บทที่ 3

การเขียนโปรแกรมบน Arduino : Analog

อาจารย์สกรณ์ บุษบง

สาขาวิทยาการคอมพิวเตอร์

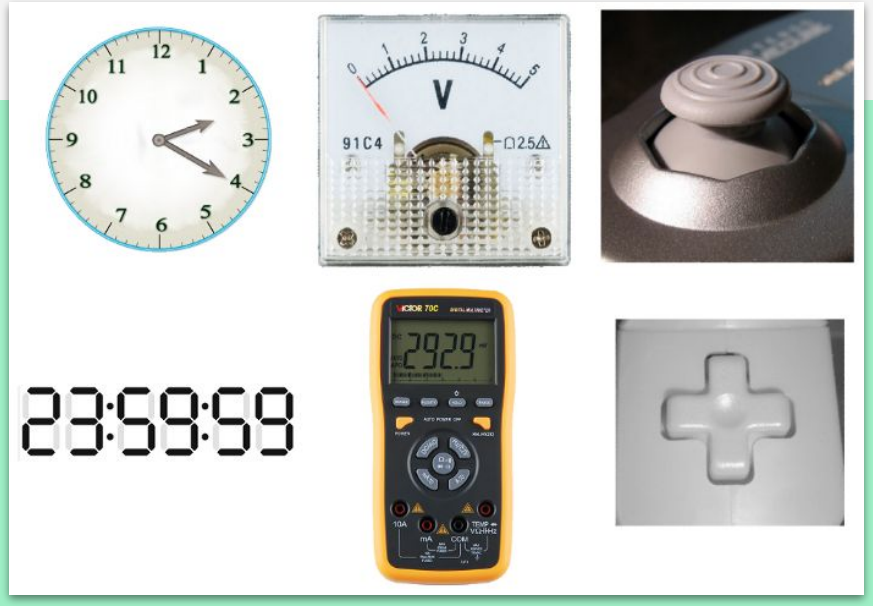
คณะวิทยาศาสตร์

มหาวิทยาลัยราชภัฏบุรีรัมย์

zagon.bb@bru.ac.th

ความแตกต่างของสัญญาณ Digital และ Analog

- ปัจจุบันรอบตัวเรามีรูปแบบข้อมูลที่เป็น Analog อยู่รอบตัว เช่น โทนเสียง, โทรสี เป็นต้น
- อุปกรณ์ต่าง ๆ ก็สามารถอยู่ในรูปแบบ Analog หรือ Digital ก็ได้



สัญญาณ คืออะไร

- สัญญาณในทางอิเล็กทรอนิกส์ คือ “ปริมาณ” ที่เปลี่ยนแปลงไปตามเวลา
- ในด้านวิศวกรรมไฟฟ้าปริมาณที่เปลี่ยนแปลงไปตามเวลาโดยปกติจะเป็นแรงดันไฟฟ้า ดังนั้น เมื่อพวกเราพูดถึงสัญญาณก็เพียงคิดว่ามันเป็นแรงดันไฟฟ้าที่เปลี่ยนแปลงตลอดเวลา
- สัญญาณจะถูกส่งระหว่างอุปกรณ์เพื่อสื่อสารกัน อาจจะเป็นภาพ เสียง วีดีโอ หรือข้อมูลที่ถูกเข้ารหัสก็ได้
- สัญญาณถูกส่งผ่าน**สาย** หรือ **ไร้สาย**ก็ได้

สัญญาณ Analog

“ทุกปรากฏการณ์ในโลกธรรมชาติเป็น analog”

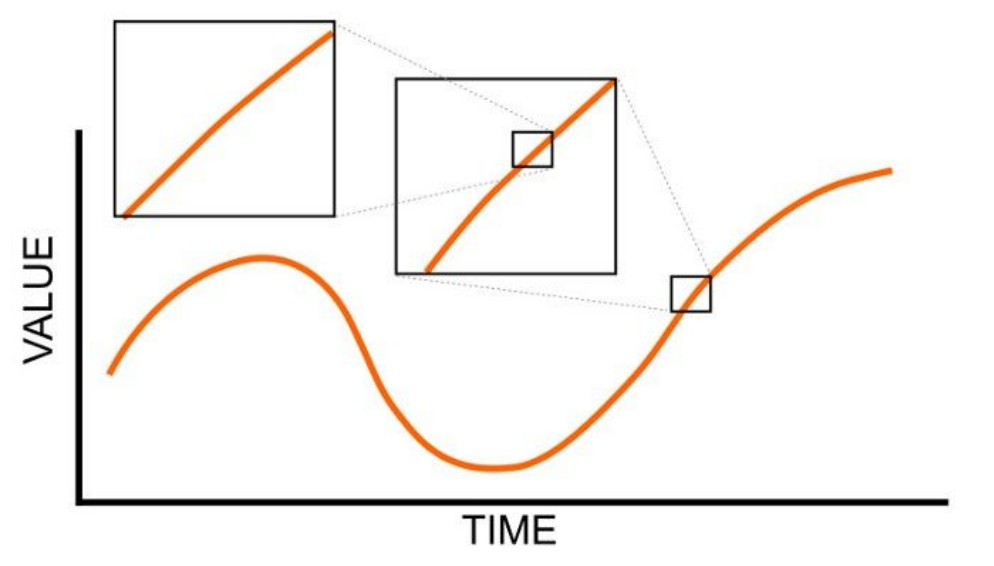
คำจำกัดความของ analog **“... ปริมาณทางกายภาพอย่างต่อเนื่องของตัวแปร ...”**

ปริมาณทางกายภาพเป็นได้ทุกอย่าง ทั้งอุณภูมิของร่างกาย, ความดันในล้อรถ, ระดับน้ำในทะเลสาบ, แรงดันไฟฟ้าภายในวงจรไฟฟ้า, ความเร็วของเครื่องบิน ฯลฯ

เนื่องจากสัญญาณมีความแตกต่างกันไปตามช่วงเวลา ซึ่งเป็นประโยชน์ในการวาดกราฟ

สัญญาณ Analog

เพื่อให้เข้าใจคำว่า **“อย่างต่อเนื่องของตัวแปร”** ให้เราพิจารณาภาพที่แสดงกราฟของค่าแรงดันไฟฟ้าเทียบกับเวลา กราฟเวลากับแรงดันของสัญญาณ analog ควรจะราบรื่นและต่อเนื่อง



แม้ว่าสัญญาณเหล่านี้อาจถูกจำกัดไว้ แต่ก็ยังคงมีค่าที่เป็นไปได้ไม่มีขีดจำกัดภายในช่วงนั้น เช่นแรงดันไฟฟ้าถูกจำกัดไว้ที่ -5 v ถึง +5 v แต่เมื่อเราเพิ่มความละเอียดเราจะพบค่าได้อย่างไม่มีสิ้นสุดที่เป็นไปได้(เช่น 3.4 v, 4.24 v, 4.674 v และไม่มีที่สิ้นสุด)

สัญญาณ Digital

- คอมพิวเตอร์มีทั้ง 1, 0 ทั้ง yes, no ทั้ง on, off ทั้ง low, high เป็นต้น สิ่งเหล่านี้เป็นจริงสำหรับคอมพิวเตอร์แบบ digital และหน่วยพื้นฐานของข้อมูลจัดเก็บในแบบ digital จัดเก็บว่า 1 หรือ 0 เรียกข้อมูลนี้ว่า bit
- 1 bit เป็นได้แค่ 0 หรือ 1 เท่านั้น

1 BIT

PATTERN	NUMBER
0	0
1	1

สัญญาณ Digital

เปรียบเทียบระหว่าง 1 bit และ 2 bit

1 BIT

PATTERN	NUMBER
0	0
1	1

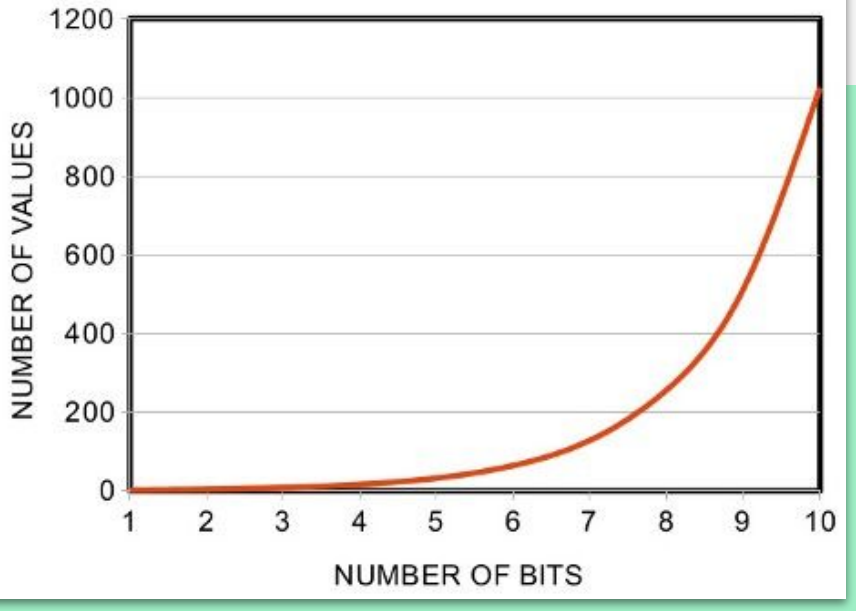
2 BITS

PATTERN	NUMBER
00	0
01	1
10	2
11	3

สัญญาณ Digital

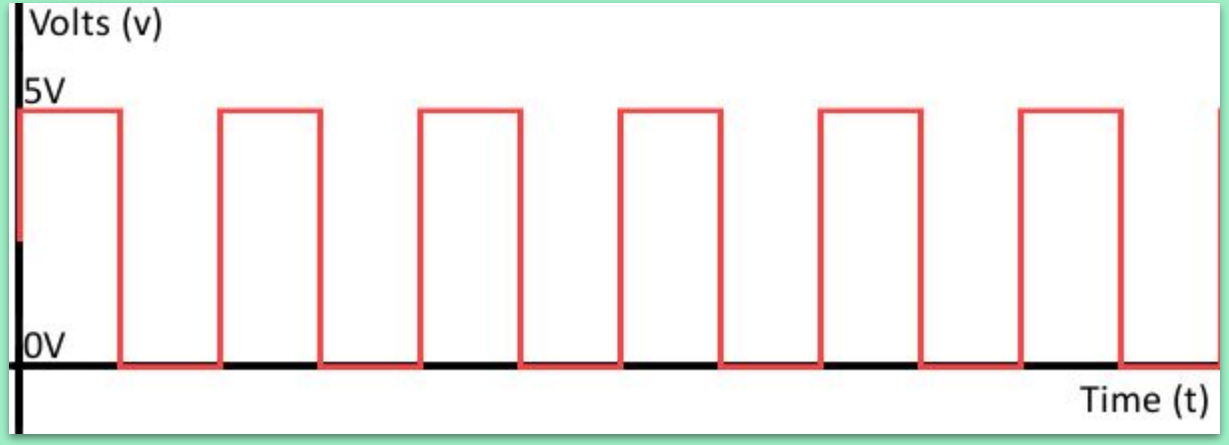
- ยิ่งเราเพิ่มจำนวน bit เข้าไปเท่าไหร่ เราก็จะใช้แสดงค่าได้มากขึ้นเท่านั้น
- การเพิ่มจำนวน bit จะเพิ่มจำนวนของปริมาณข้อมูลเข้าไปอีกเท่าตัวหนึ่ง ทำให้รูปแบบที่เพิ่มขึ้นไม่ใช่เชิงเส้น ดังรูป

NUMBER OF BITS	NUMBER OF VALUES
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512
10	1024



สัญญาณ Digital

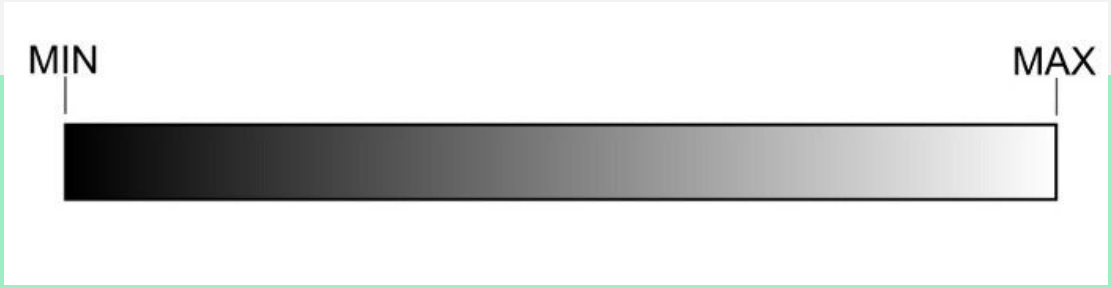
- สัญญาณ digital จะมีจำนวนค่าที่เป็นไปได้จำกัด จำนวนที่เป็นได้อาจเป็นจำนวนมากแต่ไม่ใช่อนันต์เหมือนอย่างสัญญาณ analog
- โดยทั่วไปแล้วสัญญาณจะถูกแทนด้วยแรงดันไฟฟ้า 0 v และ 5 v หากว่าเราสร้างกราฟแรงดันเทียบกับเวลาจะได้กราฟที่มีลักษณะเหมือนคลื่นสี่เหลี่ยม



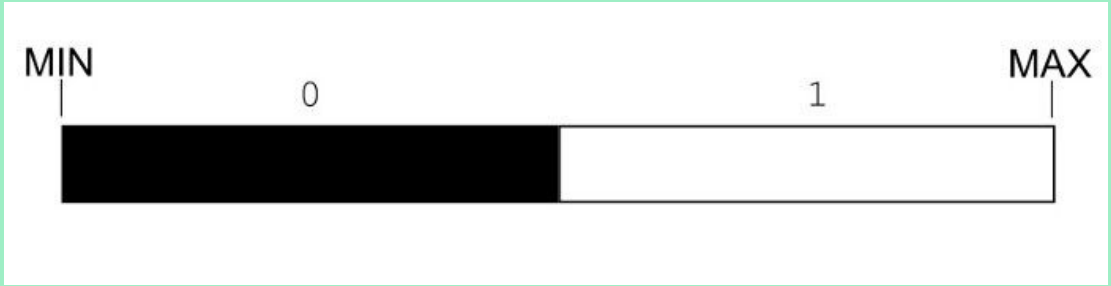
สัญญาณ Analog สู่สัญญาณ Digital

- ในเมื่อเป็นแบบนี้เราจะทำยังไงให้เราสามารถบันทึกค่า analog ที่ดีเหล่านี้ ลงไปในเครื่องคอมพิวเตอร์ของเราได้ ?
  - เพียงแค่เราใช้ตัวแปลง analog เป็น digital (ADC)

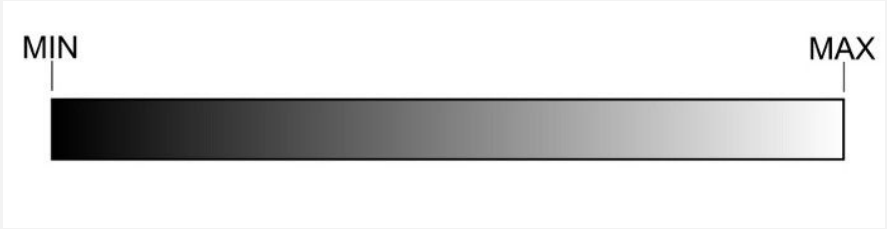
Analog



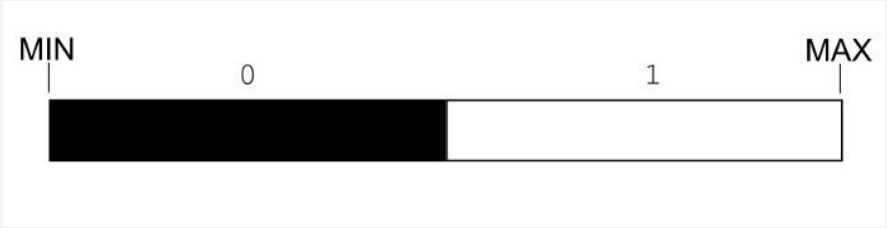
Digital 1 bit



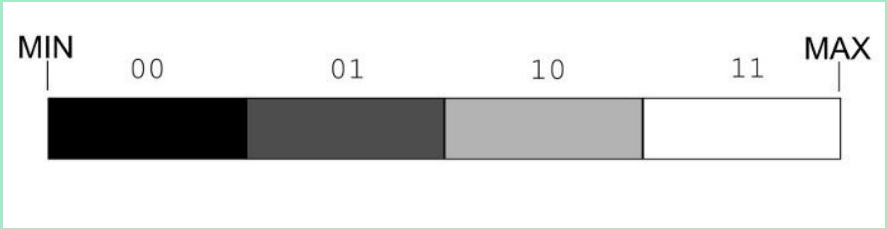
Analog



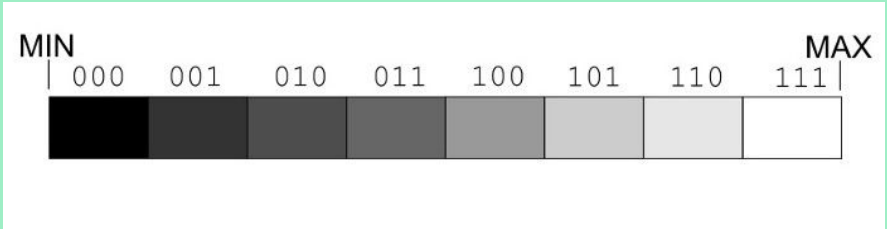
Digital 1 bit



Digital 2 bit

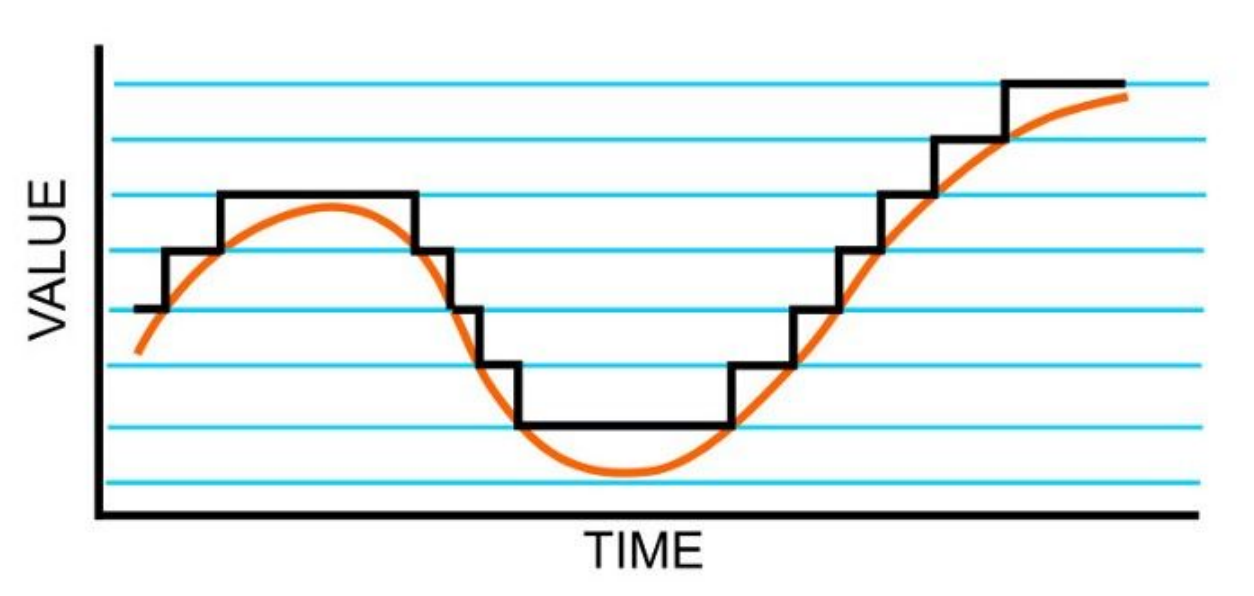


Digital 3 bit



สัญญาณ Analog สู่สัญญาณ Digital

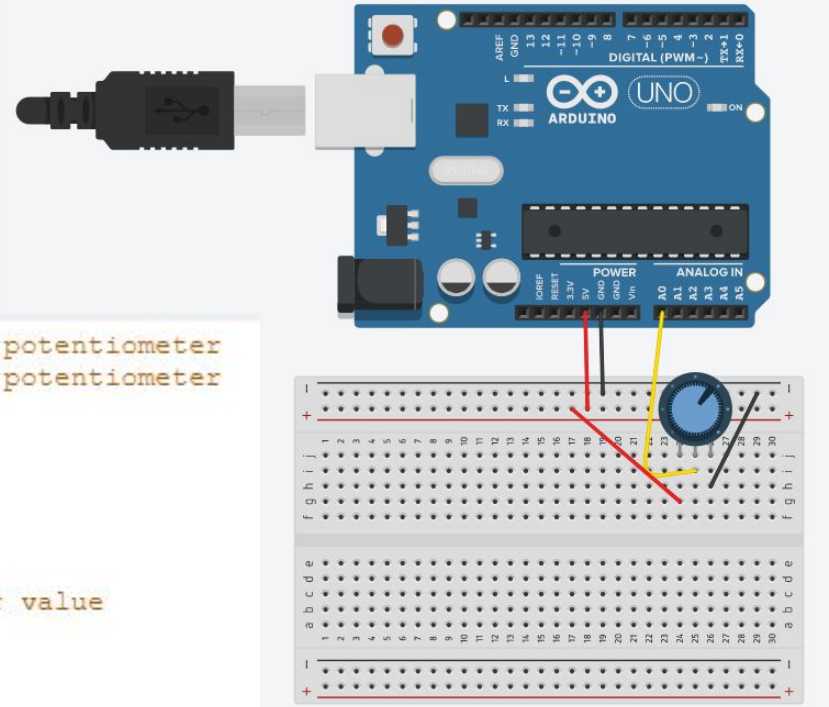
ถ้าเราใช้ ADC 3 bit มาแปลงสัญญาณจากภาพที่ 2 เราก็จะได้สัญญาณ analog (สีส้ม) ที่กลายเป็นสัญญาณ digital (สีดำ) ตามรูปแบบข้างล่าง



การอ่านค่า Analog

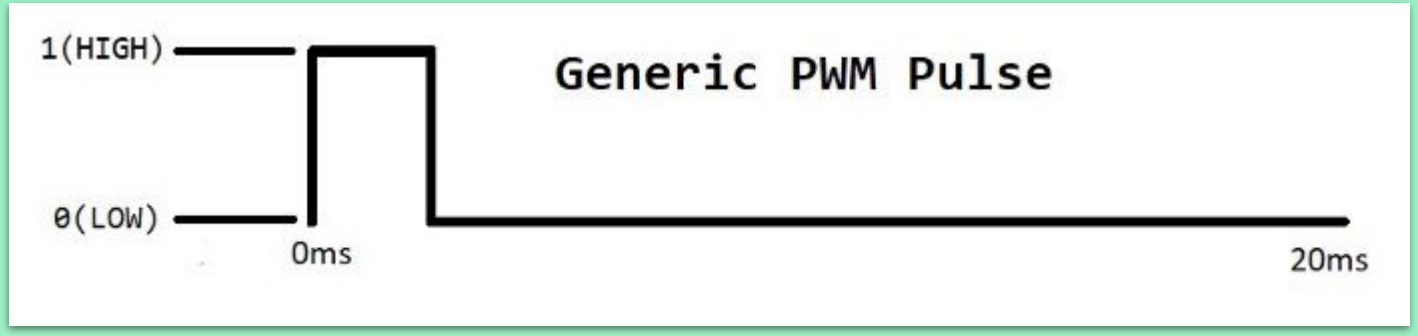
```
int potPin = A0; // Analog input pin connected to the potentiometer
int potValue = 0; // Value that will be read from the potentiometer
// Runs once at beginning of the program
void setup() {
  Serial.begin(9600);
}

void loop() {
  potValue = analogRead(potPin); // Read potentiometer value
  Serial.println(potValue);
  delay(10); // Wait for 10 ms
}
```



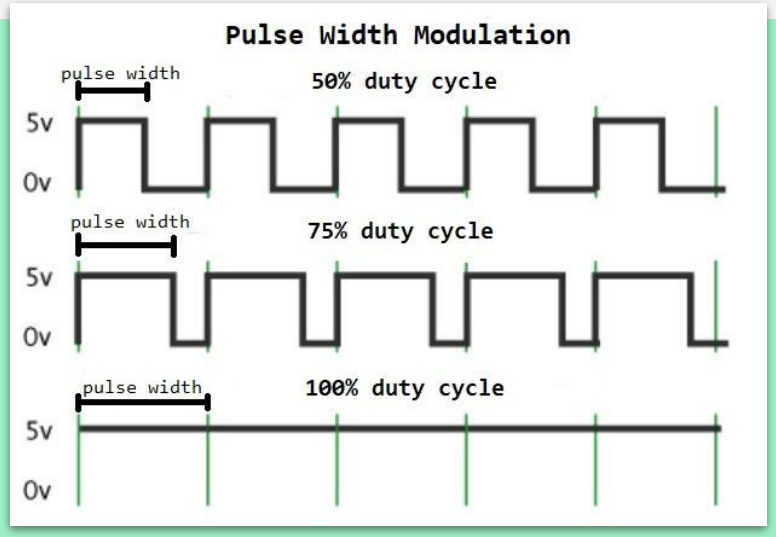
Pulse Width Modulation (PWM)

- Pulse Width Modulation(PWM) เป็นเทคนิคที่ทำให้เราสามารถอ่าน/เขียนข้อมูลแบบ analog ด้วยสัญญาณ digital ได้
- โดยจะสร้างสัญญาณคลื่นสี่เหลี่ยมออกมา สัญญาณที่สร้างออกมาจะสลับกันระหว่าง เปิด (HIGH) กับ ปิด(LOW) รูปแบบสัญญาณเปิด-ปิดนี้สามารถจำลองเป็นแรงดันไฟฟ้าระหว่าง เปิด (5 Volts) กับ ปิด (0 Volts)

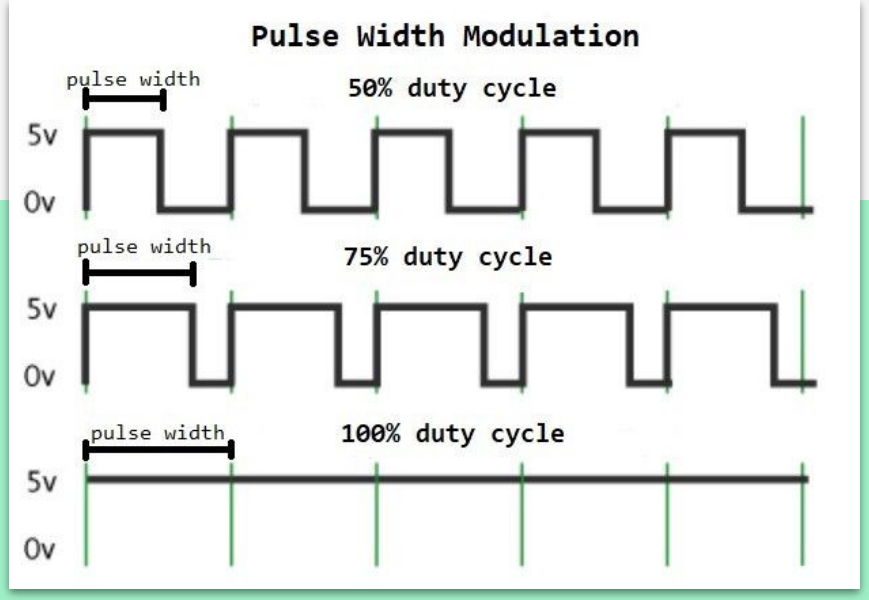


Pulse Width Modulation (PWM)

- ช่วงของเวลาที่สัญญาณเป็นเปิด เมื่อเทียบกับช่วงเวลาที่สัญญาณออกมาทั้งหมด เราจะเรียกช่วงเวลาตรงนี้ว่า ความกว้างของสัญญาณ (Pulse width)
- เพื่อให้ได้ข้อมูลแบบ analog ที่ต่างกัน เราจะใช้ความกว้างของสัญญาณที่แตกต่างกันในการบ่งบอกค่าของข้อมูล analog



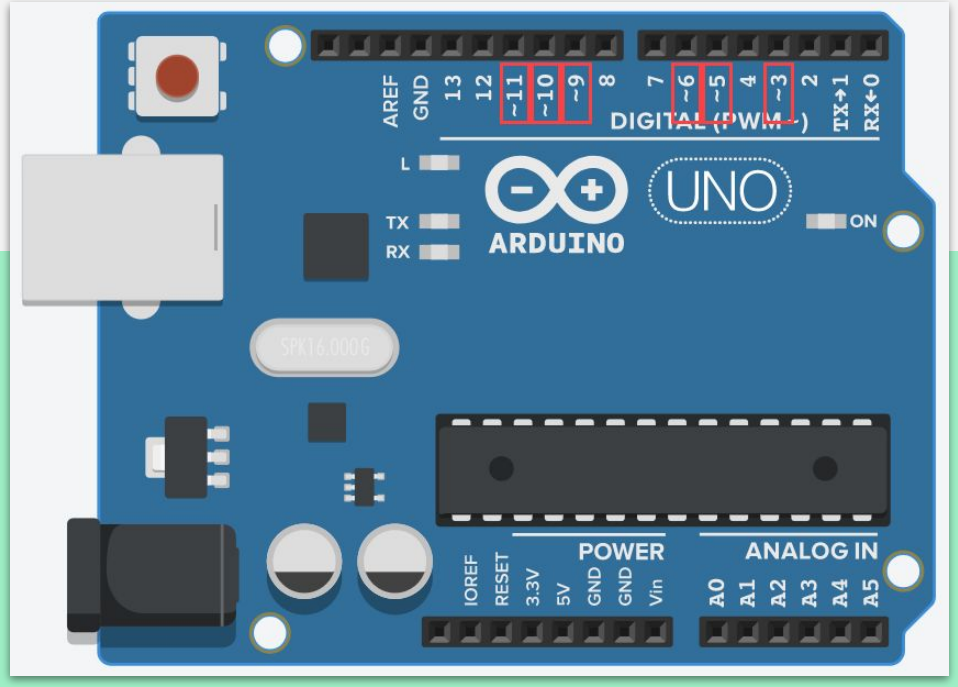
Pulse Width Modulation (PWM)



ความกว้างของเส้นสีเขียวจะแสดงถึงช่วงเวลาปกติ ระยะเวลาหรือช่วงเวลานี้เรียกว่า **period** เป็นค่าผกผันของความถี่อุปกรณ์ หากความถี่ของอุปกรณ์มีค่าประมาณ 50 Hz (ความถี่ขึ้นอยู่กับอุปกรณ์) ความกว้างของเส้นสีเขียวจะเป็นช่วงเวลา 20 ms (หาได้จาก T = 1/f)

PWM on Arduino

GPIO บน Arduino ที่รองรับ PWM คือ ขาที่มีสัญลักษณ์ตัวหนอน (~) กำกับอยู่



PWM on Arduino

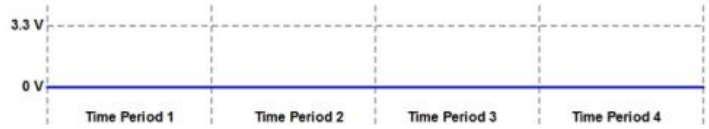
การเขียนโปรแกรมเพื่อส่งค่า PWM ออกไปที่ขา GPIO นั้นจะใช้คำคำสั่ง

analogWrite(pin,value)

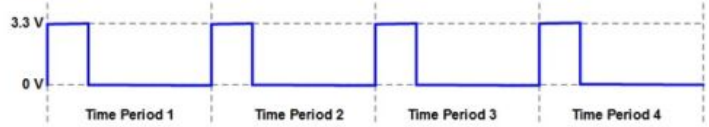
pin คือ หมายเลขขา GPIO ที่ต้องการส่งสัญญาน PWM ออกไป

value คือ ค่าที่ต้องการส่ง (0 - 1023)

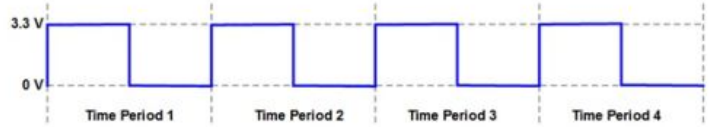
analogWrite(LedPin, 0);



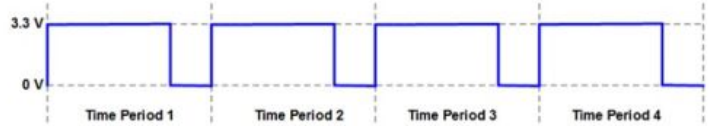
analogWrite(LedPin, 255);



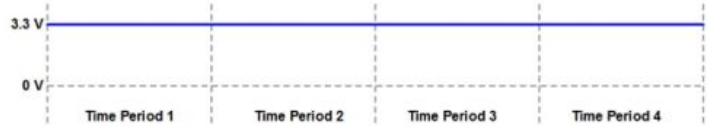
analogWrite(LedPin, 512);



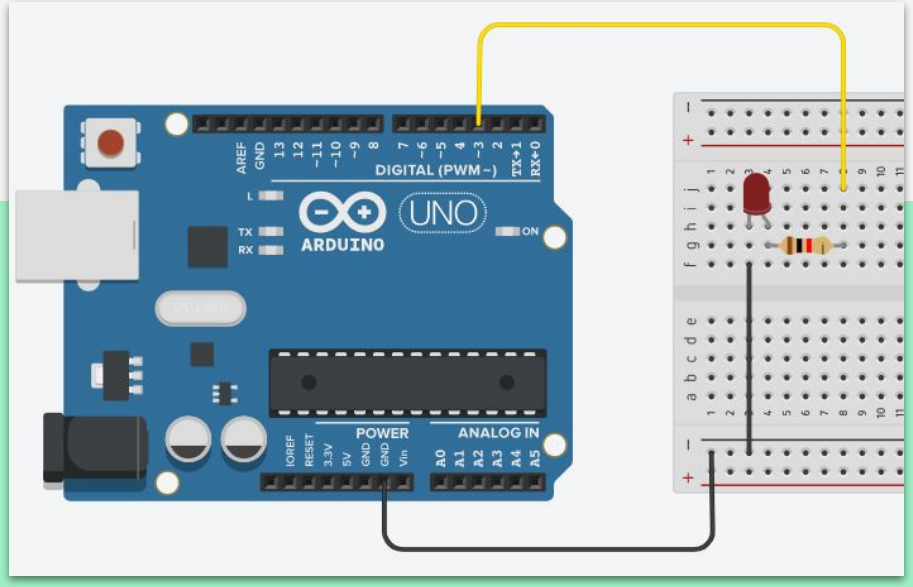
analogWrite(LedPin, 767);



analogWrite(LedPin, 1023);



PWM on Arduino
Lab : ไฟหรี่




```
1  int LedPin = 3;
2  void setup()
3  {
4    pinMode(LedPin, OUTPUT);
5  }
6
7  void loop()
8  {
9    analogWrite(LedPin, 0);
10   delay(1000);
11   analogWrite(LedPin, 255);
12   delay(1000);
13   analogWrite(LedPin, 512);
14   delay(1000);
15   analogWrite(LedPin, 767);
16   delay(1000);
17   analogWrite(LedPin, 1023);
18   delay(1000);
19 }
```

PWM on Arduino
Lab : ไฟหรี่

ปรับปรุง Code ให้สั้นลงโดยใช้ Array มาช่วย

```
int LedPin = 3;
int time[] = {0,255,512,767,1023};
void setup()
{
  pinMode(LedPin, OUTPUT);
}

void loop()
{
  for(int i = 0;i<=4;i++){
    analogWrite(LedPin, time[i]);
    delay(1000);
  }
}
```

PWM on Arduino
Lab : ไฟหรี่

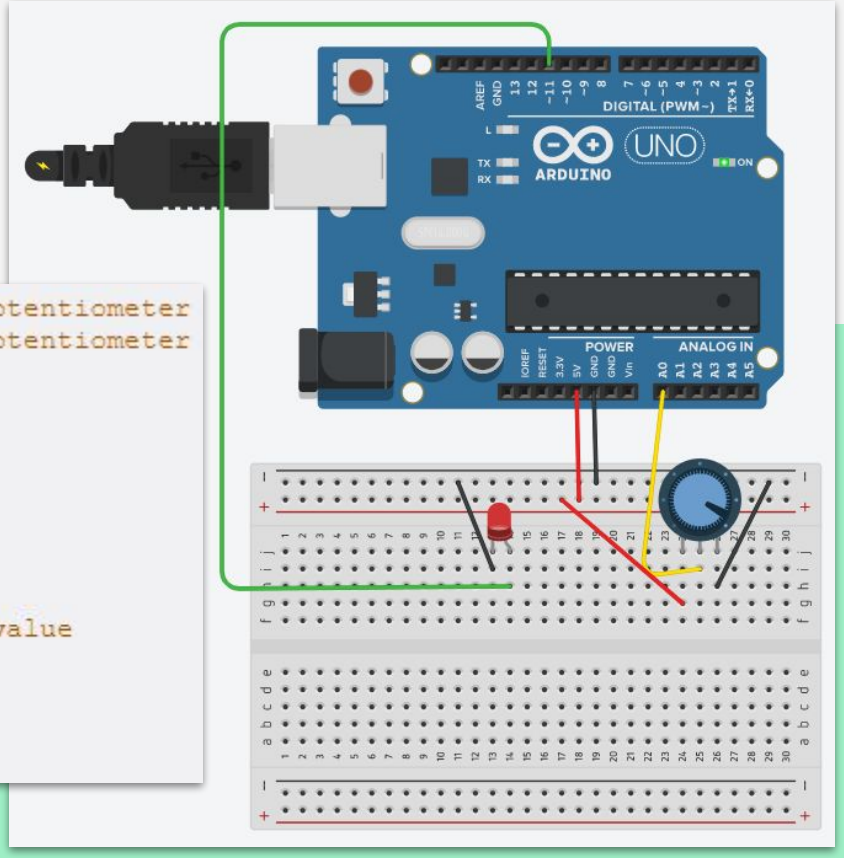
ประยุกต์ Loop มาช่วย

```
1  int LedPin = 3;
2  void setup()
3  {
4    pinMode(LedPin, OUTPUT);
5  }
6
7  void loop()
8  {
9    for(int i = 0;i<=1023;i++){
10     analogWrite(LedPin, i);
11     delay(10);
12   }
13 }
```

PWM on Arduino

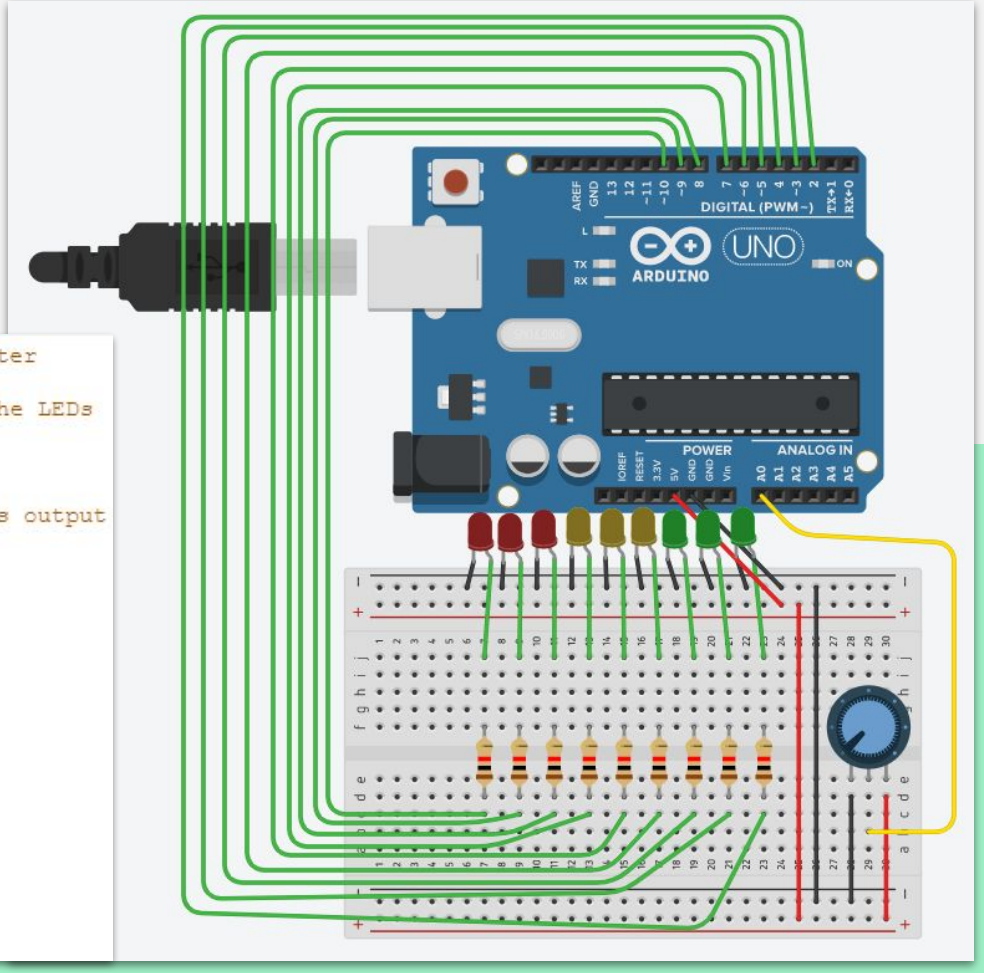
```ino
int potPin = A0; // Analog input pin connected to the potentiometer
int potValue = 0; // Value that will be read from the potentiometer
int led = 11;
// Runs once at beginning of the program
void setup() {
  Serial.begin(9600);
  pinMode(led, OUTPUT);
}

void loop() {
  potValue = analogRead(potPin); // Read potentiometer value
  Serial.println(potValue);
  analogWrite(led, potValue/4);
  delay(10); // Wait for 10 ms
}
```



BAR GRAPH

```
const int analogPin = A0; // Pin connected to the potentiometer
const int ledCount = 9; // Number of LEDs
int ledPins[] = {2,3,4,5,6,7,8,9,10}; // Pins connected to the LEDs
void setup() {
  Serial.begin(9600);
  for (int thisLed = 0; thisLed < ledCount; thisLed++) {
    pinMode(ledPins[thisLed], OUTPUT); // Set the LED pins as output
  }
}
// Start a loop
void loop() {
  int sensorReading = analogRead(analogPin); // Analog input
  Serial.println(sensorReading);
  int ledLevel = map(sensorReading, 0, 1023, 0, ledCount);
  for (int thisLed = 0; thisLed < ledCount; thisLed++) {
    if (thisLed < ledLevel) { // Turn on LEDs in sequence
      digitalWrite(ledPins[thisLed], HIGH);
    }
    else { // Turn off LEDs in sequence
      digitalWrite(ledPins[thisLed], LOW);
    }
  }
}
```



แบบฝึกหัด

ให้นักศึกษาอธิบายการทำงานของ Lab **BAR GRAPH**

จบบทที่ 3

Ref:

https://medium.com/icreativesystems/analog-vs-digital-c932477b0b63
https://medium.com/icreativesystems/to-know-pwm-373efa1322ce