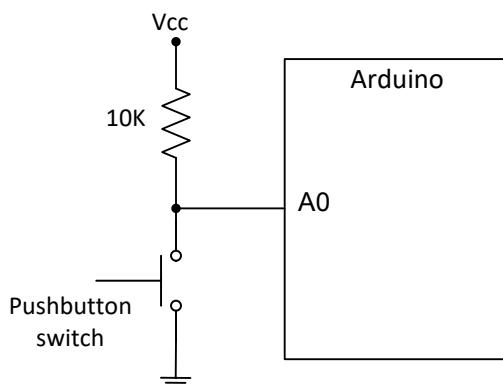
14,15,16 →



## 01236255 INTRODUCTION TO INTERNET OF THINGS

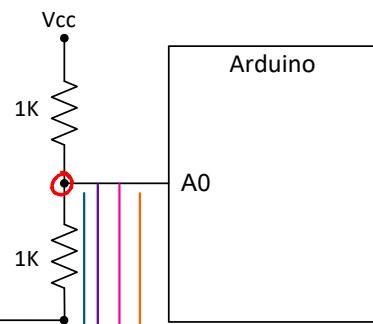
การทดลองเขียนโปรแกรมเพื่อสั่งงานให้ในโครค่อนโตรลเลอร์อ่านค่าสัญญาณของขาที่ทำการเชื่อมต่ออยู่กับวงจรที่เป็นอุปกรณ์ภายนอก เมื่อมีการกำหนดให้ขาใดขาหนึ่งของไมโครค่อนโตรลเลอร์ทำหน้าที่เป็น Input ด้วยคำสั่ง pinMode ก็สามารถใช้คำสั่ง digitalRead เพื่อสั่งให้ในโครค่อนโตรลเลอร์อ่านค่าสัญญาณที่เป็นแบบดิจิตอลเข้ามายังอุปกรณ์ที่เชื่อมต่ออยู่กับขาที่นั้นๆ ได้ ด้วยการใช้คำสั่ง digitalRead(pin) โดยที่ pin เป็นค่าของหมายเลขดิจิตอลที่ต้องการอ่านค่าว่าเป็นสัญญาณ HIGH หรือ LOW ในบอร์ด Arduino จะมีขาที่มีวงจรที่ทำหน้าที่แปลงสัญญาณสัญญาโนนาลีอคไปเป็นสัญญาณดิจิตอลหรือ Analog to Digital Converter (ADC) ขนาด 10 บิต ซึ่งจะใช้ในการอ่านค่าของสัญญาณที่เป็นแบบอนาลีอคเข้ามายังวงจรภายนอกหรือเซนเซอร์ต่างๆ ที่เป็นแบบอนาลีอคที่เชื่อมต่ออยู่ ซึ่งจะต้องใช้เป็นคำสั่ง analogRead(pin) โดยที่ pin จะเป็นหมายเลขอินพุตที่เป็นสัญญาโนนาลีอคซึ่งจะนับต้นด้วย A ใน Arduino จะมีขาที่เป็นอนาลีอคกอยู่ทั้งหมด 8 ขา ซึ่งค่าของสัญญาโนนาลีอคที่อ่านได้นี้จะต้องถูกแปลงค่าจากสัญญาโนนาลีอคไปเป็นสัญญาณดิจิตอลขนาด 10 บิต ทำให้ได้ค่าที่อ่านออกมาก้าทั้งหมดเท่ากับ  $2^{10}$  ซึ่งค่าที่ได้จะอยู่ในช่วง 0 ถึง 1023 นอกจากนี้แล้วยังมีขา Analog Reference ใช้สำหรับอ้างอิงค่า Analog ในการเบริญบเทียบเที่ยวนแรงดันแบบ Analog

17. ให้ต่อตัวความด้านทาน  $10\text{ K}\Omega$  อนุกรมกับสวิตช์ เข้ากับขา Vcc ของบอร์ด Arduino และลงกราวด์ โดยให้ขา A0 ที่ทำหน้าที่เป็น Analog to Digital Converter ต่อเข้ากับจุดต่อร่วมระหว่างตัวความด้านทานกับสวิตช์

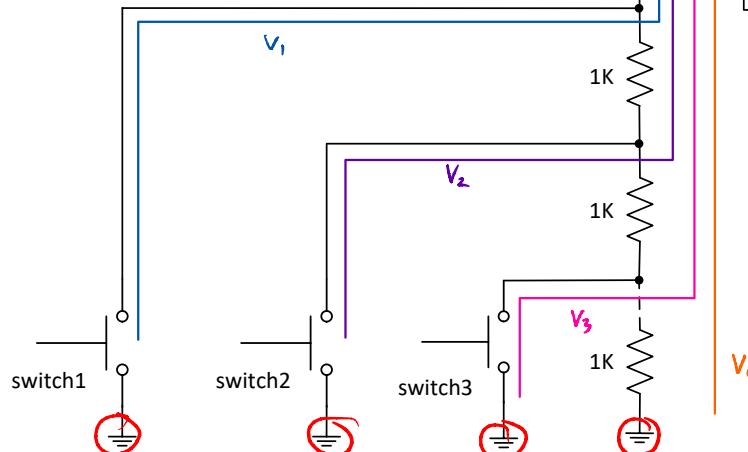


18. ให้เขียนโปรแกรมทดสอบการกดสวิตช์ โดยอ่านจากขา A0 และให้มันทึกค่าที่ได้  
เมื่อกดสวิตช์ Temp มีค่าเท่ากับ ..... 0 .....  
เมื่อปล่อยสวิตช์ Temp มีค่าเท่ากับ ..... 969 .....
19. ให้แก้ไขโปรแกรมในข้อ 18 โดยกำหนดให้มีกดสวิตช์ให้ LED ขาที่ 9 จะสว่าง และเมื่อปล่อยสวิตช์ให้ LED ขาที่ 10 สว่าง โดยใช้คำสั่ง if (...) else
20. ให้ทดลองต่อตัวความด้านทาน  $1\text{ K}\Omega$  จำนวน 5 ตัวอนุกรมกันแล้วต่อเข้ากับขา Vcc ของบอร์ด Arduino เพื่อทำวงจรแบ่งแรงดันไฟฟ้า (Voltage Divider) และให้ขา A0 ต่อเข้ากับจุดต่อร่วมระหว่างตัวความด้านทานจุดแรกแล้วให้ใช้สวิตช์ 3 ตัวต่อเข้ากับจุดต่อร่วมของตัวความด้านทานที่เหลือแล้วลงกราวด์ โดยกำหนดให้ค่าที่อ่านออกมากำไรไม่ให้ซ้ำกัน และมันทึกผลที่ได้  
เมื่อไม่กดสวิตช์ Temp มีค่าเท่ากับ ..... 819 .....  
เมื่อกดสวิตช์ตัวที่ 1 Temp มีค่าเท่ากับ ..... 519 .....  
เมื่อกดสวิตช์ตัวที่ 2 Temp มีค่าเท่ากับ ..... 682 .....  
เมื่อกดสวิตช์ตัวที่ 3 Temp มีค่าเท่ากับ ..... 768 .....

- ① A0 - GND 97% ตัวน้อย ( $V = V_T$ )  
 ② Vcc - A0 เต่า 5 伏ต ( $V = 5 - V_T$ )



$$V = \frac{R}{R_T} V_T$$



21. จากข้อ 20 ให้แสดงวิธีคำนวนหาค่า A0 ที่ได้จากการแบ่งแรงดันไฟฟ้า (Voltage Divider) เมื่อกำหนดเงื่อนไข  
ไว้ดังนี้

เมื่อไม่กดสวิตช์ A0 มีค่าแรงดันไฟฟ้าเท่าไร ?

A0 - GND : 3.742 V

$$V = \frac{4}{5} \times 5 = 4 V$$

เมื่อกดสวิตช์ตัวที่ 1 A0 มีค่าแรงดันไฟฟ้าเท่าไร ?

A0 - GND : 2.339 V

$$V = \frac{1}{2} \times 5 = 2.5 V$$

เมื่อกดสวิตช์ตัวที่ 2 A0 มีค่าแรงดันไฟฟ้าเท่าไร ?

A0 - GND : 3.118 V

$$V = \frac{2}{3} \times 5 = 3.3 V$$

10 ลิตรน้ำ = grav 9.81  
VCC - GND 10 ลิตรน้ำ

เมื่อกดสวิตช์ตัวที่ 3 A0 มีค่าแรงดันไฟฟ้าเท่าไร ?

A0 - GND : 3.507 V

$$V = \frac{3}{4} \times 5 = 3.75 V$$

## 01236255 INTRODUCTION TO INTERNET OF THINGS

22. ให้อธิบายว่าค่า Temp ในข้อ 20 กับค่า A0 ในข้อ 21 ว่ามีความสัมพันธ์กันอย่างไร และถ้ากำหนดให้ A0 ที่ได้จากวงจรแบ่งแรงดันไฟฟ้า (Voltage Divider) มีค่าเท่ากับ 2 V จะคำนวณหาค่าด้วย Temp ที่ได้จากขา Analog to Digital Converter ของ Arduino ว่าจะอ่านเข้ามาเป็นค่าเท่ากับเท่าไร

$$V = \frac{A0}{1023} \times V_T$$

$$A0 = 1023 \cdot \frac{V}{V_T}$$

$V = IR$  เมื่อ I เหลือที่ผ่าน R ที่นี่เกิด  $V$  โดย  $V \propto Temp$

$$\therefore V_{ao} = 2V ; Temp = 1023 \times \frac{2}{5} = 409.2$$

23. จากข้อ 20 ให้เขียนโปรแกรมที่มีข้อกำหนดคือ เมื่อกดสวิตช์ตัวที่ 1 ให้ LED ขาที่ 9 ติดสว่าง ถ้ากดสวิตช์ตัวที่ 2 ให้ LED ขาที่ 10 ติดสว่าง และถ้ากดสวิตช์ตัวที่ 3 ให้ LED ขาที่ 11 ติดสว่าง โดยค่าต่างๆที่นำมาเบริยบเทียบห้ามใช้จากตัวอ่อนแรงดังรูป

ถ้า 5v ได้

4v ได้

1023

$$1023 \times 4 / 5 = 818$$

5x1/2  
(511)  
2.5v

0-2.9v  
(0-593)

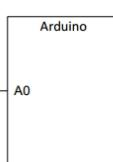
5x2/3  
(675)  
3.3v

2.9-3.5v  
(593-716)

5x3/4  
(757)  
3.7v

3.5-3.9v  
(716-787)

ไม่กด SW 4v (818)



ไม่กด SW

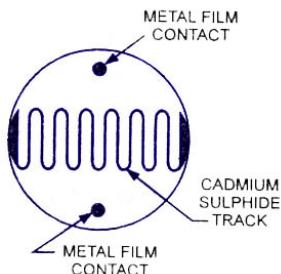
3.9-5v  
(787-1023)

และโปรแกรมจะใช้คำสั่ง switch (...) case หรือ if (...) else ก็ได้ ดังตัวอย่าง

```
void loop()
{
    Temp = analogRead(A0);
    Serial.println(Temp);
    if (Temp>787) // 3.9v (sw3=3.7v , no=4v)
        ...
        // ไม่กด sw (3.9v-5v) {787-1023}
    else
        // (<3.9v)
    if (Temp>716) // 3.5v (sw2=3.3v)
        ...
        // กด sw3 (3.5v-3.9v) {716-787}
    else
        // (<3.5v)
    if (Temp>593) // 2.9v (sw1=2.5)
        ...
        // กด sw2 (2.9v-3.5v) {593-716}
    else
        // (<2.9v)
        ...
        // กด sw1 (0v-2.9v) {0-593}
```

**LDR (Light Dependent Resistor)** เป็นตัวด้านท่านที่เปลี่ยนค่าความนำไฟฟ้าได้เมื่อมีแสงมากระทบ หรือเรียกว่าโฟโตริซิสเตอร์ (Photo Resistor) ทำมาจากสารกึ่งด้านนำ (Semiconductor) ประเทกแคนเดเมียมชัลไฟด์ (Cadmium Sulfide) หรือแคนเดเมียมเซเลนไนด์ (Cadmium Selenide) ซึ่งเป็นสารประเภทกึ่งด้านนำที่สามารถลงบนแผ่นเซรามิกที่ใช้เป็นฐานรองแล้วต่อขาจากสารที่ฐานนั้นออกมามีเมื่อมีแสงตกกระทบลงบนสารกึ่งด้านนำที่ฐานอยู่นี้จะถูกดูดพลังงานทำให้เกิดโคลกับอิเล็กตรอนอิสระวิ่งพล่านกันมากเป็นผลให้ค่าความด้านท่านลดลง

L ↑ R ↓

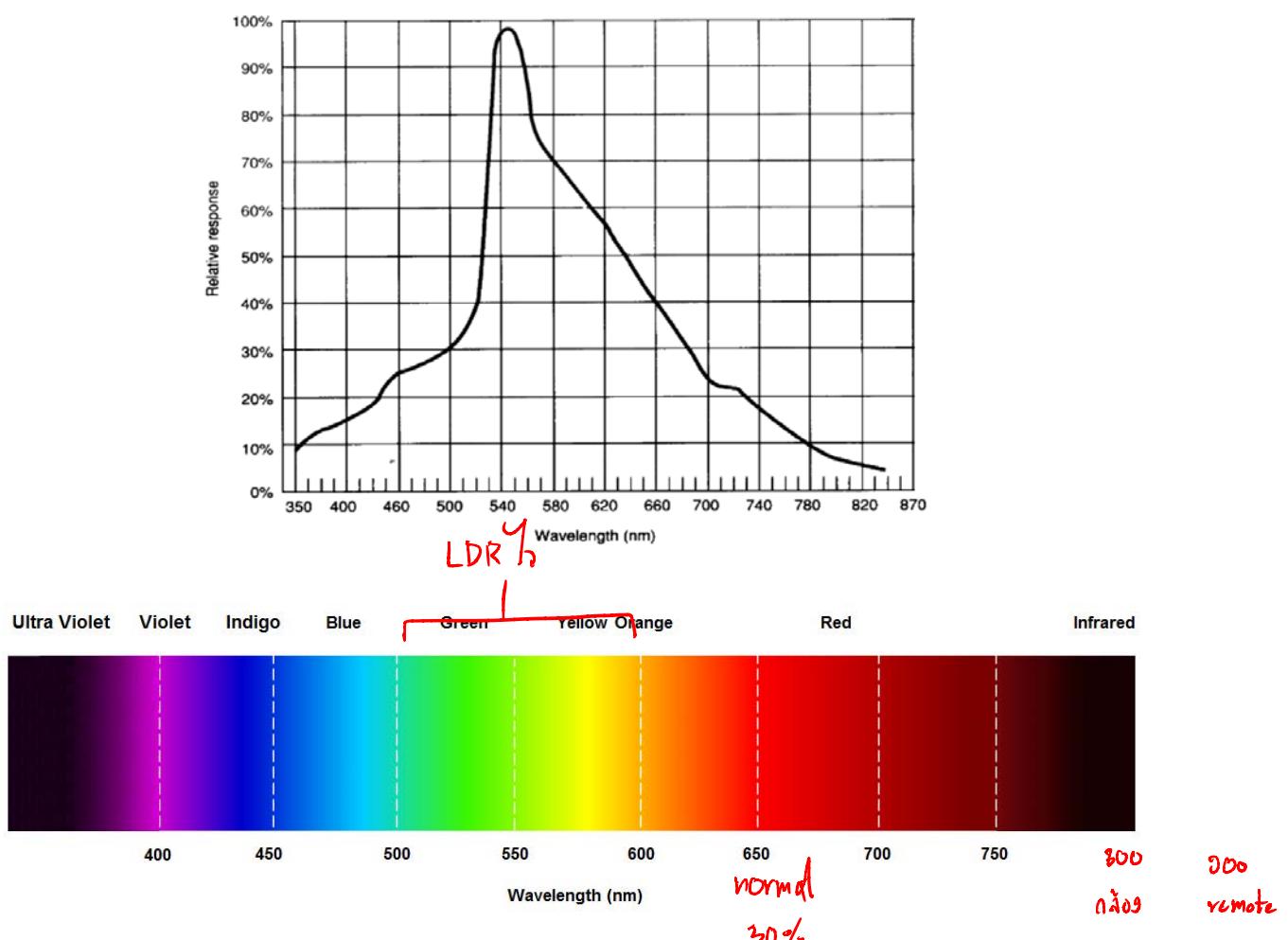


550 nm

LDR Basic Structure

LDR ที่ทำจากแคนเดเมียมชัลไฟด์ จะมีตอบสนองทางสเปกตรัมต่อแสงได้ดีที่ช่วงความยาวคลื่น (Wavelength) ประมาณห้าร้อยห้าสิบนาโนเมตร (nm) ซึ่งจะเป็นช่วงที่อยู่ระหว่างแสงสีเขียวกับสีเหลือง และในช่วงแสงสีแดงผลการตอบสนองจะลดลงเหลือเพียง 30% เมื่อเทียบกับแสงสีเขียว ดังแสดงในกราฟด้านล่าง

Spectral Response

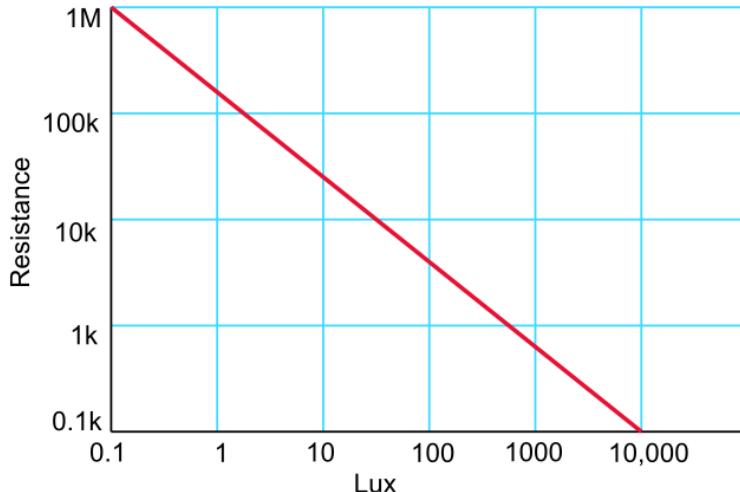


รับฟังก์ชัน

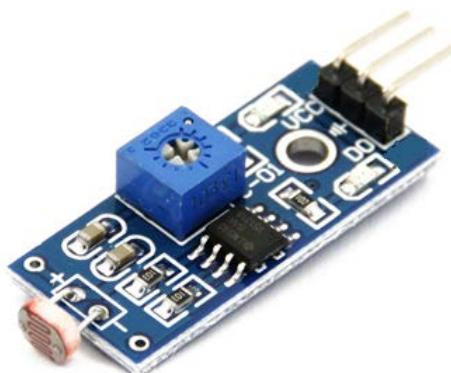
ทำงาน LDR  
ก็จะฟัง infrared

## 01236255 INTRODUCTION TO INTERNET OF THINGS

เมื่อไม่แสงมาต่อกลไกในสภาวะมีค่าทำให้ค่าความด้านท่านระหัวงหัวจะสูงถึง  $1\text{ M}\Omega$  หรือมากกว่านั้น ความด้านท่านจะลดลงตามระดับแสงที่เพิ่มขึ้นและจะลดลงเหลือไม่กี่ร้อยโอมที่ความสว่างสูงดังแสดงในรูปกราฟ แต่ อุปกรณ์ชนิดนี้ยังมีผลตอบสนองทางเวลา (Response time) ที่ไม่ดีนักคือจะช้ากว่าอุปกรณ์พากไฟโดยประมาณมาก โดย จะมีค่าอยู่ในช่วงประมาณ 2 ถึง 50 mSec รายละเอียดต่างๆเพิ่มเติมให้เปิดดูได้จากเอกสาร LDR 3190 DataSheet

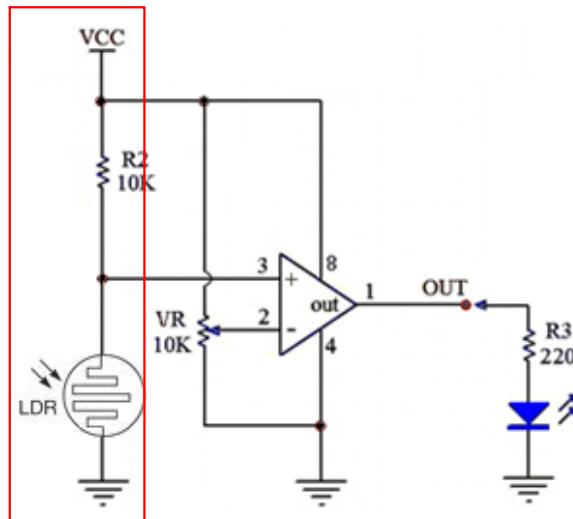


LDR มักนิยมใช้ในวงจรสวิทช์ที่เปิด-ปิดไฟด้วยแสง ตัวอย่างตามรูป วงจรภายในมักจะใช้ Op-Amp ทำเป็น วงจร Comparator โดยมีตัวด้านท่านปรับค่าได้ ทำหน้าที่ปรับระดับแสงที่ต้องการให้เปิดปิดไฟ

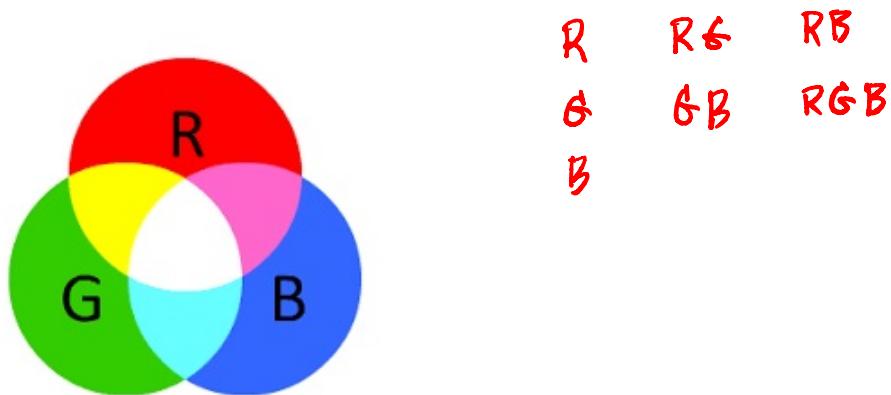


LDR Photoresistor Light Detection Sensor Module

24. การทดลองจะไม่ใช้ Op-Amp แต่จะใช้การเขียนโปรแกรมเพื่อเปรียบเทียบบน Arduino แทน โดยนำสัญญาณที่ต่อ เข้าขา 3 ของ Op-Amp ไปป้อนเข้า Analog to Digital Converter (ADC) ขาอ่านล็อกอินพุท A0 บนบอร์ด Arduino แทน ซึ่งจะทำให้ได้ค่าเป็นสัญญาณดิจิตอลขนาด 10 บิต ในช่วงระหว่าง 0 ถึง 1023 หรือก็คือได้ค่าทั้งหมดเท่ากับ  $2^{10}$

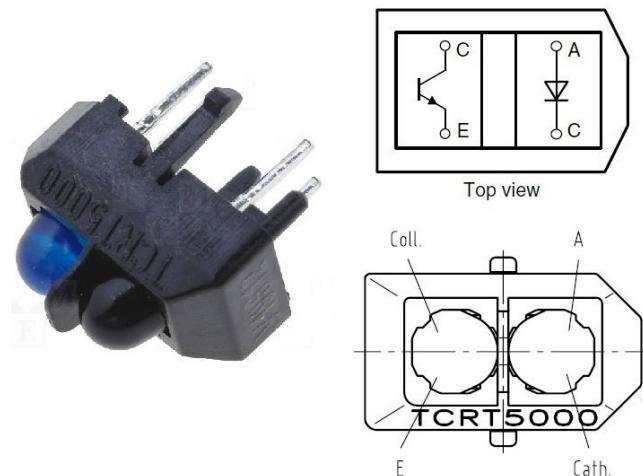


25. ให้นักศึกษาทำการแก้ไขโปรแกรมในข้อ 23 เพื่อทำเป็นเครื่องวัดความเข้มของแสงสว่าง โดยใช้ LDR เป็นเซ็นเซอร์วัดแสง และให้แสดงผลออกมารูปสีต่างๆ 7 สี ด้วย LED ที่เป็นแม่สี 3 สีคือ RGB โดย LED ที่ใช้ในการทดลองนี้จะต้องให้ขาของ LED ที่เป็นแม่สีทั้ง 3 ขาต่อกับตัวความต้านทาน  $220\ \Omega$  และต่อเข้ากับขา 9, 10, 11 ของ Arduino ตามลำดับ แล้วให้ขา Common Cathode ที่เป็นจุดร่วมซึ่งจะเป็นขาที่ขาวที่สุดให้ต่อลงกราวด์ (GND)

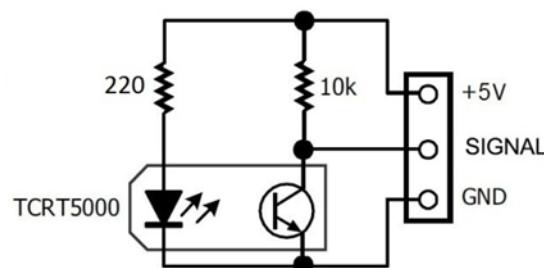


## 01236255 INTRODUCTION TO INTERNET OF THINGS

การทดลอง Sensor สำหรับตรวจจับแสง TCRT5000 โดยใช้ Infrared Emitting Diode ส่งแสงที่มีความยาวคลื่น 950 นาโนเมตรออกไป และมีตัวรับแสงใช้ Photo Transistor ทำหน้าที่รับแสงสะท้อนสามารถตรวจจับได้ที่ระยะห่างระหว่าง 0.2 ถึง 15 mm



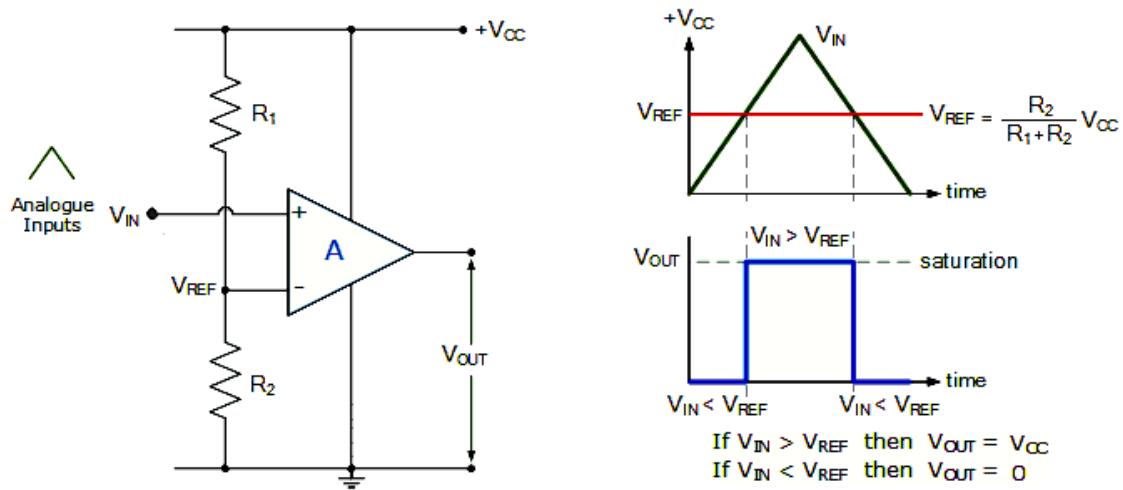
ใช้แรงดันไฟฟ้าในการทำงาน 5V ให้สัญญาณเอาท์พุตออกมาแบบอนาล็อก



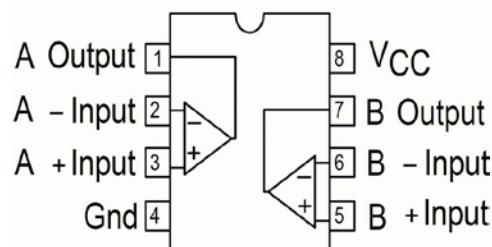
จากสัญญาณอนาล็อกที่ออกมาที่เอาท์พุตจะต้องแปลงให้เป็นสัญญาณดิจิตอล โดยใช้ Operational Amplifier เบอร์ LM358 ทำเป็นวงจร comparator ให้ป้อนสัญญาณอนาล็อกเข้าที่อินพุตไปยังขั้วบวกของ OpAmp ส่วนขั้วลบจะถูกต่อ กับความต้านทานที่ทำหน้าที่เป็นวงจรแบ่งแรงดัน (Voltage Dividers) ที่ประกอบด้วยความต้านทาน  $R_1$  และ  $R_2$  ต่ออนุกรมคร่อมแหล่งจ่ายไฟ 5V แรงดันจากแหล่งจ่ายไฟจะถูกแบ่งระหว่างความต้านทานทั้งสอง ให้แรงดันเอาท์พุตเป็น  $V_{REF}$  ซึ่งก็คือแรงดันไฟที่คร่อมที่  $R_2$  ค่าที่ได้เป็น

$$V_{REF} = \frac{R_2}{R_1 + R_2} V_{CC}$$

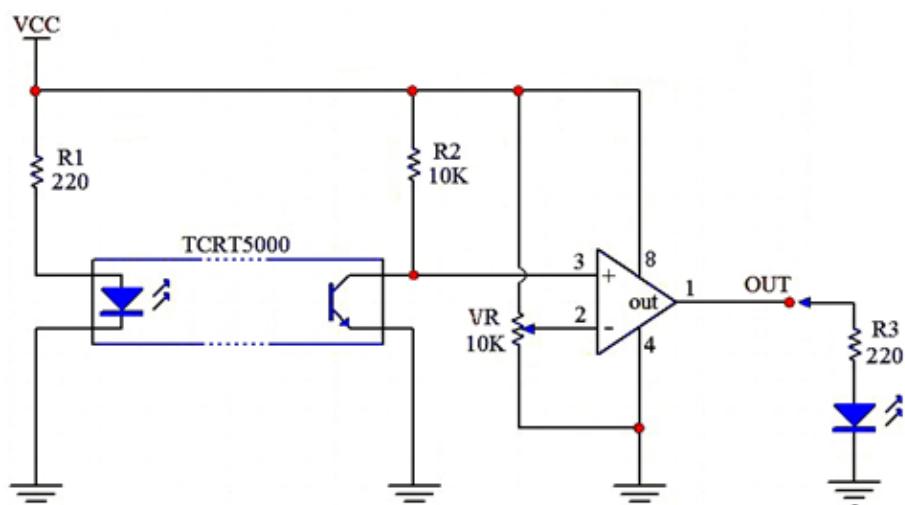
ถ้า  $V_{IN}$  มีค่ามากกว่า  $V_{REF}$  จะทำให้อเอาท์พุตออกมานเป็น โลจิก 1 และถ้า  $V_{IN}$  มีค่าน้อยกว่า  $V_{REF}$  จะให้อเอาท์พุตออกมานเป็น โลจิก 0 ดังแสดงในรูป



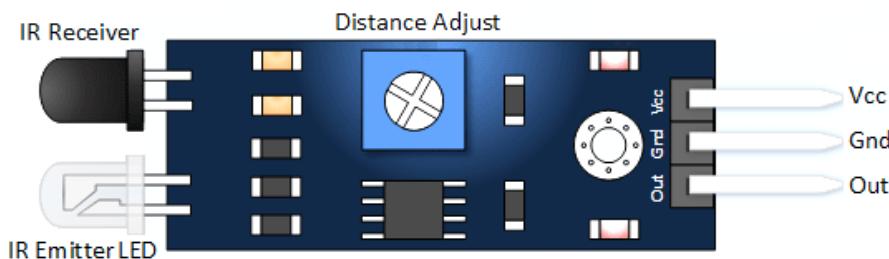
ตัวอย่างตำแหน่งขาของไอซี LM358



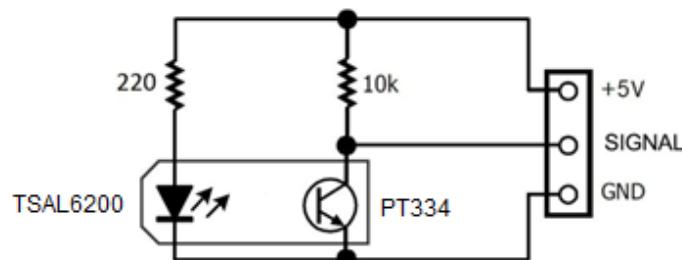
วงจรตามในรูป มีขั้วลงของ OpAmp ต่อ กับ ด้วยความต้านทานที่ปรับค่าได้ (VR) ขนาด 10 K เพื่อทำหน้าที่เป็นวงจรแบ่งแรงดัน (Voltage Dividers) ให้เป็น  $V_{REF}$



การทดลองนี้จะเน้นว่างจรเซ็นเซอร์ตรวจจับแสงที่ใช้ Op-Amp ทำเป็นวงจร Comparator มาแก้ไขโดยใช้ Infrared Emitting Diode TSAL6200 ที่ทำหน้าที่ส่งแสงความยาวคลื่น 940 นาโนเมตรออกไป และใช้ Phototransistor ชนิด NPN silicon เป็นอินฟารेनเซ็นเซอร์ ช่วง 840-1200 nm ทำหน้าที่รับแสงที่สะท้อนเข้ามาเพื่อใช้ในการวัดระยะห่าง รายละเอียดต่างๆของอุปกรณ์ทั้งสองชนิดนี้ให้เปิดคู่ได้จากเอกสาร Infrared Emitting Diode TSAL6200 DataSheet และ Phototransistor PT334 DataSheet



26. ให้ต่อวงจรเซ็นเซอร์ที่มีหน้าที่ตรวจจับแสงอินฟารेनเซอร์ที่สะท้อนเพื่อใช้ในการวัดระยะห่างตามรูปด้านล่าง แล้วนำสัญญาณ SIGNAL ที่ได้ป้อนเข้า ADC ขาอ่านล็อกอินพุต A1 ของบอร์ด Arduino

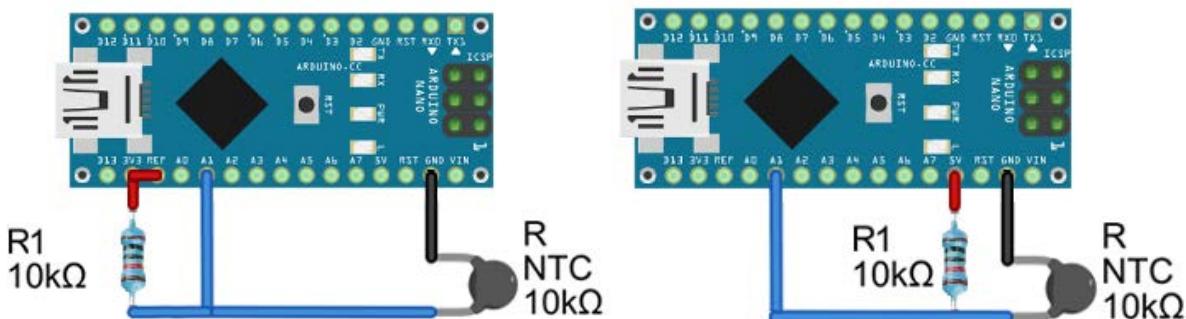


27. ให้นักศึกษาทำการแก้ไขโปรแกรมในข้อ 23 เพื่อทำเป็นเครื่องตรวจนับตระปลอมแบบอัตโนมัติ โดยเพิ่มวงจรใช้ Ultra Violet (UV) LED ที่มีช่วงความยาวคลื่น 390 นาโนเมตร ต่ออนุกรมกันตัวความด้านทาน 100 โอห์ม เมื่อคำนับตรมานเข้าใกล้ให้สั่งเปิด LED ที่เป็นแสง UV ให้ติดสว่าง

## Temperature Sensor

$$T \propto R \propto \frac{1}{V}$$

การทดลองการวัดอุณหภูมิ จะใช้เซ็นเซอร์สำหรับวัดอุณหภูมิเป็นตัวเทอร์มิสเตอร์ ซึ่งจะเป็นตัวด้านท่านที่มีค่าของความด้านท่านเปลี่ยนแปลงตามอุณหภูมิ การทดลองจะต้องวัดค่าความด้านท่านของเทอร์มิสเตอร์นี้และเปลี่ยนค่าความด้านท่านที่ได้ไปเป็นระดับแรงดันไฟฟ้า โดยต่อค่าแรงดันไฟฟ้าที่วัดได้ผ่านเข้าทางขาอะนาล็อกของ Arduino และใช้การคำนวณค่าอุณหภูมิโดยสมการ Steinhart-Hart ซึ่งอธิบายค่า thermistor resistance – temperature curve โดยวงจรที่จะใช้ทดลองเป็นวงจรแบ่งแรงดันไฟฟ้าแสดงได้ดังรูป



การวัดแรงดันไฟฟ้า เราจะต้องเชื่อมต่อตัวเทอร์มิสเตอร์เข้ากับตัวด้านท่าน R1 ขนาด  $10\text{k}\Omega 1\%$  และถูกต่อเข้ากับแรงดันไฟฟ้า  $V_{cc}$  ของวงจร เพื่อทำวงจรแบ่งแรงดันไฟฟ้า ตัวความด้านท่านของเทอร์มิสเตอร์ R จะใช้ NTC Thermistor เบอร์ MF52-3435 มีความด้านท่าน  $10\text{k}\Omega$  ที่  $25^\circ\text{C}$  และมีความคลาดเคลื่อน  $1\%$  รายละเอียดต่างๆเพิ่มเติมให้เปิดดูได้จากเอกสาร Thermistor MF52\_3435 DataSheet

กำหนดให้แรงดันขาเอาท์พุทเป็น  $V_o$ , แหล่งจ่ายไฟเป็น  $V_{cc}$ , ความด้านท่านของตัวแปรเทอร์มิสเตอร์เป็น R และตัวความด้านท่านคงที่เป็น R1 จะได้แรงดันขาเอาท์พุทคือ

$$V_o = V_{cc} \frac{R}{R + R1}$$

แรงดันขาเอาท์พุทเชื่อมต่อเข้ากับขาแบบอนามัย A1 ของ Arduino Micro เป็นวงจร ADC ทำหน้าที่แปลงสัญญาอนามัยเป็นดิจิตอล (Analog to Digital Converter) ขนาด 10 บิต ซึ่งจะทำให้แรงดันไฟฟ้าถูกแปลงเป็นตัวเลขระหว่าง 0 ถึง 1023 ค่า ADC ที่วัดจาก Arduino Micro จะได้แรงดันขาเอาท์พุทคงนี้

$$V_o = V_{cc} \frac{A1}{1023}$$

โดยการแทนค่า  $v_o$  ทั้งสองสมการเข้าด้วยกันเป็น

$$V_{cc} \frac{R}{R + R1} = V_{cc} \frac{A1}{1023}$$

จะได้

$$\frac{R}{R + R1} = \frac{A1}{1023}, \quad \frac{R + R1}{R} = \frac{1023}{A1}$$

การวัดอุณหภูมิได้ตัวแปรความด้านท่านเทอร์มิสเตอร์ R คือ

$$R = R1 \frac{A1}{1023 - A1}$$

$$1 + \frac{R1}{R} = \frac{1023}{A1}$$

$$\frac{R1}{R} = \frac{1023}{A1} - \frac{A1}{A1} \quad \dots(1)$$

$$\frac{R1}{R} = \frac{1023 - A1}{A1}$$

$$R = R1 \left( \frac{A1}{1023 - A1} \right)$$

## 01236255 INTRODUCTION TO INTERNET OF THINGS

เพื่อให้การวัดความต้านทานของเทอร์มิสเตอร์มีเสถียรภาพมากขึ้น ป้องกันไม่ให้ค่าที่วัดได้เปลี่ยนแปลงไปตามแรงดันไฟฟ้าของแหล่งจ่ายไฟที่มาจาก USB ของคอมพิวเตอร์ ซึ่งใช้เป็นแหล่งจ่ายพลังงานกับบอร์ดและวงจรต่างๆ ดังนั้นอาจจะมีสัญญาณรบกวนได้ จึงอาจจะใช้การเชื่อมต่อ **Vcc กับขา Arduino 3V แทนขา 5V** เพราะมันจะผ่านมาจากการคุณแรงดันอิเล็กทรอนิกส์และความถูกต้องของอุณหภูมิจะขึ้นอยู่กับแรงดันไฟฟ้าที่ต่ำกว่า

การทำงจะต้องใช้อุปกรณ์ขึ้นส่วนอิเล็กทรอนิกส์แบบพาสซีฟทุกชิ้นที่มีค่าความคลาดเคลื่อนน้อยที่สุด เพราะจะมีความสัมพันธ์กับค่าที่อ่านออกมากความผิดพลาด ดังนั้นจึงให้เลือกใช้เทอร์มิสเตอร์  $10\text{ K}\Omega$  ที่มีความคลาดเคลื่อน  $1\%$  ซึ่งจะมีผลให้ค่าความต้านทานเกิดความผิดพลาดได้สูงสุด  $100\text{ m}\Omega$  ที่อุณหภูมิ  $25^\circ\text{C}$  องศาเซลเซียส โดยที่อุณหภูมิ  $25^\circ\text{C}$  องศาเซลเซียสแสดงถึงค่าความต้านทานที่มีความผิดพลาด  $1\%$  จะได้ความผิดพลาดของอุณหภูมิประมาณ  $0.2^\circ\text{C}$  องศาเซลเซียส การแปลงค่าความต้านทานไปเป็นการวัดอุณหภูมิของเทอร์มิสเตอร์มีความสัมพันธ์ที่ก่อนเขียนชี้แจงระหว่างความต้านทานและอุณหภูมิโดยที่ไปแล้วสามารถใช้ตารางการค้นหาความต้านทานต่อการเปลี่ยนแปลงอุณหภูมิได้ตาม datasheet ของอุปกรณ์ได้ แต่ในที่นี้จะใช้ **สมการ Steinhart-Hart (สมการพารามิเตอร์ B)** ซึ่งเป็นการคำนวณค่าของความต้านทานเทอร์มิสเตอร์ที่มีความสัมพันธ์กับอุณหภูมิ ได้

$$\frac{1}{T} = \frac{1}{T_0} + \frac{1}{B} * \ln \frac{R}{R_0} \quad \dots (2)$$

โดยที่ **R** เป็นความต้านทานของเทอร์มิสเตอร์ที่อุณหภูมิ **T** ในขณะนั้น

**R<sub>0</sub>** คือความต้านทานที่  $T_0 = 25^\circ\text{C}$

**B** เป็นค่าคงที่ขึ้นอยู่กับเทอร์มิสเตอร์ ค่า **B** มักอยู่ระหว่าง  $3000-4,000$

สมการขึ้นอยู่กับพารามิเตอร์ ( $R_0, T_0$  และ  $B$ ) ซึ่งหาได้จาก datasheet ของ thermistor ที่ใช้

28. การทดลองจะใช้ตัวเข็มเซอร์วัสดอุณหภูมิตัวอย่าง Thermistor ขนาด  $10\text{ K ohm}$  เบอร์ NTC-MF52-103/3435

คลาดเคลื่อน  $1\%$  ต่อระหว่างขา **A1 กับกราวด์** และใช้ตัวความต้านทาน **R1** ขนาด  $10\text{ K ohm}$   $1\%$  ต่อระหว่างขา **A1 กับไฟบวก 5V** เพื่อทำเครื่องวัดอุณหภูมิระบบดิจิตอล

29. ให้ทดลองป้อนโปรแกรมโดยกำหนดค่าต่างของอุปกรณ์ต่างๆ ซึ่งได้จาก datasheet ของ thermistor และอ่านค่าที่ได้จาก ADC ขา **A1** และแสดงผลออกไปทาง Serial Monitor ดังนี้

```
#define THERMISTOR A1           // thermistor pin
#define R0 10000                  // Ω resistance at 25 Celsius
#define B 3435                     // B: 3435 K the beta coefficient of the thermistor
#define R1 10000                  // 10KΩ the value of the series resistor

float T0 = 25;                   // °C reference temp.

void setup()
{
    T0 = T0 + 273.15;           // conversion from Celsius to kelvin
    Serial.begin(9600);
}

void loop()
{
    int samples;

    samples = analogRead(THERMISTOR); // read the input on analog pin 0
    Serial.print("Analog reading : "); // print out the value
    Serial.println(samples);
```

```

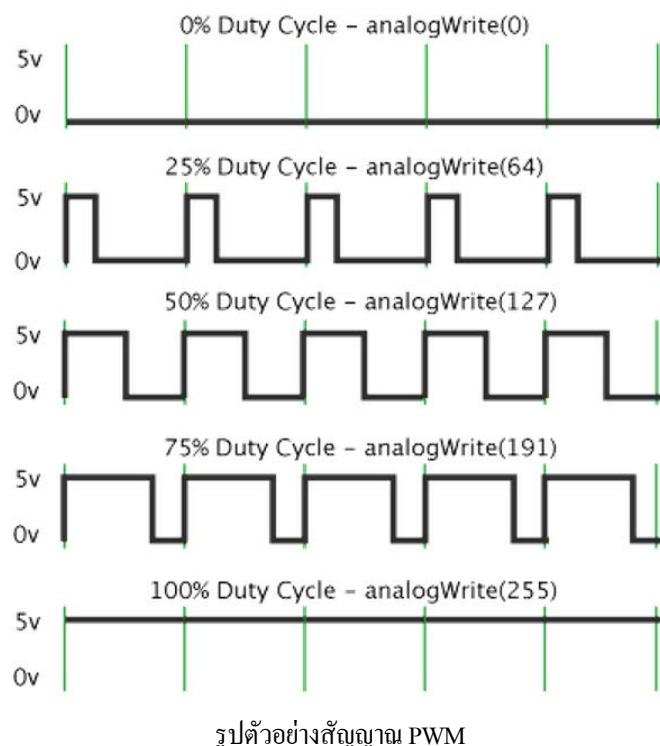
        delay(1000); // Wait for next sample
    }
}

```

30. ให้ทำการเพิ่มโปรแกรมการคำนวณเบลี่ยนค่าที่อ่านได้จากขา A1 ตัวแปร sample ไปเป็นค่าความด้านทานของตัว Thermistor โดยใช้สมการที่ 1 และกำหนดให้ตัวแปรความด้านทานเทอร์มิสเตอร์ R เป็นชนิด float แล้วให้พิมพ์ผลที่ได้ออก Serial.print(R);
31. ให้เพิ่มโปรแกรมการคำนวนค่าของความด้านทานเทอร์มิสเตอร์ที่มีความสัมพันธ์กับอุณหภูมิ โดยใช้สมการที่ 2 ซึ่งจะได้ค่าอุณหภูมิของเทอร์มิสเตอร์ออกมาเป็นตัวแปร T โดยค่าอุณหภูมิที่ได้นี้จะเป็น kelvin ให้แปลงค่าเป็น Celsius และพิมพ์ผลที่ได้ออกไปทาง Serial Monitor
32. จากการทดลองจะเห็นได้ว่าค่าที่อ่านออกมานาจจะกระโดดไปมาไม่นิ่ง ให้แก้ไขโปรแกรมเพิ่มการคำนวณหากค่าเฉลี่ยการวัดอุณหภูมิแสดงผลออกมากทุกครั้งวินาที

### Pulse Width Modulation Pin

Arduino สามารถส่งสัญญาณที่เป็น Pulse width modulation หรือเรียกย่อว่า PWM ซึ่งเป็นเทคนิคในการสร้างสัญญาณอนามัยอักด้วยค่าเฉลี่ยของสัญญาณดิจิตอล ออกแบบผ่านทางขา PWM ได้ ในบอร์ด Arduino Nano มีขา PWM ให้ใช้งานทั้งหมด 6 ขา แต่ละขาจะเป็นขนาด 8 bit โดยผู้ใช้สามารถสร้างความถี่ที่เป็นสัญญาณดิจิตอลรูปคลื่นสี่เหลี่ยม (square wave) พร้อมกับควบคุม Percent of Duty Cycle ได้ด้วยคำสั่ง analogWrite การกำหนดเพื่อปรับค่าความถี่ไฟเกิด จะเป็นการควบคุมความเวลาของสัญญาณที่เป็นalogic 1 เทียบกับความเวลาที่เป็นalogic 0 ซึ่งจะทำให้ค่าแรงดันเฉลี่ยของสัญญาณที่จำลองเป็นค่าอนามัยอักต่างกันออกไป โดยค่าของความถี่ไฟเกิดจะเรียกเป็นเบอร์เซ็นต์ ดังข้างของคำสั่งตามรูป



การทดลองจะเขียนโปรแกรมควบคุมไฟ LED ที่อยู่บนบอร์ดไมโครคอนโทรลเลอร์ หรือ LED ที่เรานำมาต่อวงจรเพิ่มเข้าไป สามารถปรับความสว่างได้ ด้วยคำสั่งที่ใช้ในการสร้างสัญญาณ PWM คือ analogWrite คำสั่งนี้ทำที่กำหนดจะสร้างสัญญาณคลื่นสี่เหลี่ยมด้วย Duty Cycle ค่าหนึ่งตามที่กำหนด และจะไม่เปลี่ยนค่าสัญญาณจนกว่าจะมีการเรียกคำสั่ง analogWrite ในครั้งต่อไป การใช้คำสั่ง analogWrite ไม่จำเป็นจะต้องมีการกำหนดขาด้วยคำสั่ง pinMode ก่อน รูปแบบของคำสั่งคือ analogWrite(pin, value) โดยที่ pin คือขาที่ต้องการให้สร้างสัญญาณ PWM และ value จะเป็นเบอร์เซ็นต์ของ duty cycle ที่เราต้องการ โดย 0 หมายถึง 0 เมอร์เซ็นต์ และ 255 หมายถึง 100 เมอร์เซ็นต์

33. ให้ต่อวงจรใช้ตัวค้านทาน 220 ohm และ LED เช็คที่ขา D3 โดยขาอีกด้านหนึ่งให้ต่อลงกราวด์ และป้อนโปรแกรมดังด้านล่างที่จะใช้คำสั่ง analogWrite ในการควบคุมปรับความสว่างของ LED

```

int led = 3; // LED connected to digital pin 3
int fade = 5; // how many points to fade the LED

void setup()
{
}

```

## 01236255 INTRODUCTION TO INTERNET OF THINGS

```
void loop()
{
    for(int brightness = 0; brightness <= 255; brightness +=fade)      // fade in from min to max
    {
        analogWrite(led, brightness);                                // sets the brightness
        delay(30);
    }
    for(int brightness = 255; brightness >= 0; brightness -=fade)      // fade out from max to min
    {
        analogWrite(led, brightness);                                // sets the brightness
        delay(30);
    }
}
```

34. ถ้าต้องการจะให้หลอดไฟ LED มีวิธีอะไรบ้าง

.....  
.....  
.....

35. ให้ต่อ LED เพิ่มเป็นจำนวน 6 หลอด โดยใช้ขา PWM แล้วให้เขียนโปรแกรมควบคุมให้หลอดไฟเหลือดีวั่งแล้วหรือลักษณะเหมือนฟันดาบทก หรือมีรูปแบบอื่นๆ ตามที่ต้องการมาหลายๆแบบ