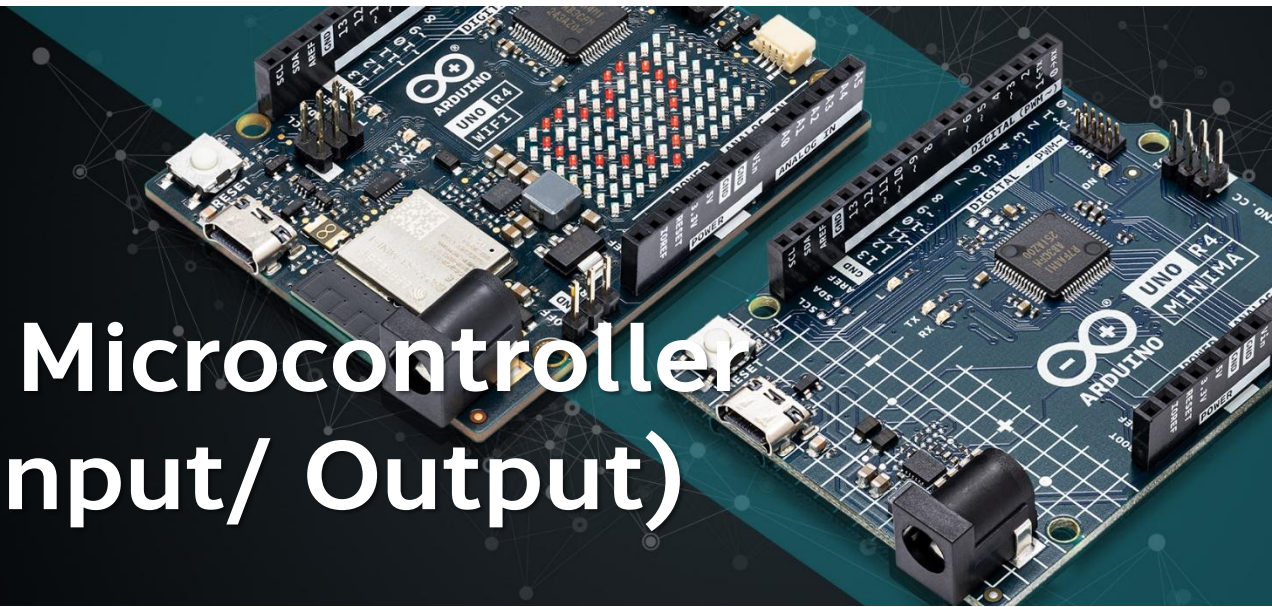




Physical Computing



ครั้งที่ 14-2. Microcontroller
(Analog Input/ Output)



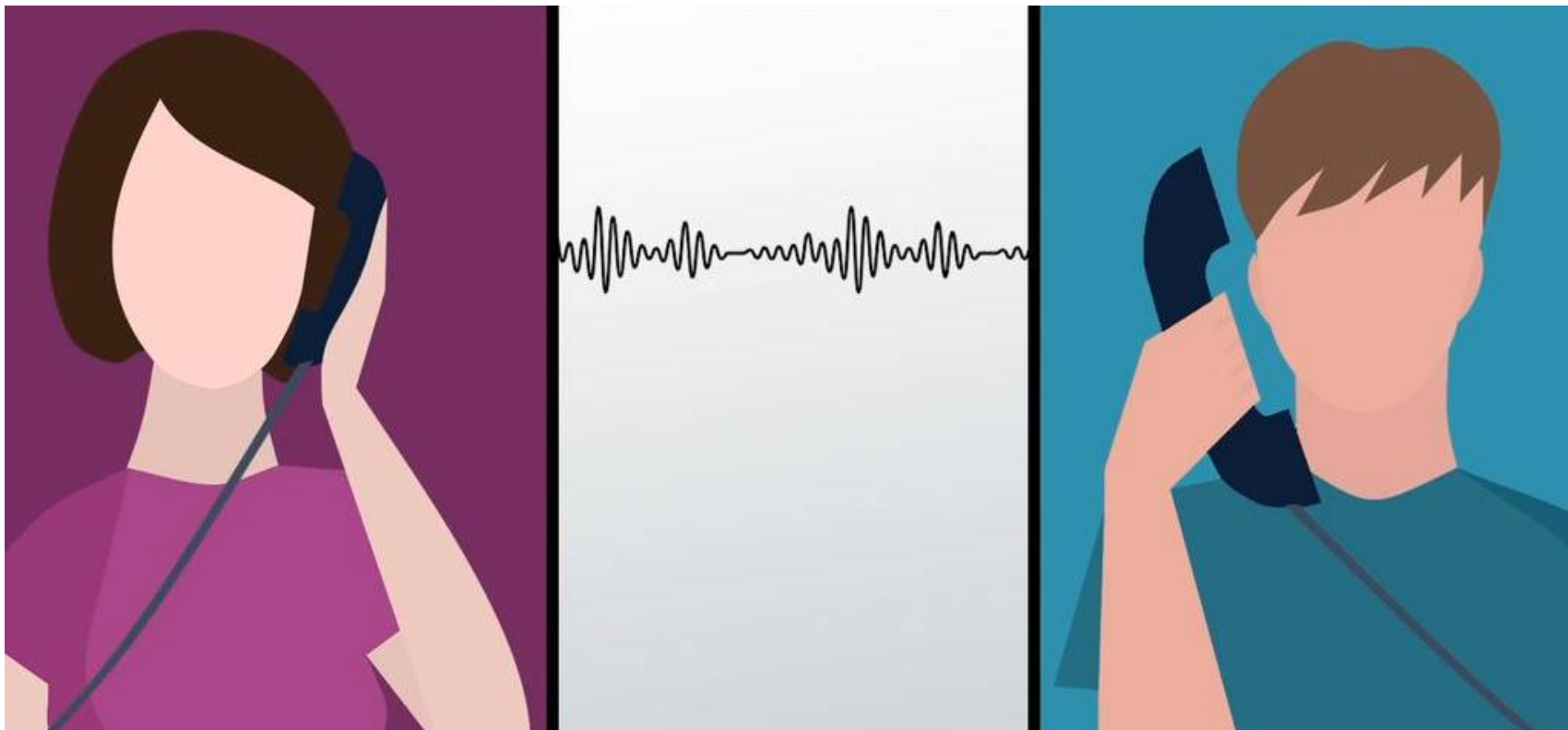
1. Analog Signal

Analog เป็นภาษากreek มาจากคำว่า analogous แปลว่า เหมือนกัน คล้ายคลึงกัน อุปมาเหมือน

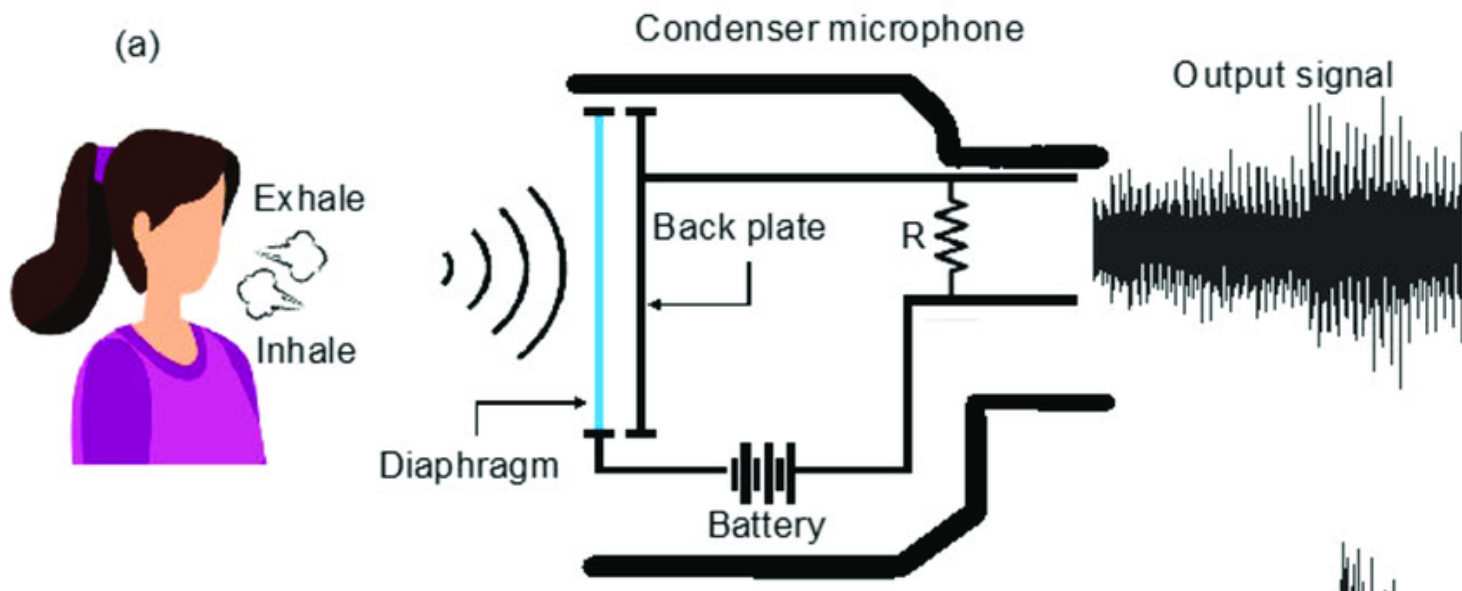
สัญญาณ Analog คือ สัญญาณที่คงรูปเดิมเหมือนกับต้นฉบับ จะมีลักษณะรูปคลื่นเป็นสัญญาณที่ต่อเนื่อง

จากลักษณะทางกายภาพของสัญญาณ Analog แปลงมาเป็น สัญญาณทางไฟฟ้าเพื่อแสดงให้เห็นจริง (วัดได้) จะใช้อุปกรณ์ที่ เรียกว่า **Sensor**

ตัวอย่างสัญญาณ Analog



ตัวอย่างสัญญาณ Analog



ตัวอย่างสัญญาณ Analog

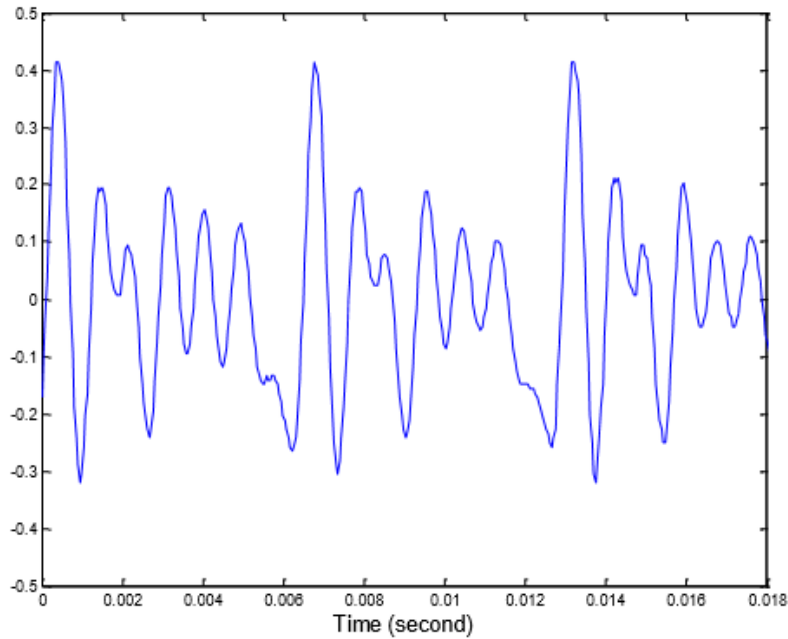
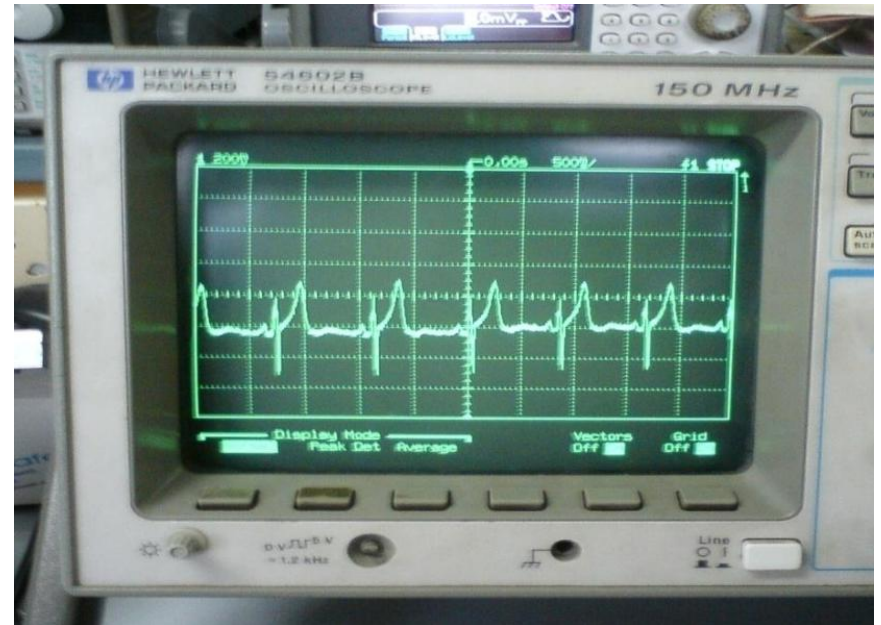


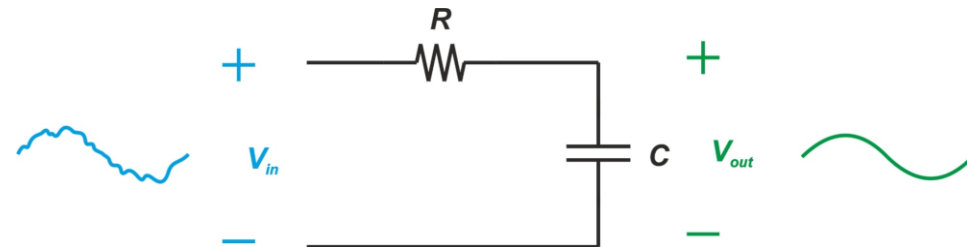
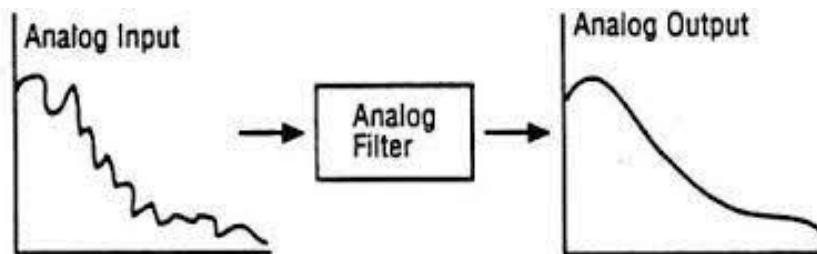
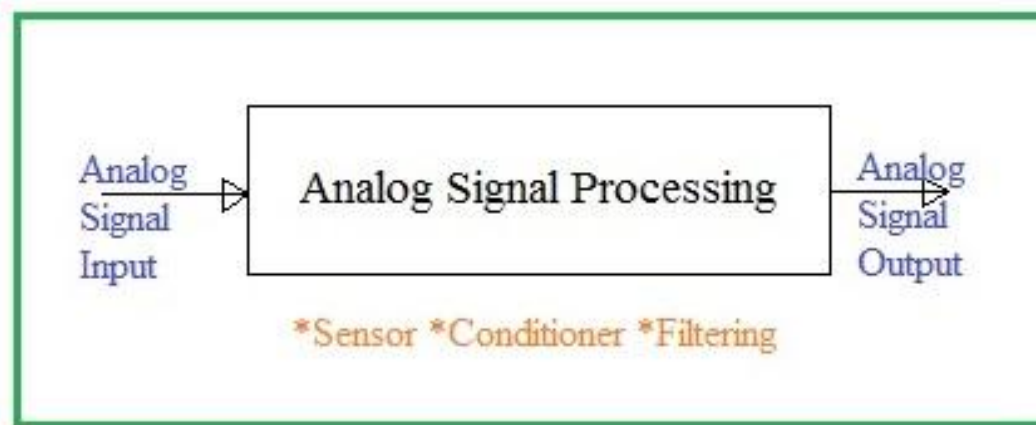
Figure 2. Example of speech signal



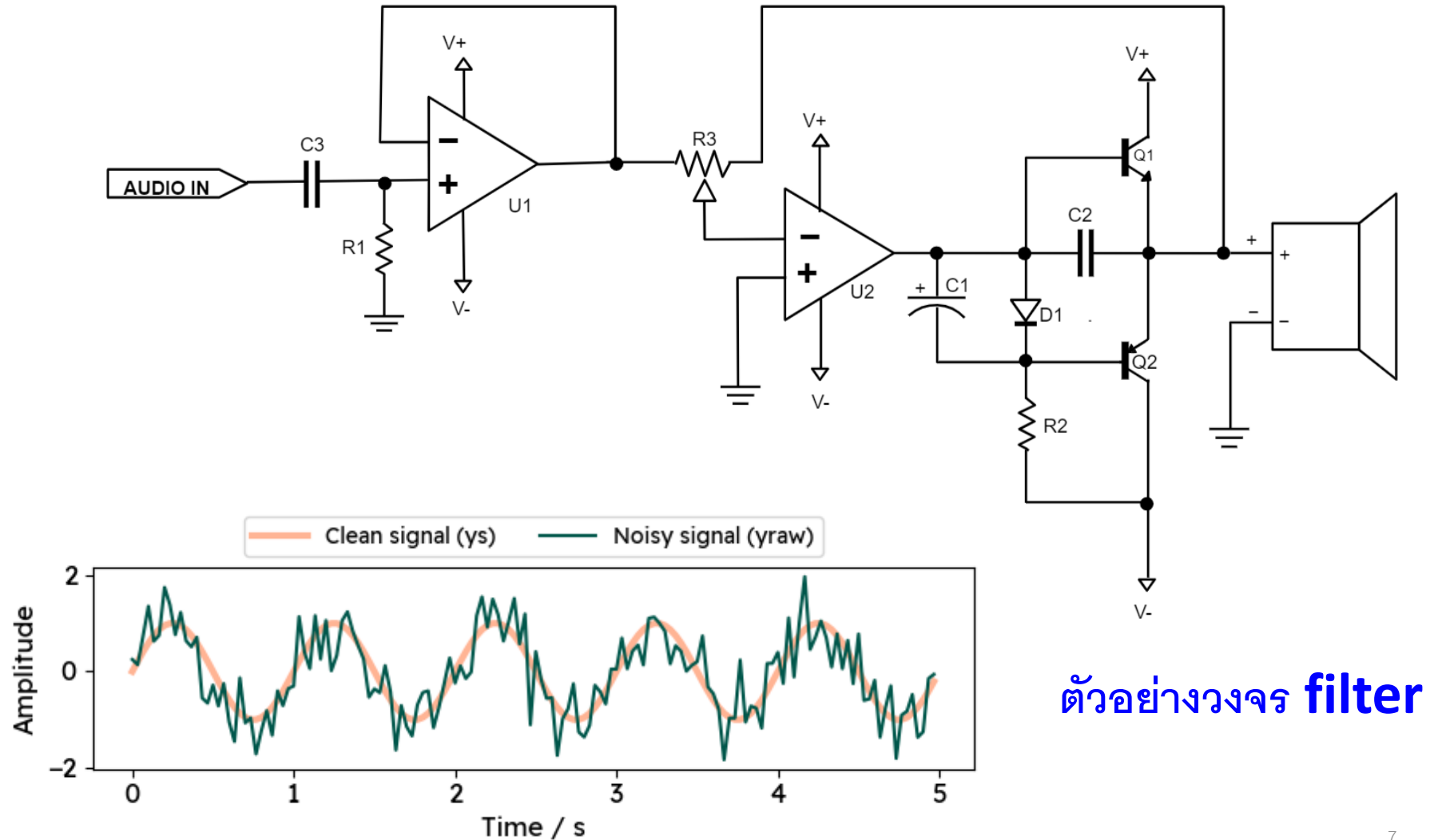
Example of ECG Signal

2. Analog Signal Processing

การประมวลผลสัญญาณ (signal processing) เช่น การขยายสัญญาณ (amplifier)
การกำจัดสัญญาณรบกวน (filter) การผสมสัญญาณ (Mix)

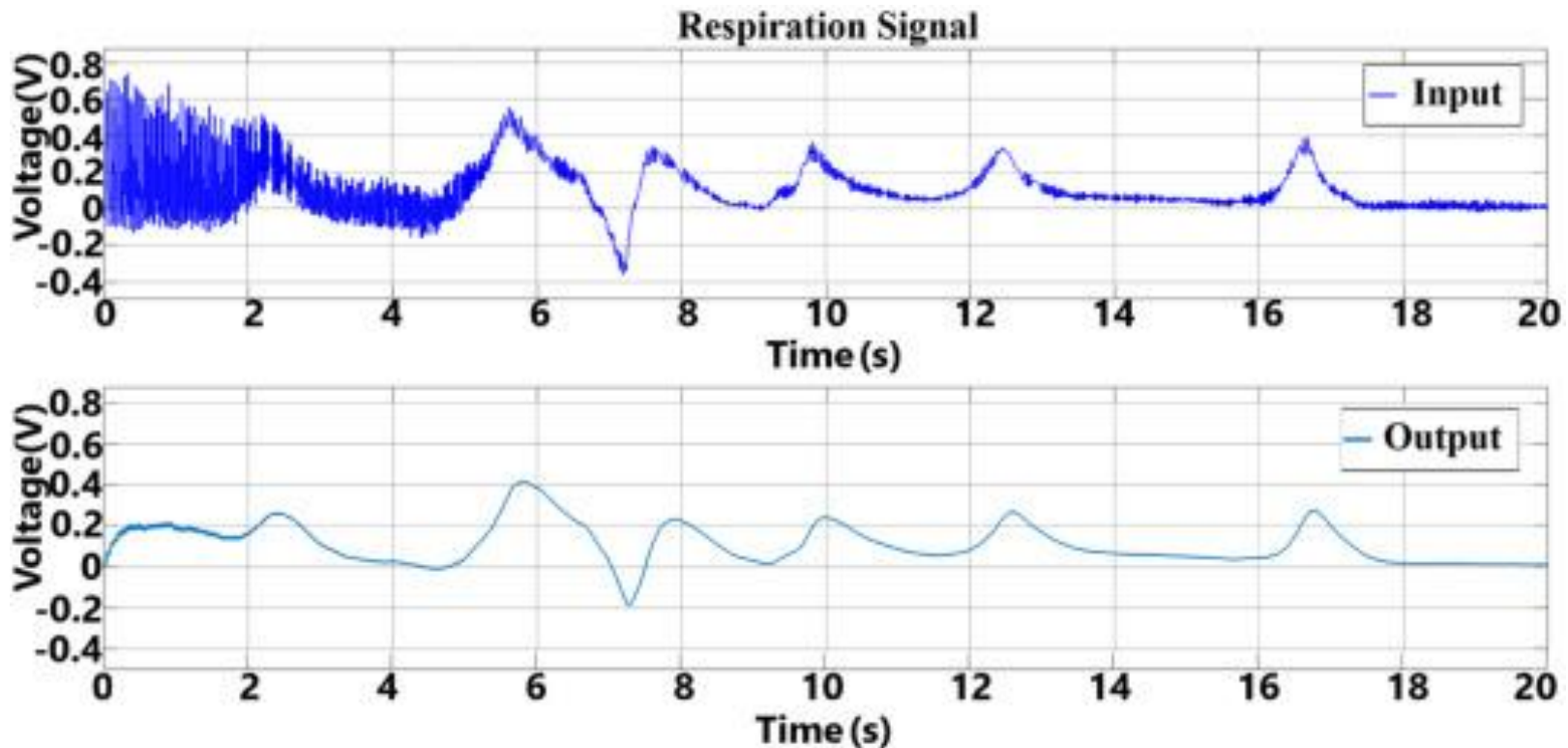


ตัวอย่าง Analog Signal Processing circuit



ตัวอย่างวงจร **filter**

ตัวอย่าง Analog Signal Processing

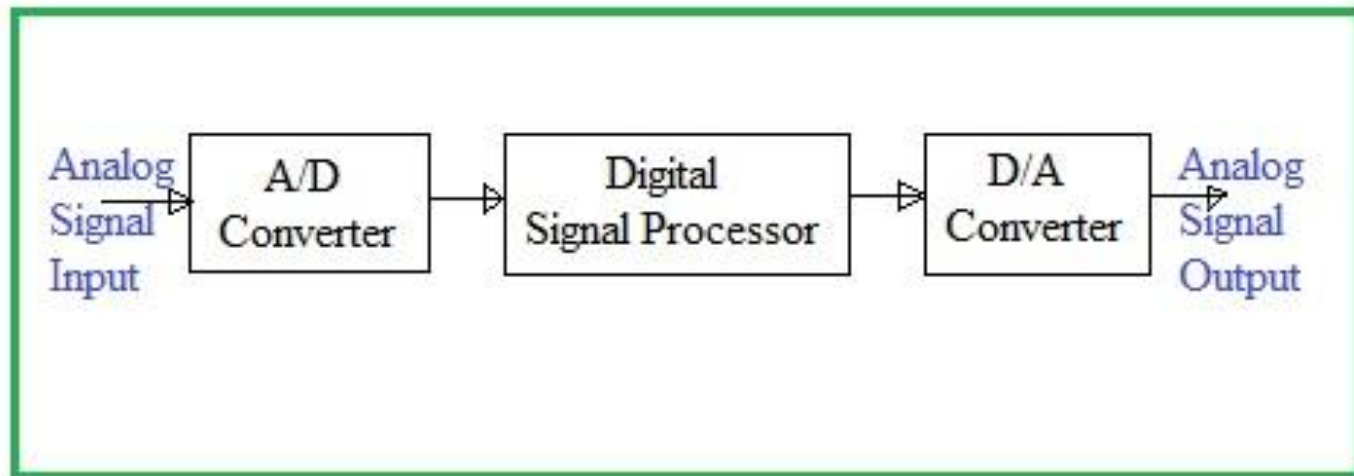


ตัวอย่าง การ **filter** สัญญาณ

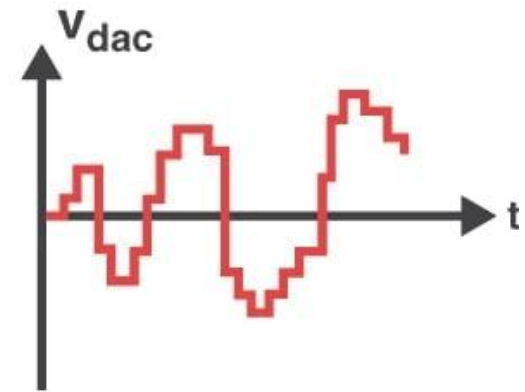
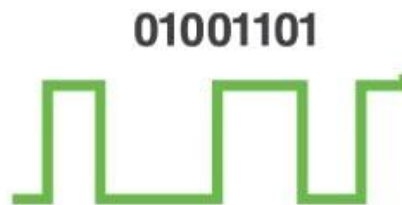
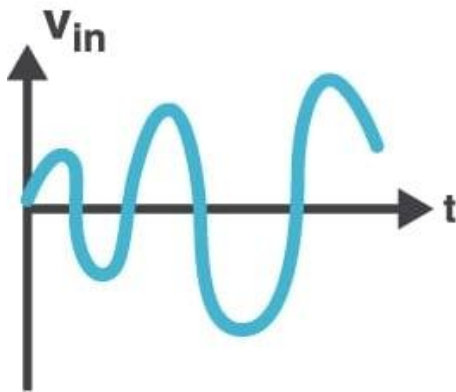
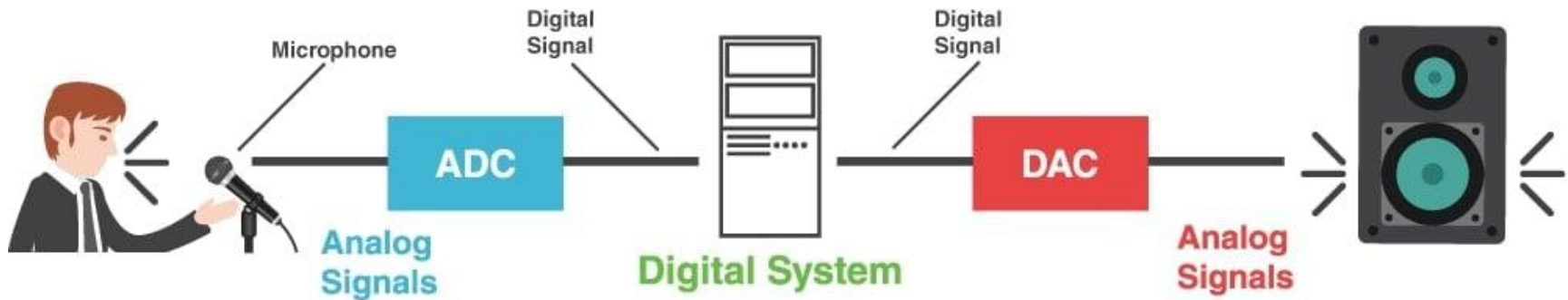
3. Digital Signal Processing

ในอดีต การประมวลสัญญาณ นิยมต่อวงจร ซึ่งมีข้อจำกัดคือ วงจรมีขนาดใหญ่ เสียเวลาในการต่อ การผลิต และต่อมาเทคโนโลยี **chip** มีการพัฒนามากขึ้น สามารถทำงานได้ไวขึ้น จึงมีแนวคิด นำสัญญาณมาแปลงเป็นสัญญาณดิจิทัล แล้วทำสัญญาณดิจิทัลนั้น มาประมวลด้วยการเขียนโปรแกรม จึงเป็นที่มาของ **Digital Signal Processing**

มี Block Diagram หลักๆ ดังรูป



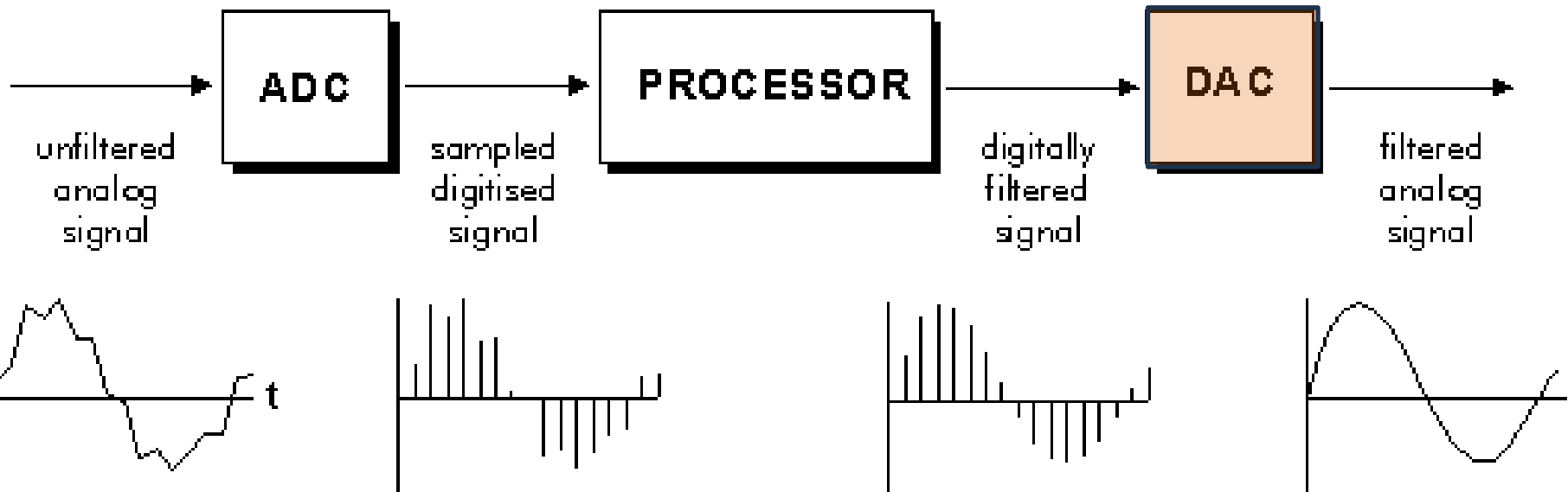
Digital Signal Processing



Digital Signal Processing

Analog to Digital Converter

Digital to Analog Converter



DSP

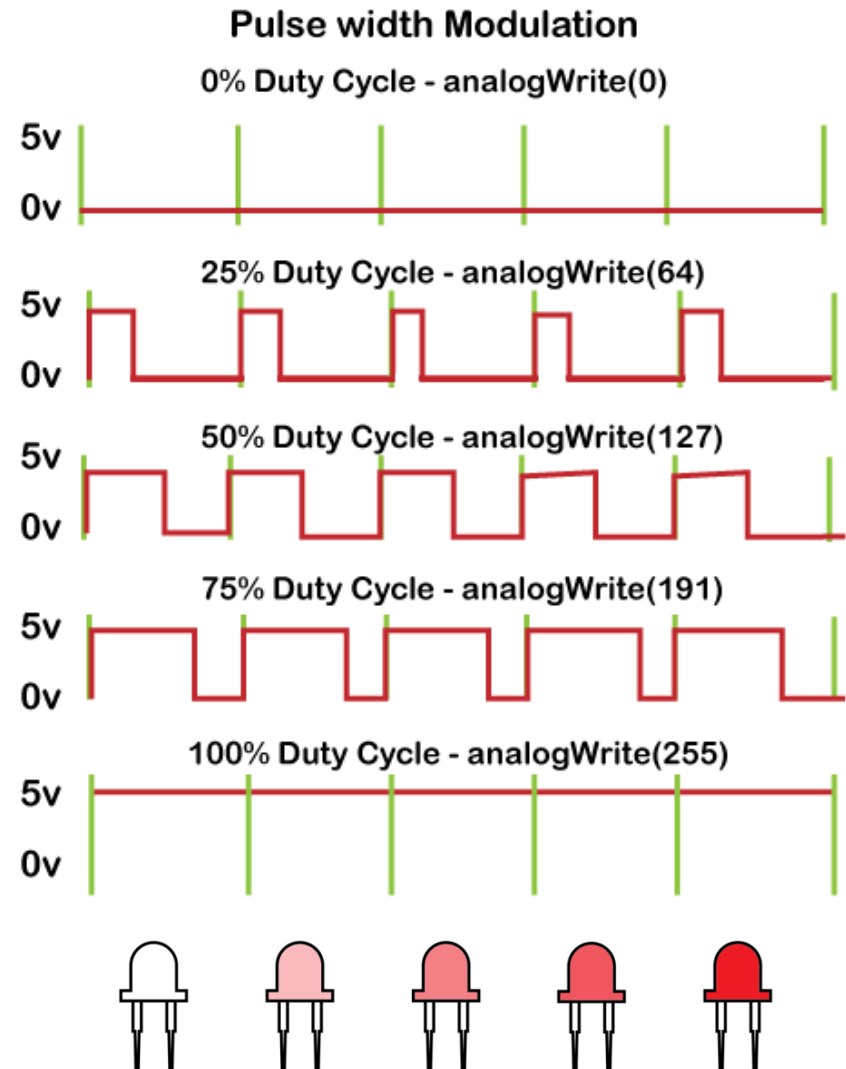
ข้อดี คือ ประมวลผลด้วยการเขียนโปรแกรม วงจรไม่สลับซับซ้อน สามารถทำงานได้แม่นยำกว่า

ข้อเสีย คือ ต้องมีความรู้ในการเขียนโปรแกรม มีความเข้าใจในสมการคณิตศาสตร์ และ **Chip** อาจจะมีราคาแพงกว่าการต่อวงจรอนาล็อก

4. การส่งค่า Analog Output ออก port (DAC)

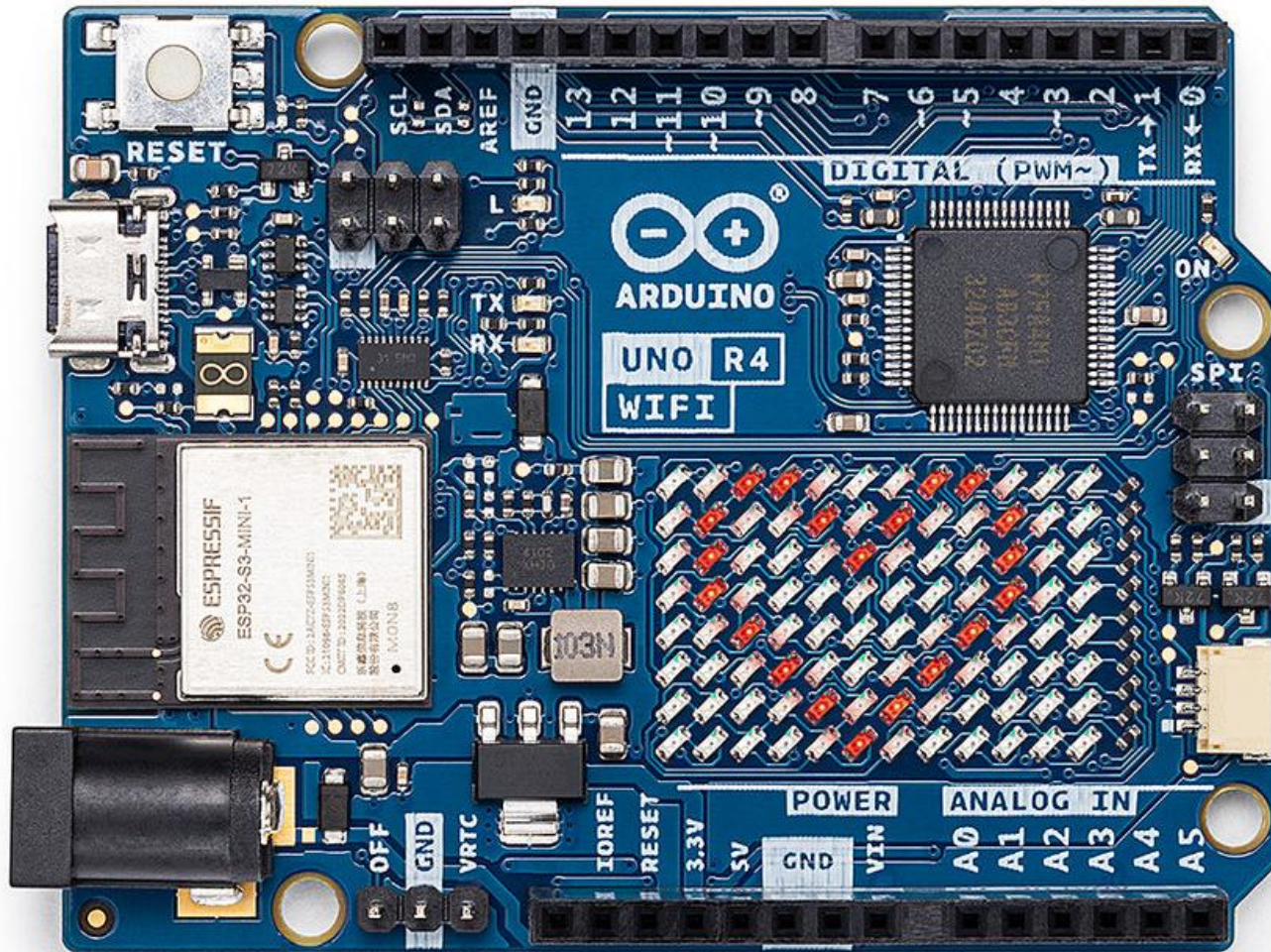
analogWrite()

- **Description**
- Writes an analog value ([PWM wave](#)) to a pin. Can be used to light a LED at varying brightnesses or drive a motor at various speeds. After a call to `analogWrite()`, the pin will generate a steady square wave of the specified duty. The frequency of the PWM signal is approximately 490 Hz.
- On most Arduino boards (those with the ATmega168 or ATmega328), **this function works on digital pins 3, 5, 6, 9, 10, and 11.**
- You do not need to call `pinMode()` to set the pin as an output before calling `analogWrite()`.



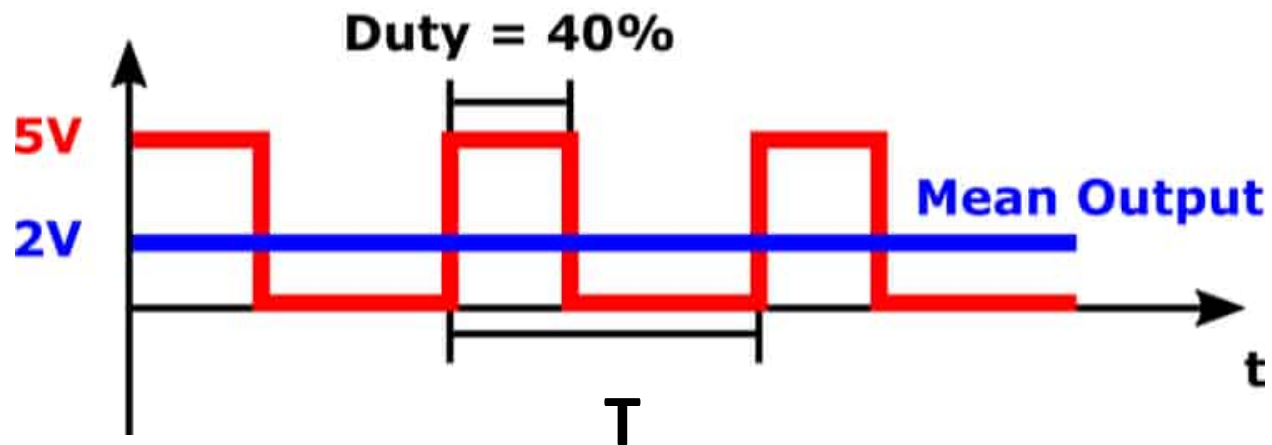
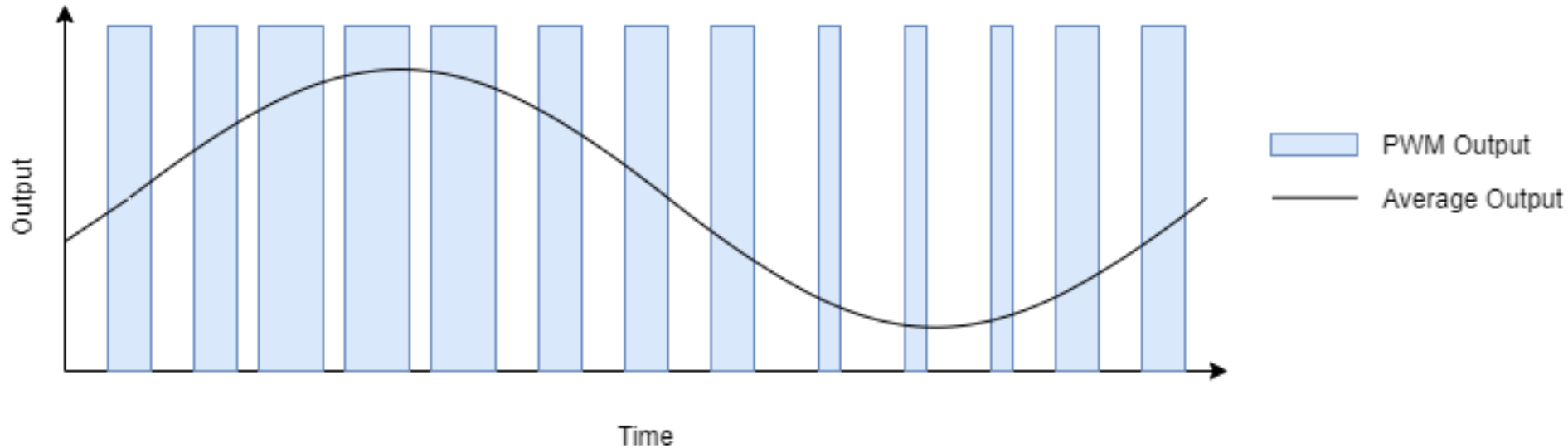
Arduino Uno R4 Board

Analog Output : ขาที่มีสัญลักษณ์ ~ (3,5,6,9,10,11)



Analog Input
(ADC)

PWM Signal

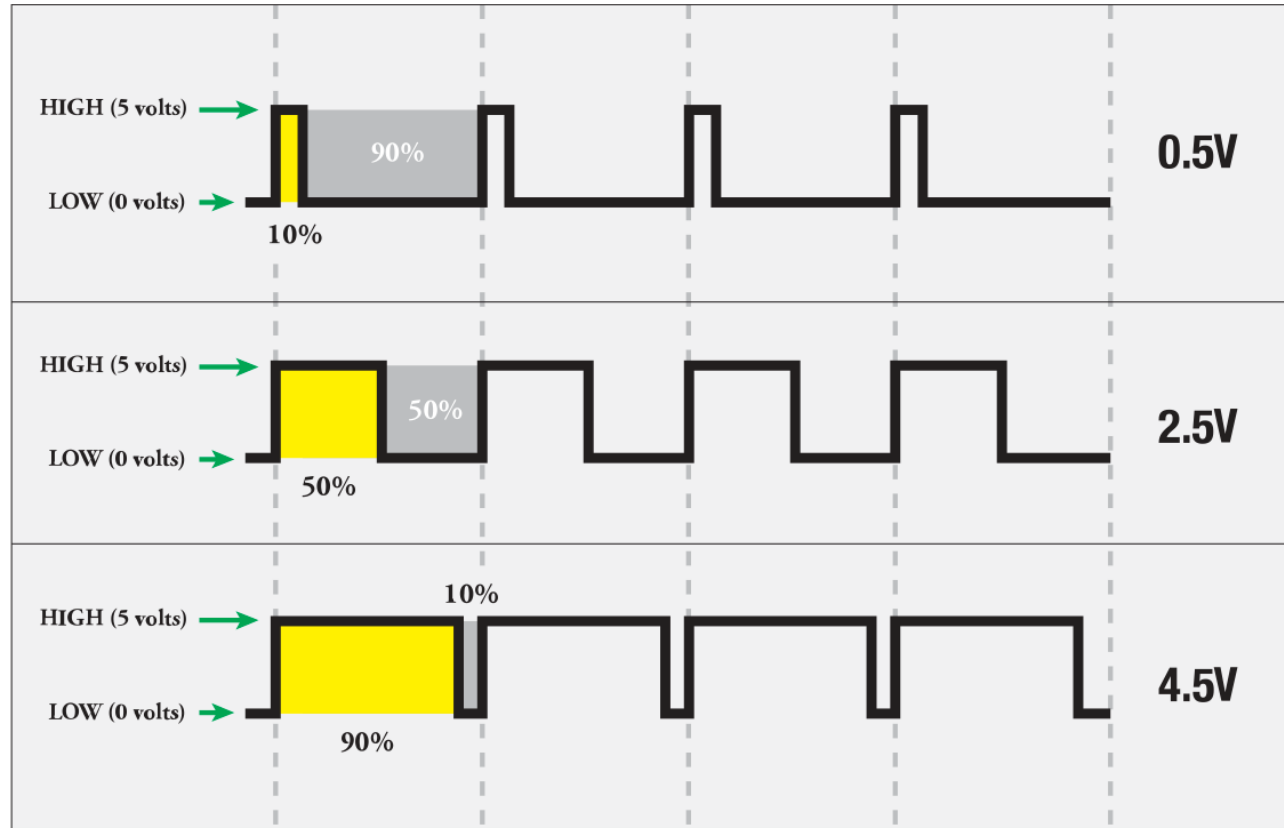
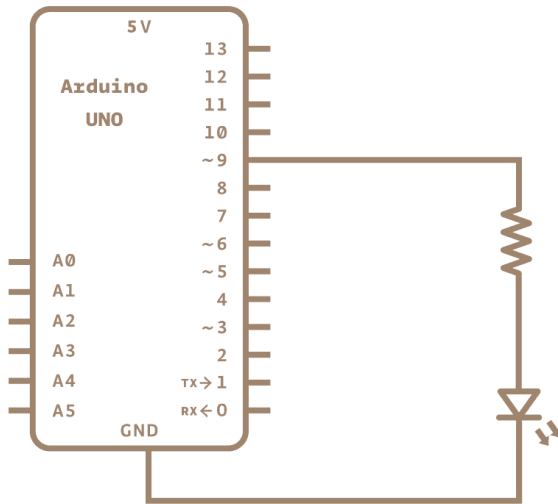


การส่งค่า Analog Output ออก port (DAC)

`analogWrite(9,25)`

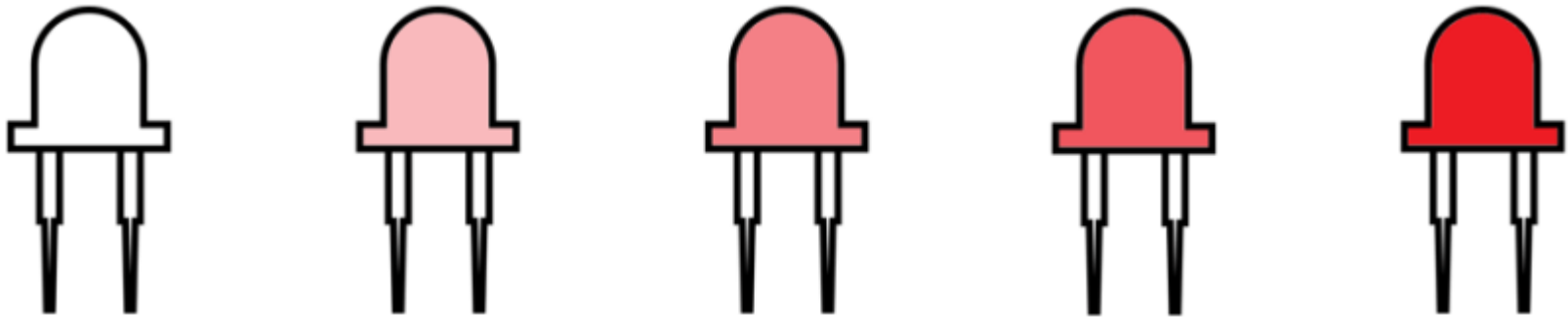
`analogWrite(9,127)`

`analogWrite(9,229)`



$$V_{\text{DC output}} = (\text{duty cycle}) \times V_{\text{cc}}$$

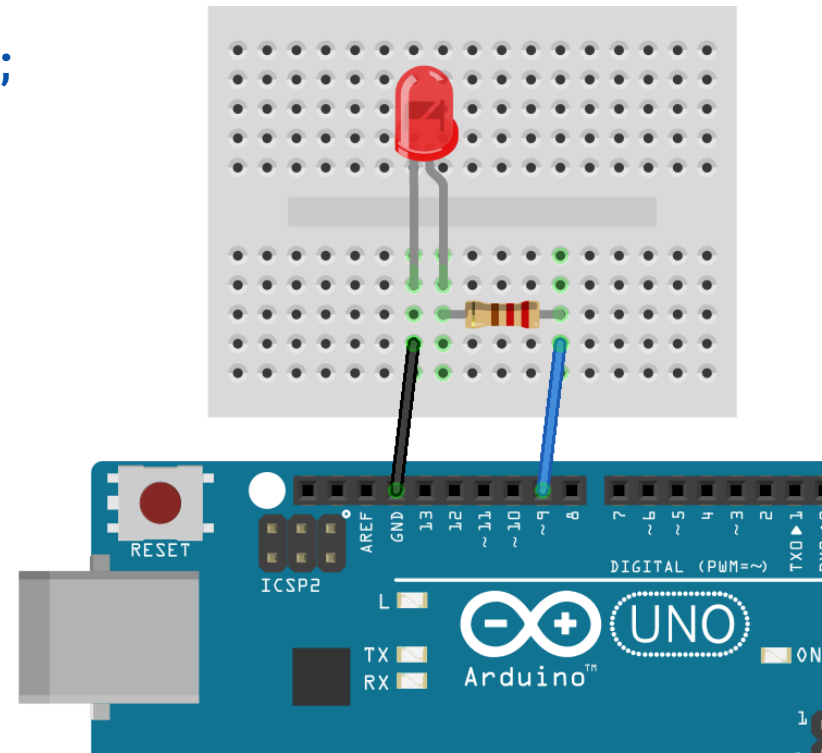
การทดลองที่ 3 . LED Fading : Off to Full bright



- The PWM pins work with the “`analogWrite(pin,value)`” command where
pin: the pin to write to.
value: the duty cycle: between 0 (always off) and 255 (always on).
- To turn LED to half-bright, use `analogWrite(9,128)`

โปรแกรม LED Fading : Off to Full bright

```
void setup() {  
    pinMode(9, OUTPUT);  
}  
  
void loop() {  
  
    for (int fadeValue = 0 ; fadeValue <= 255; fadeValue += 5)  
    {  
        analogWrite(9, fadeValue);  
        delay(30);  
    }  
}
```



Note !!

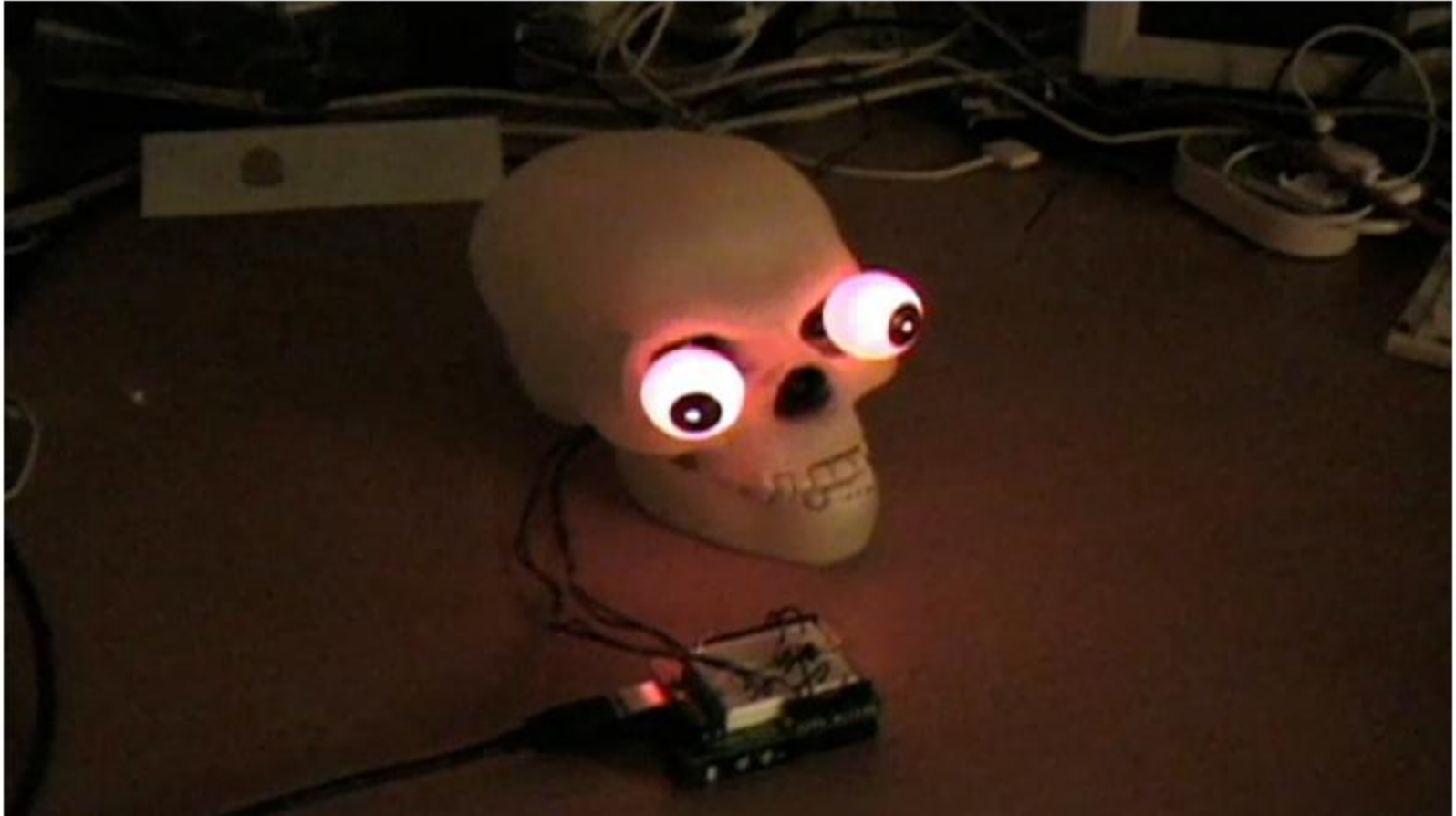
- สำหรับ Arduino uno R4 จะมีค่า **Default** ของ **Resolution DAC** เท่ากับ **8 bit**
 - ดังนั้น $\text{analogWrite}(255) = \text{duty cycle } 100\% = 5V$
- ถ้าต้องการปรับค่า **Resolution** สามารถทำได้โดยใช้คำสั่ง
 - $\text{analogWriteResolution}(xx)$ $xx = 8,9,10,11,12$
 - ใน UNO R4 นี้ มีค่า resolution สูงสุดคือ 12 บิต
 - เช่น $\text{analogWriteResolution}(12)$
 - $\text{analogWrite}(4095) = \text{duty cycle } 100\% = 5V$

แบบฝึกหัดที่ 1. จงเขียนโปรแกรม LED Fading

- เลือกใช้ PWM Pin อื่นๆ เป็นจำนวน 4 Pin
- โดยแต่ละ Pin มีการ Fading ดังนี้
 - LED1 Fading จาก Off ไปยัง Full-Bright
 - LED2 Fading จาก Off ไปยัง Full-Bright เร็วเป็นสองเท่าของ LED1
 - LED3 Fading จาก Full-Bright ไปยัง Off
 - LED4 Fading จาก Half-Bright ไปยัง Full-Bright



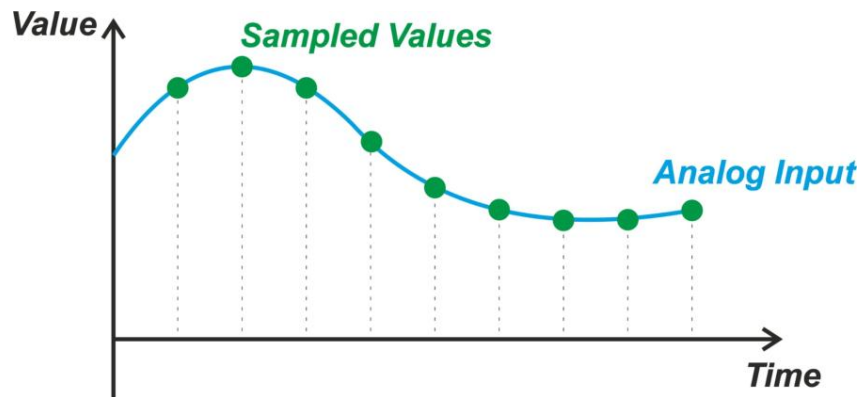
Evil Glowing Eyes



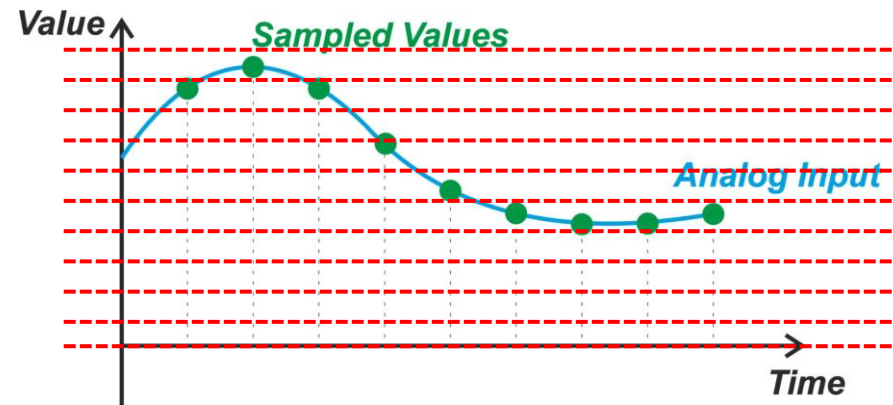
5. Analog to Digital Convertor (ADC)

กระบวนการแปลงสัญญาณ Analog ไปเป็นสัญญาณ Digital จะมีขั้นตอนหลักๆ คือ

- การ Sampling (ต้อง Sampling ด้วยความถี่ ที่มากกว่า 2 เท่าของความถี่สูงสุดของสัญญาณที่เข้ามา $f_s \geq 2f_m$)
- การ Encoding หรือ การ Quantize (กำหนดโดยค่า Resolution)



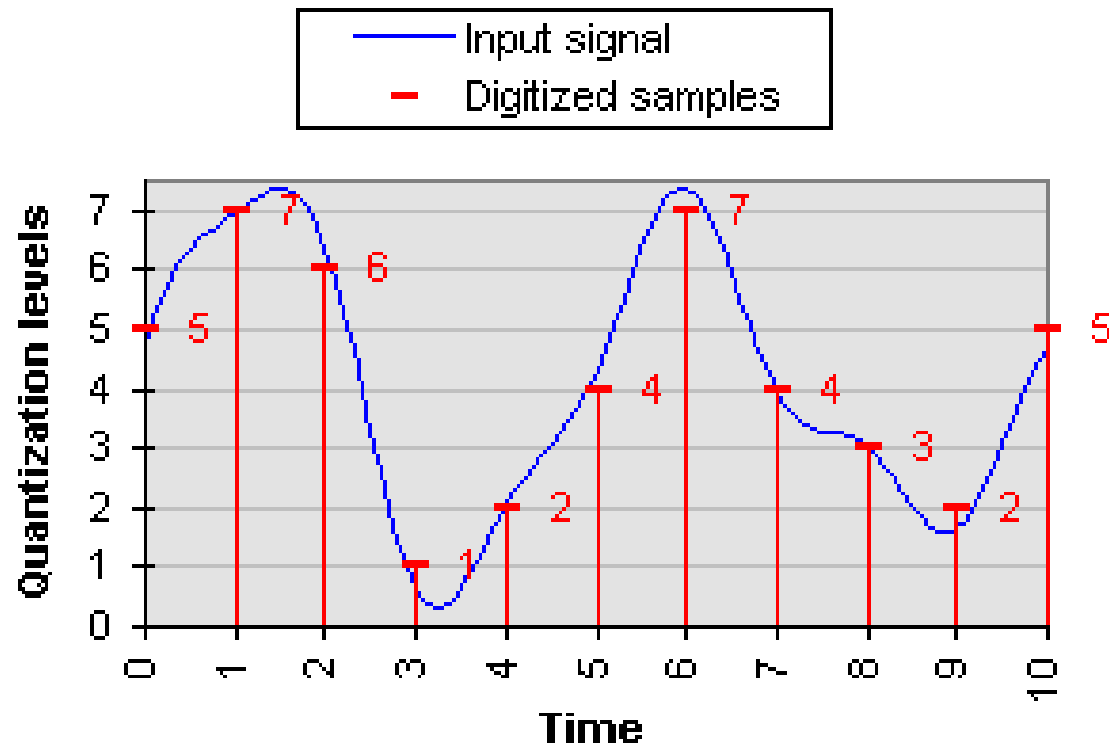
Sampling



Quantize

Analog to Digital

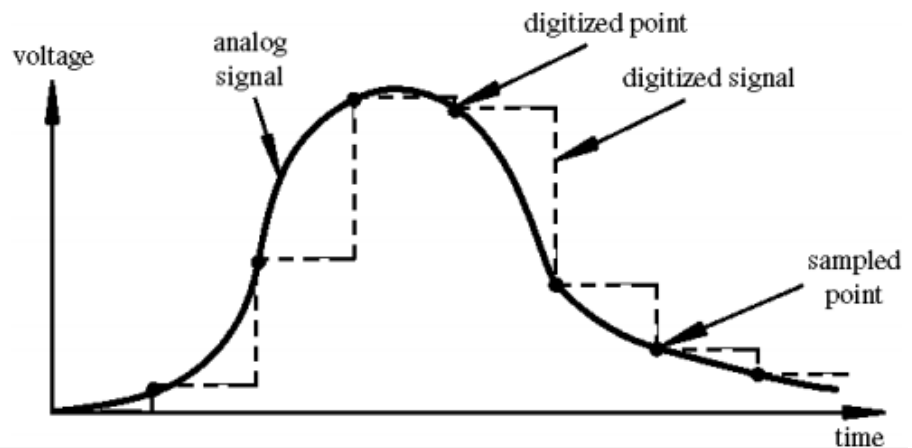
Quantizing and Digitizing a Signal



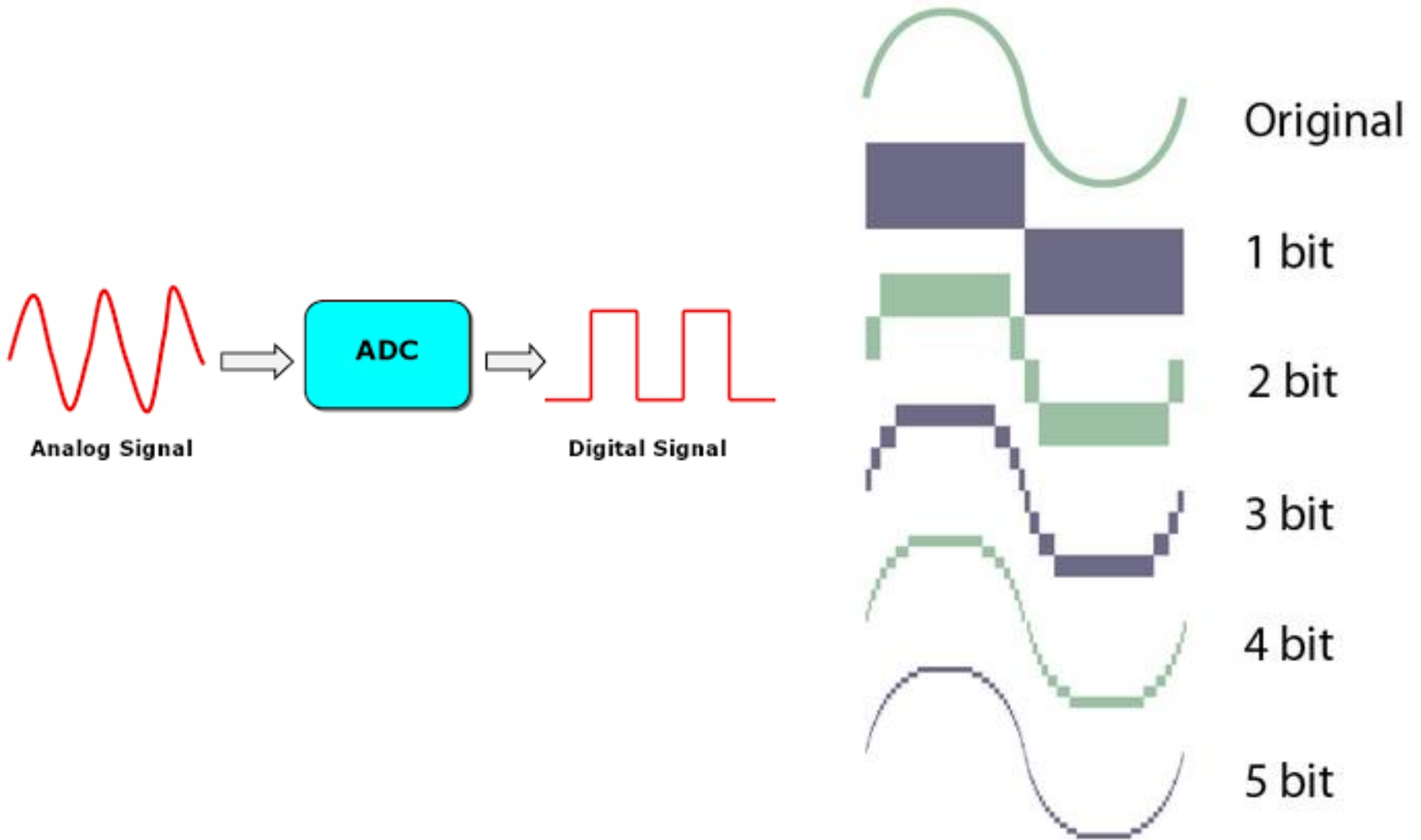
ถ้า sample ไม่ตรงระดับ จะเกิด Quantize Noise

Analog to Digital

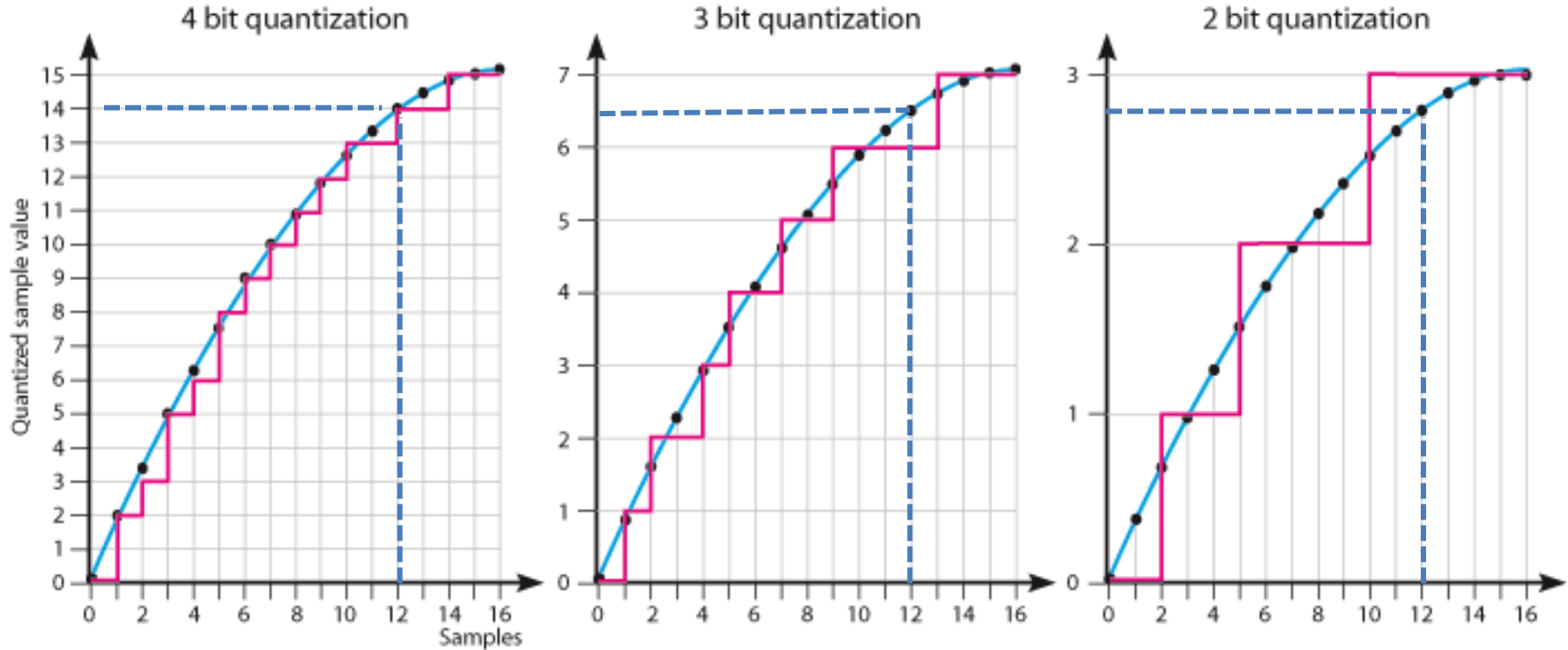
- Many states, not just two (HIGH/LOW)
- Number of states (or values, or “bins”) is *resolution*
- Common computer resolutions:
 - 8-bit = 256 values
 - 16-bit = 65,536 values
 - 32-bit = 4,294,967,296 values



6. Resolution



Resolution



พิจารณา ตัวอย่าง sample ที่ 12

4bit

อยู่ที่ระดับที่ 14 = 1110

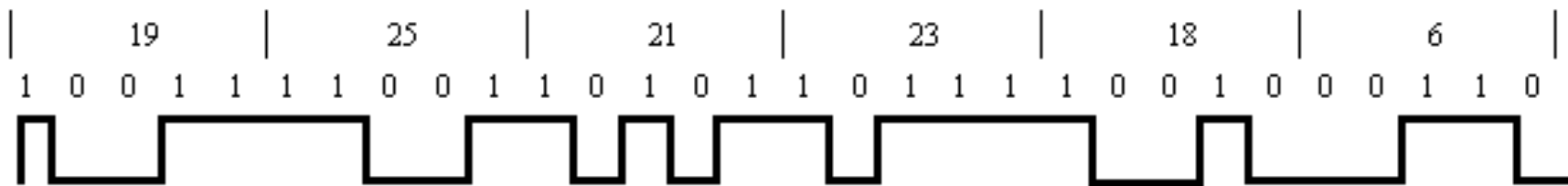
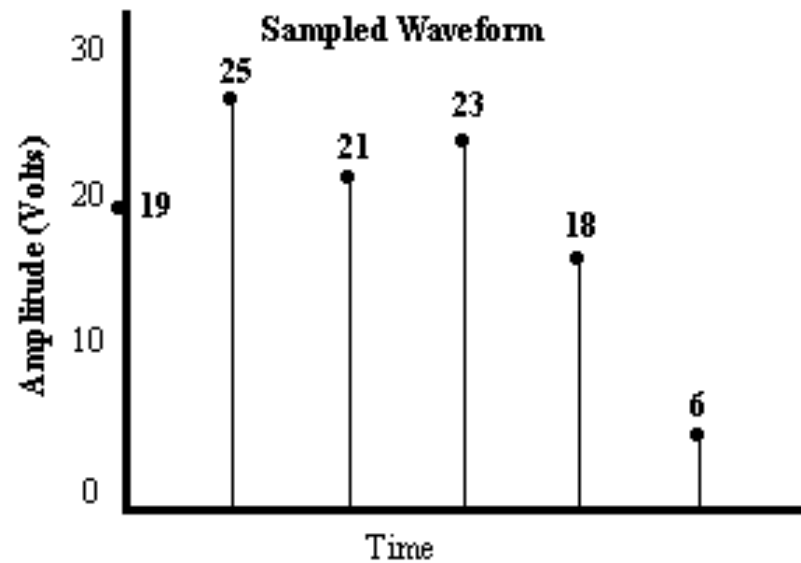
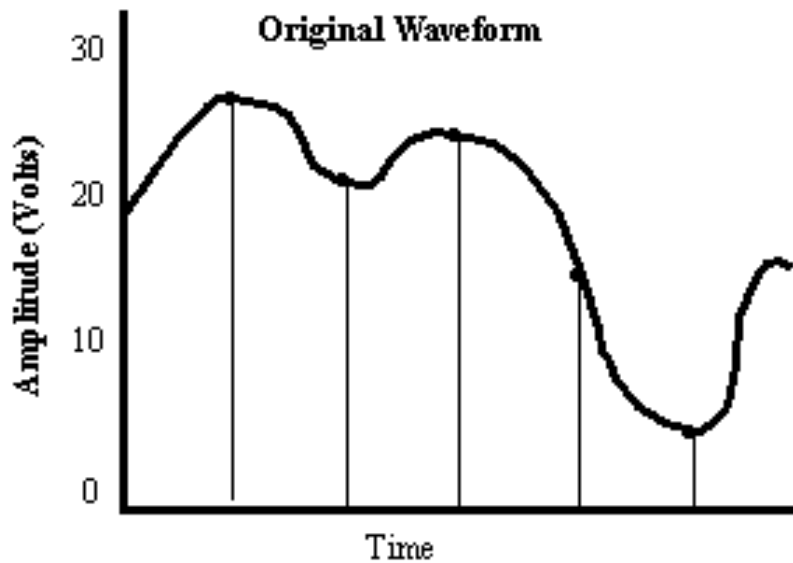
3 bit

อยู่ที่ระดับที่ 7 = 111

2 bit

อยู่ที่ระดับที่ 3 = 11

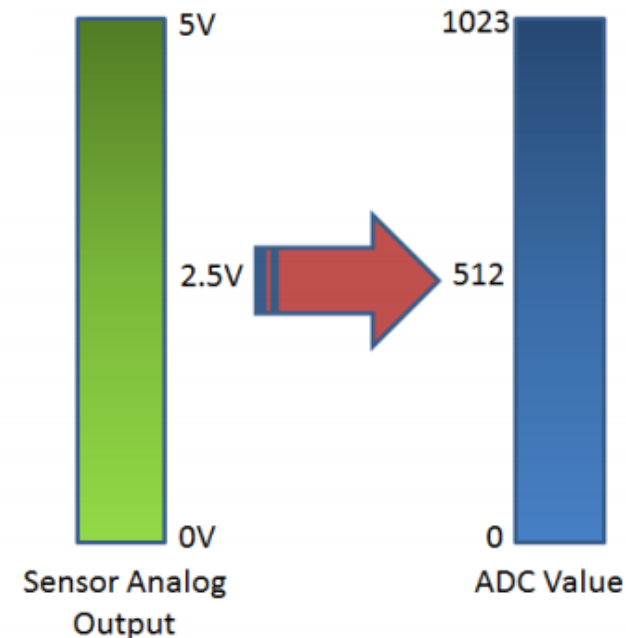
Resolution ตัวอย่าง 5 bit



Analog-to-Digital Conversion

Analog Input

- Arduino has six ADC inputs
- (ADC = Analog to Digital Converter)
- Reads voltage between 0 to 5 volts
- Resolution is 10-bit (1024 values)
- In other words, $5/1024 = 4.8$ mV smallest voltage change you can measure



Analog Input

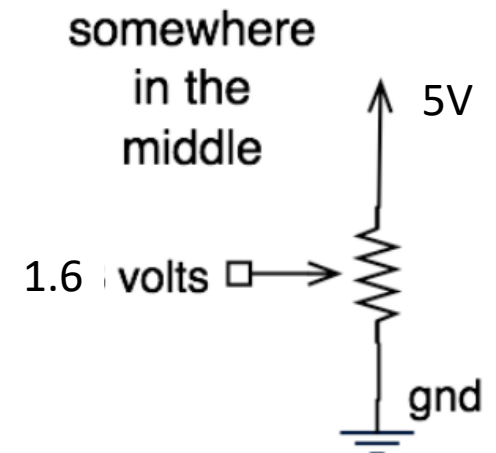
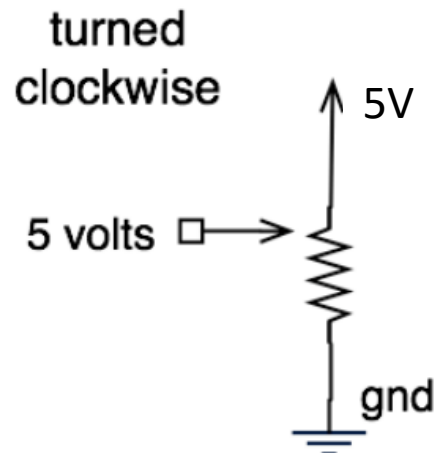
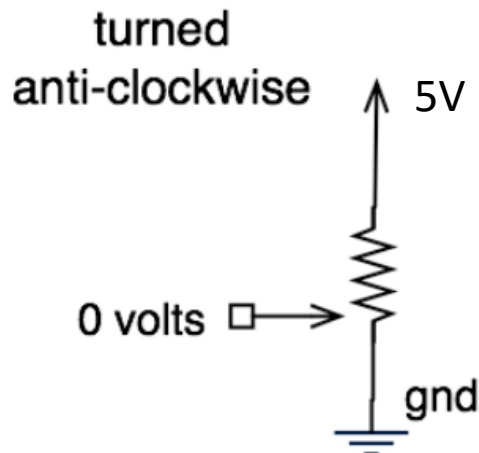
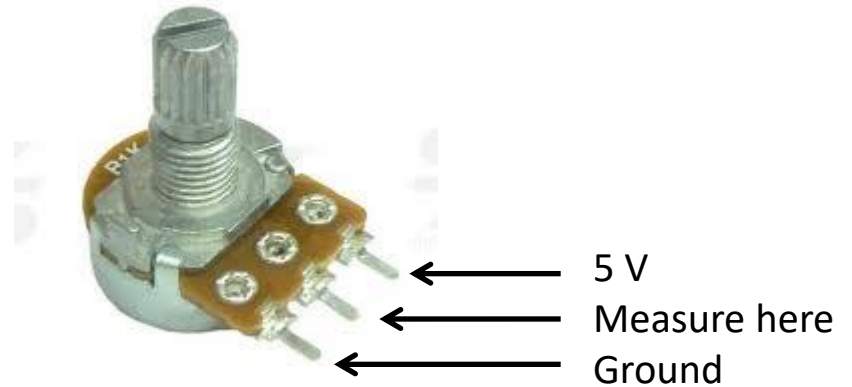
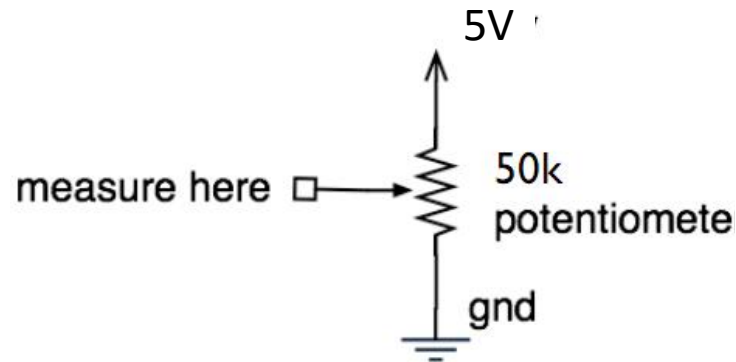
The default resolution is set to 10-bit, but can be updated to 12 and 14-bit resolutions. To do so, use the following method in the `setup()` of your sketch.

- `analogReadResolution(10)` (default)
- `analogReadResolution(12)`
- `analogReadResolution(14)`

```
void setup(){
    analogReadResolution(14); //change to 14-bit resolution
}

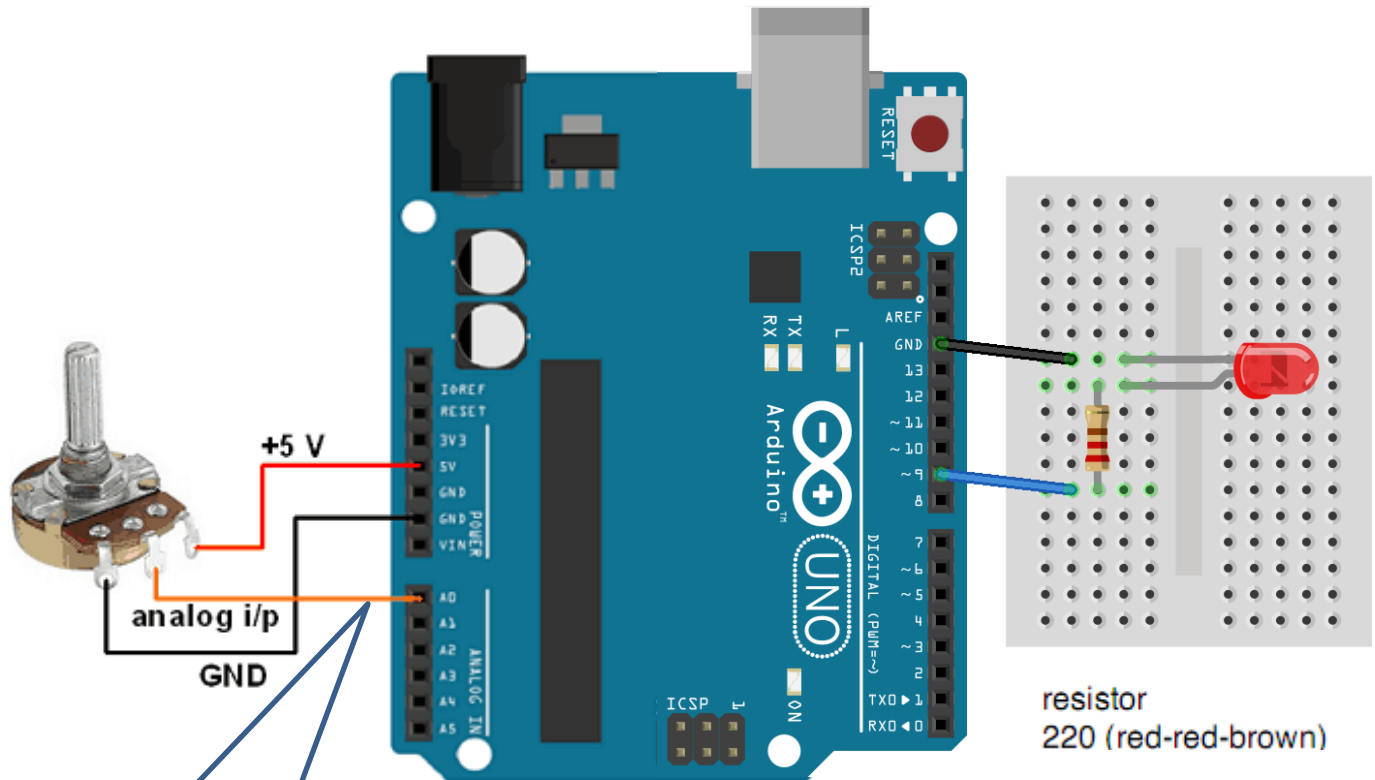
void loop(){
    int reading = analogRead(A3); // returns a value between 0-16383
}
```


How to make a varying voltage ?



การทดลองที่ 1. อ่านค่า สัญญาณ Analog

ต่อวงจรตามรูป



เมื่อหมุนตัวต้านทาน ค่าแรงดันที่ขา A0
จะมีค่าเปลี่ยนแปลงตั้งแต่ 0 ถึง 5 Volt


Note: คำสั่ง `analogRead` ใช้ได้กับขา Analog IN pin: A0, A1, ...A5
คำสั่ง `analogWrite` ใช้ได้กับขา Digital pin : 3, 5, 6, 9, 10, 11.

โปรแกรมอ่านค่าจากตัวต้านทานปรับค่าได้ แสดงค่าออก LED

```
void setup() {  
    Serial.begin(9600);  
    pinMode(9, OUTPUT);  
}  
  
void loop() {  
    int adc_value = analogRead(A0);  
    int brightness = map(adc_value, 0, 1023, 0, 255);  
    analogWrite(9, brightness);  
  
    Serial.print("Analog: ");  
    Serial.print(adc_value);  
    Serial.print(", Brightness: ");  
    Serial.println(brightness);  
    delay(100);  
}
```

- Note:** 1. การระบุขา analog input ต้องมีตัว A นำหน้า เช่นขา A0, A1,...A5
2. การใช้ขา analog input ไม่ต้องใช้คำสั่ง pinMode เนื่องจากทำหน้าที่เป็น input อยู่แล้ว

ผลลัพธ์ที่ได้

 COM6

```
Analog: 0, Voltage: 0.00
Analog: 0, Voltage: 0.00
Analog: 126, Voltage: 0.62
Analog: 281, Voltage: 1.37
Analog: 517, Voltage: 2.53
Analog: 754, Voltage: 3.69
Analog: 906, Voltage: 4.43
Analog: 1023, Voltage: 5.00
Analog: 1023, Voltage: 5.00
```

☐ Autoscroll ☐ Show timestamp Newline ▼

การทดลองที่ 2. อ่านค่า Sensor อุณหภูมิ

แสดงขาของ Sensor อุณหภูมิ MCP9700 และกราฟคุณสมบัติของเซ็นเซอร์ ดังรูป

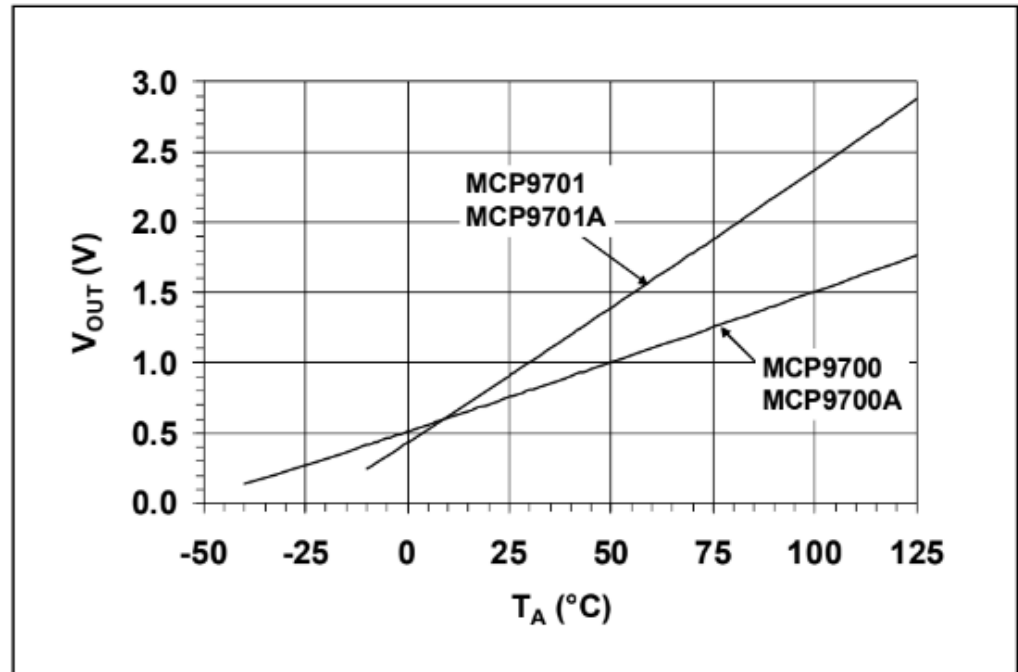
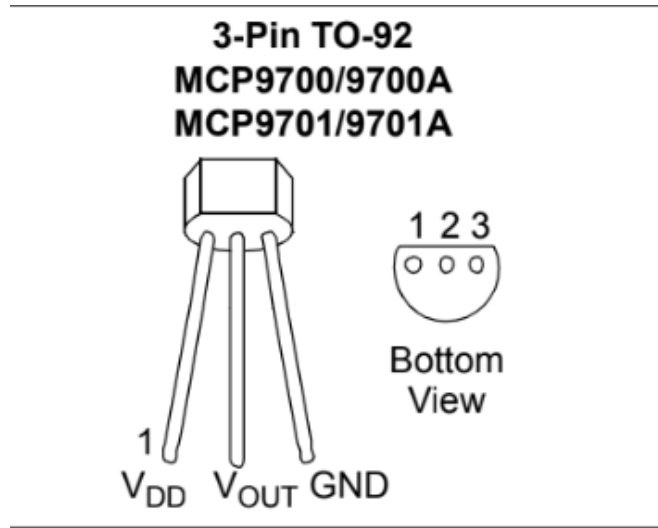


FIGURE 2-16: Output Voltage vs. Ambient Temperature.

จากกราฟ เห็นว่า ที่อุณหภูมิ 50 องศา แรงดันเอาต์พุต มีค่า 1 โวลต์
ที่อุณหภูมิ 125 องศา แรงดันเอาต์พุต มีค่า 1.75 โวลต์

คำนวณหาความสัมพันธ์ของ อุณหภูมิกับแรงดันเอาต์พุต

จากกราฟ เห็นว่า ที่อุณหภูมิ 50 องศา แรงดันเอาต์พุต มีค่า 1 โวลต์

ที่อุณหภูมิ 125 องศา แรงดันเอาต์พุต มีค่า 1.75 โวลต์

คำนวณหา ความชันของกราฟ

$$m = \frac{\Delta y}{\Delta x} = \frac{\Delta \text{Output Voltage}}{\Delta \text{Temp}} = \frac{1.75 - 1}{125 - 50} = \frac{0.75}{75} = \frac{7.5}{750}$$

คำนวณหา สมการเส้นตรง

$$y - y_1 = m(x - x_1)$$

$$\text{Output Voltage} - \text{Output Voltage}_1 = m(\text{Temp} - \text{Temp}_1)$$

$$\text{Output Voltage} - 1 = \frac{7.5}{750}(\text{Temp} - 50) \quad \text{หรือจัดรูปได้ว่า}$$

$$\text{Temp} = \frac{750}{7.5}(\text{Output Voltage} - 1) + 50$$

คำนวณหาความสัมพันธ์ของ อุณหภูมิกับแรงดันเอาต์พุต

จากสมการ

$$\text{Temp} = 100 * (\text{Output Voltage} - 1) + 50$$

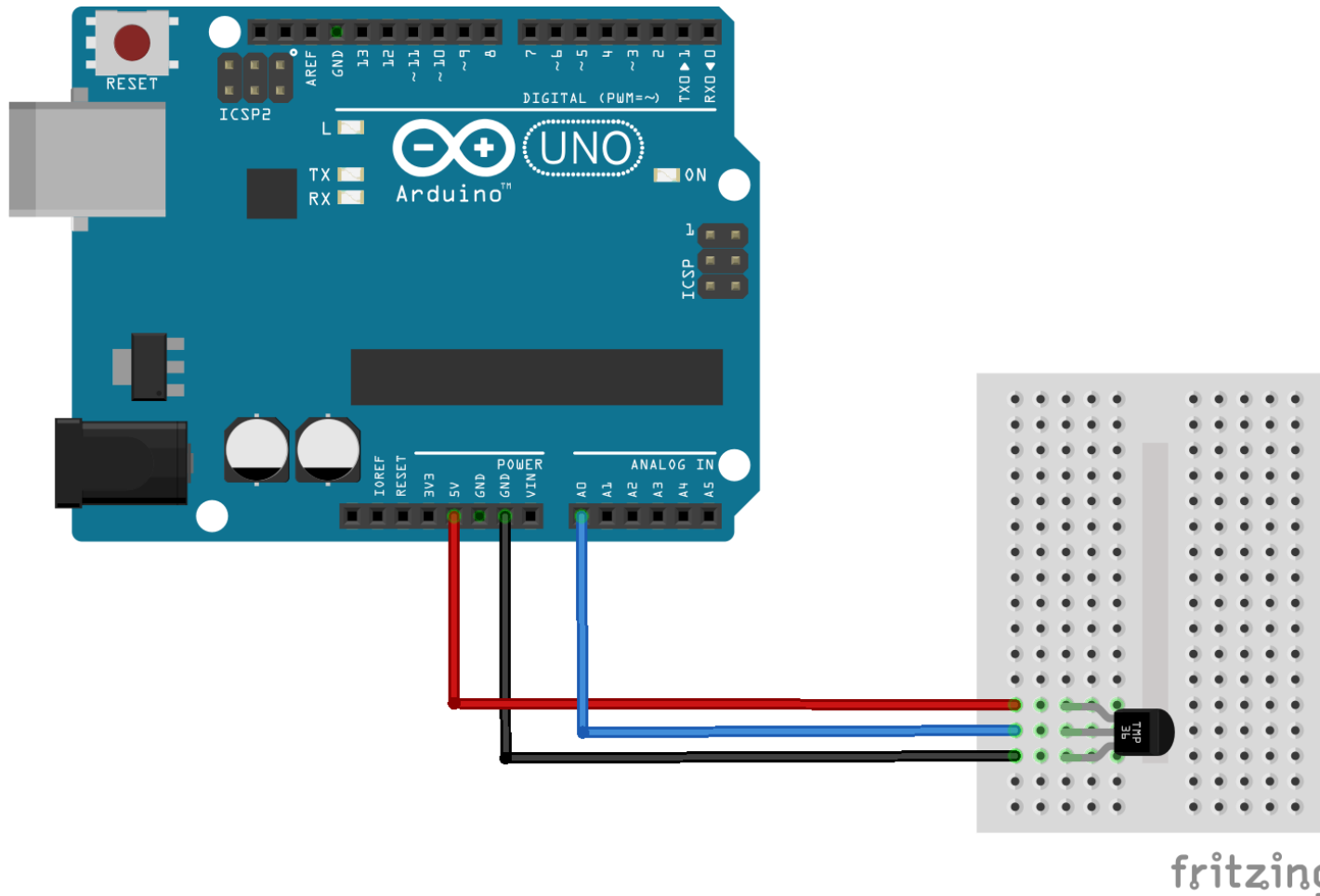
เนื่องจากค่า Voltage ที่อ่านได้จาก A/D เป็นค่าดิจิทัลขนาด 10 bit
ซึ่งแปลงมาจากค่า Analog Voltage ช่วง 0-5V และได้ค่าดิจิทัลออกเป็น 0 - 1023

ถ้ากำหนดให้ Val = ค่าดิจิทัลที่อ่านได้
จะได้ว่า $\text{Analog Voltage} = (\text{Val} * 5) / 1023$

แทนค่า Analog Voltage ลงไปในสมการ Temp จะได้ว่า
$$\text{Temp} = 100 * ((\text{Val} * 5) / 1023 - 1) + 50$$

การทดลองที่ 2. อ่านค่า **Sensor** อุณหภูมิ*

ตัวอย่าง **Sensor** อุณหภูมิ และเขียนโปรแกรมแสดงค่าอุณหภูมิที่วัดได้



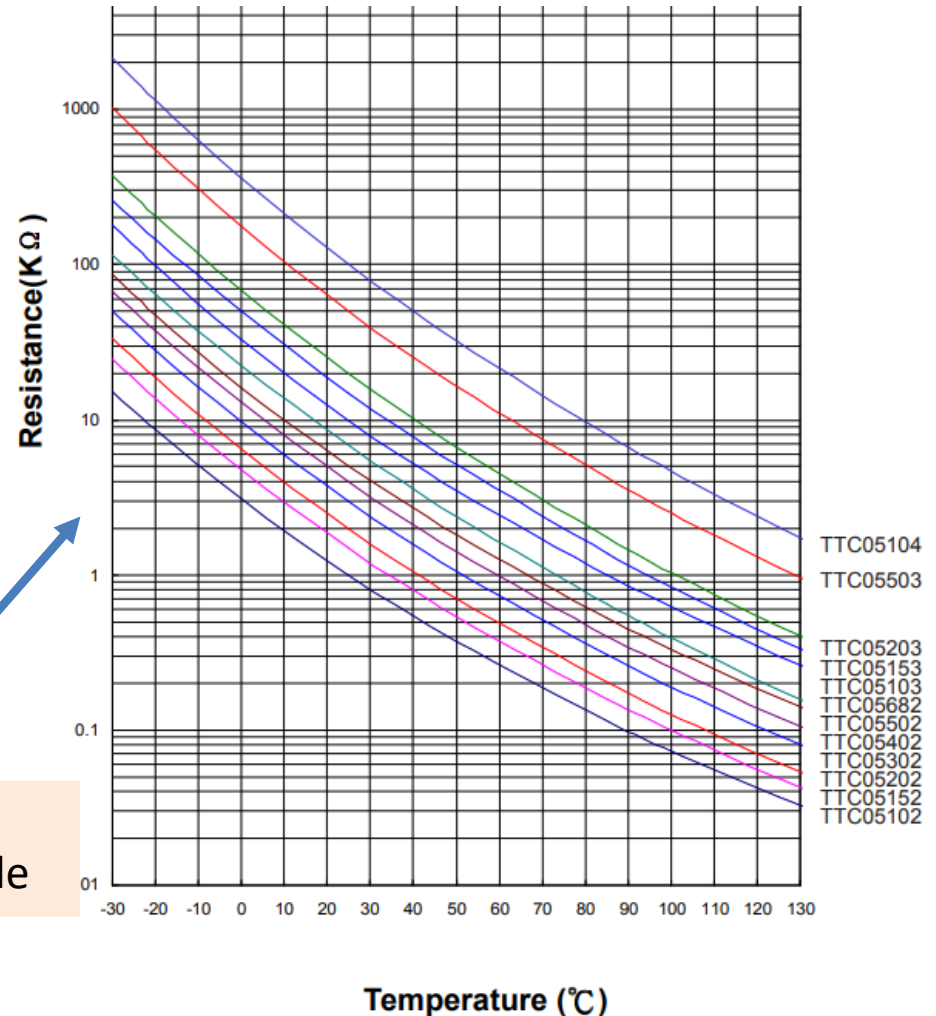
การทดลองที่ 3. Thermistor

Thermistor คือ เป็นอุปกรณ์ที่ค่าความต้านทาน จะเปลี่ยนแปลงตาม อุณหภูมิที่เปลี่ยนแปลง



เบอร์ TTC05103

ข้อสังเกต
แกนตั้งเป็น Log Scale



Thermistor Equation

Thermistor Equation

$$B_{(T_1/T_2)} = \frac{T_2 \times T_1}{T_2 - T_1} \times \ln\left(\frac{R_1}{R_2}\right)$$

Where:

T_1 is the first temperature point in Kelvin

T_2 is the second temperature point in Kelvin

R_1 is the thermistors resistance at temperature T_1 in Ohms

R_2 is the thermistors resistance at temperature T_2 in Ohms

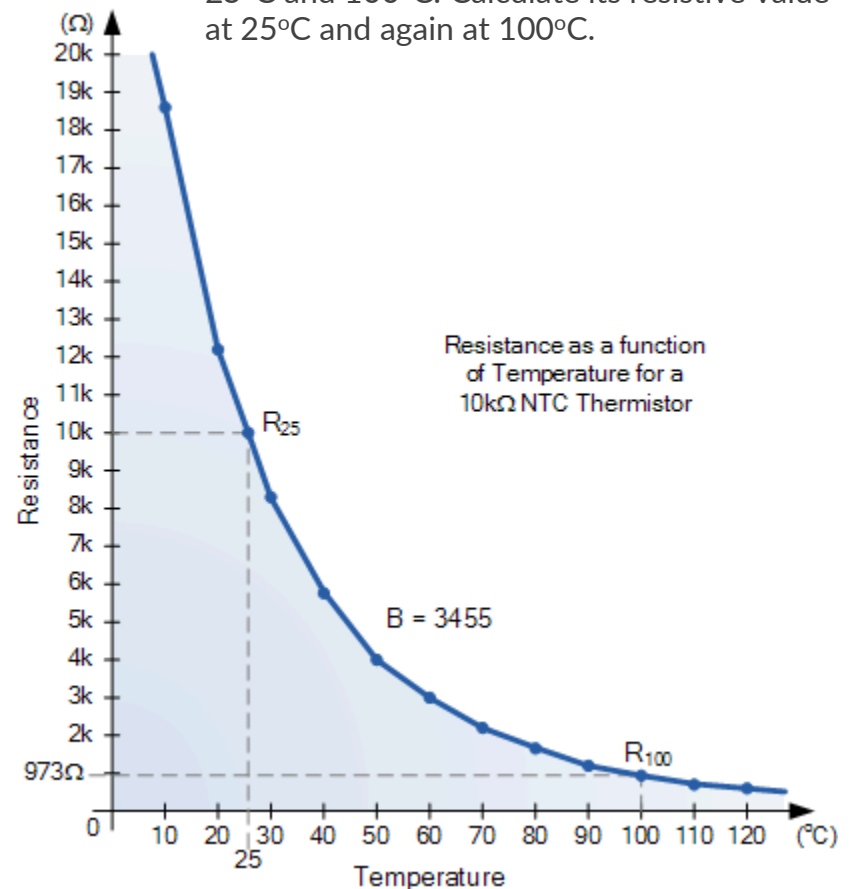
จากสมการ Thermistor Equation

ทำการจัดรูปสมการ แทน $R_2 = R_0$, $T_2 = T_0$ จะได้ว่า

$$T = \frac{T_0 \cdot B}{T_0 \cdot \ln(R/R_0) + B}$$

$T_0 = 298.15K$ (Note that the formula uses Kelvin,
 R_0 is given for $25^\circ C = 298.15K$)

A $10k\Omega$ NTC thermistor has a “B” value of 3455 between the temperature range of $25^\circ C$ and $100^\circ C$. Calculate its resistive value at $25^\circ C$ and again at $100^\circ C$.



Thermistor เบอร์ TTC05103

Datasheet <https://www.es.co.th/Schemetic/PDF/TTC05.PDF>

■ Electrical Characteristics

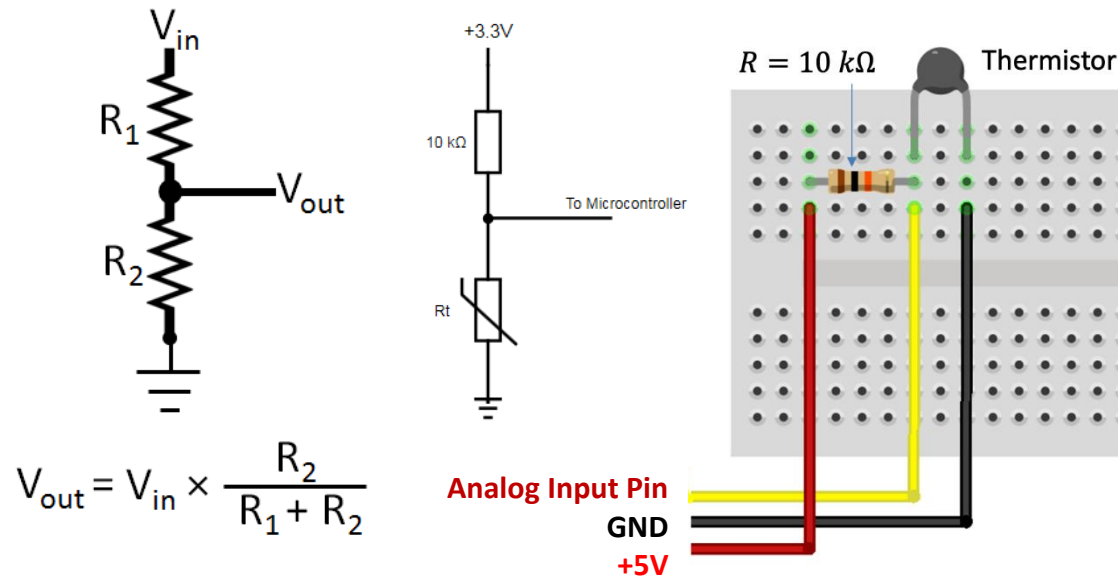
Part No.	Zero Power Resistance at 25°C	Tolerance of R ₂₅	B _{25/50} Value	Max. Power Rating at 25°C	Dissipation Factor	Thermal Time Constant	Operating Temperature Range	Safety Approvals			
	R ₂₅ (Ω)	(±%)	(K)	P _{max} (mW)	δ(mW/°C)	τ(Sec.)	T _L ~T _U (°C)	UL	CSA	TUV	CQC
TTC05005□	5		2400						✓	✓	✓
TTC05010□	10		2800						✓	✓	✓
TTC05015□	15		2800					✓	✓	✓	✓
TTC05020□	20		2800					✓	✓	✓	✓
TTC05025□	25		2900					✓	✓	✓	✓
TTC05045□	45		3100					✓	✓	✓	✓
TTC05050□	50		3100					✓	✓	✓	✓
TTC05060□	60		3100					✓	✓	✓	✓
TTC05085□	85		3200					✓	✓	✓	✓

TTC05802□	8000		4050					✓	✓	✓	✓
TTC05103□	10000		4050					✓	✓	✓	✓
TTC05123□	12000		4050					✓	✓	✓	✓
TTC05153□	15000		4150					✓	✓	✓	✓
TTC05203□	20000		4250					✓	✓	✓	✓
TTC05303□	30000		4250					✓	✓	✓	✓
TTC05473□	47000		4300					✓	✓	✓	✓

↑
R₀

↑
B

การทดลองที่ 3. Thermistor



ต่อวงจรตามรูปจะเห็นว่า
ค่าแรงดันที่เข้าขา **input** ของ
ADC จะมีค่า

$$V_{out} = \frac{R_{\text{thermistor}}}{(10\text{k} + R_{\text{thermistor}})} (5)$$

จงเขียนโปรแกรม แสดงค่าอุณหภูมิที่วัดได้ เทียบกับค่าที่ได้จาก **mcp9700**

Hint : แปลงแรงดัน เป็นค่าความต้านทาน แล้วแปลงไปเป็น อุณหภูมิ

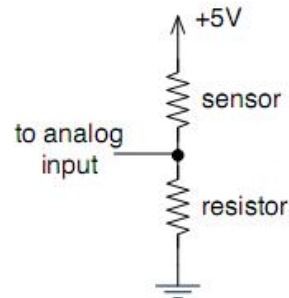
แนวทางการพัฒนา

- นำไปใช้อ่านค่าจาก **Resistive Sensor** ต่างๆ ได้ เช่น



thermistor
(temperature)

circuit is the same
for all these



photocell
(light)

.2"



.5"

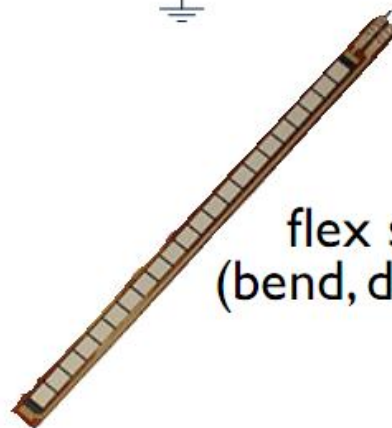


force sensors
(pressure)

1.5"



flex sensor
(bend, deflection)



also air pressure
and others

Flex sensors

- **Flex sensors** change their resistance when they are bent at varying angles: they're often used in interactive gloves

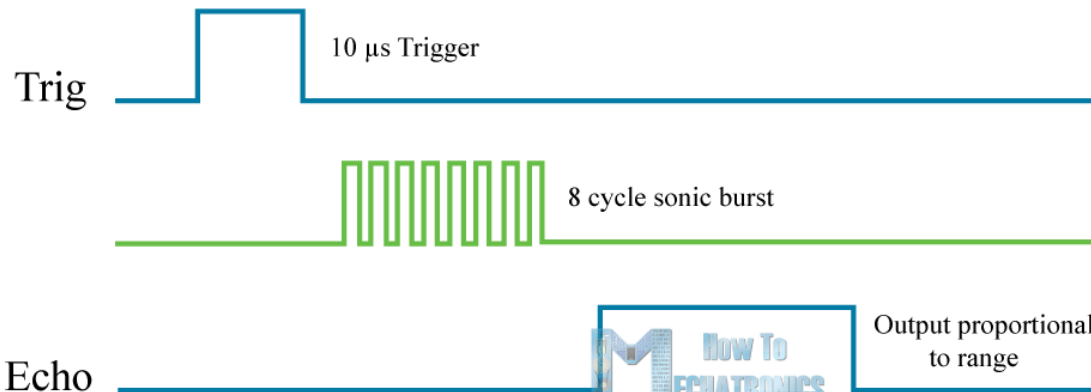


Ultrasonic Module



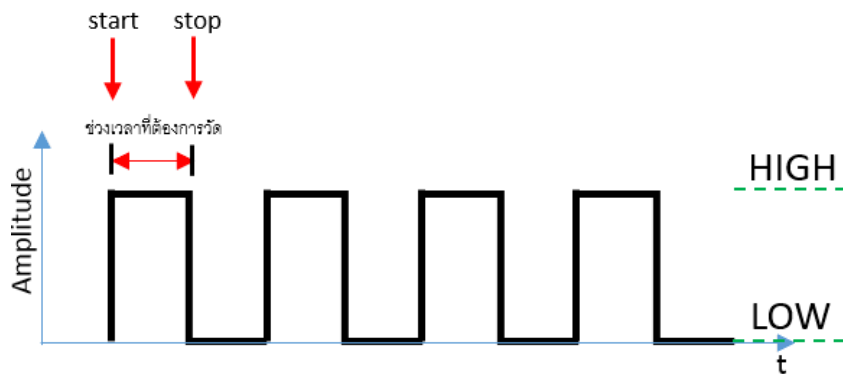
โมดูล HC-SR04 เป็นอุปกรณ์อิเล็กทรอนิกส์ราคา
ถูก สำหรับวัดระยะห่างด้วยคลื่นอัลตราโซนิก (ใช้
คลื่นเสียงความถี่ ประมาณ 40kHz)

ในการวัดระยะห่างแต่ละครั้ง จะต้องสร้างสัญญาณแบบ **Pulse** ที่มีความกว้าง (Pulse Width) อย่าง
น้อย 10 micro second ป้อนให้ขา **TRIG** และหลังจากนั้นให้วัดความกว้างของสัญญาณช่วง **HIGH**
จากขา **ECHO** ถ้าวัตถุอยู่ใกล้ ความกว้างของสัญญาณ **Pulse** ที่ได้ก็จะน้อย แต่ถ้าวัตถุอยู่ไกลออกไป
ก็จะได้ค่าความกว้างของสัญญาณ **Pulse** ที่มากขึ้น (ความกว้าง = ระยะเวลาที่ไป-กลับ)

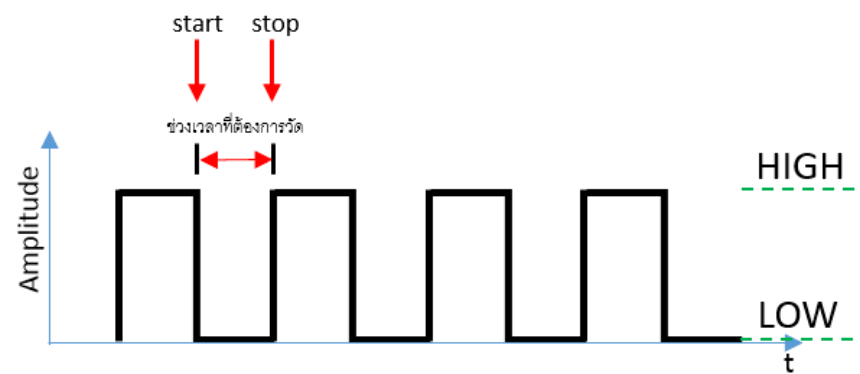


การใช้งานคำสั่ง `pulseIn`

- ใช้วัดความกว้าง Pulse ของสัญญาณ



(a)



(b)

Syntax

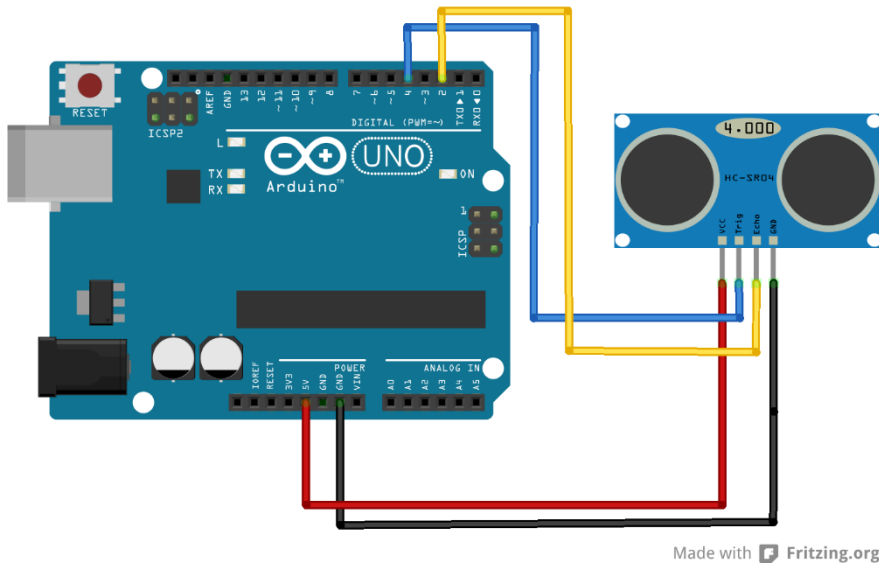
```
pulseIn(pin, value)  
pulseIn(pin, value, timeout)
```

รูป (a) value = HIGH รูป (b) value = LOW

timeout คือช่วงเวลาสูงสุดที่ฟังก์ชันนี้ยังทำงานอยู่ ค่า default คือ 1 วินาที หรือ 1,000,000 ไมโครวินาที

การทดลองที่ 4. การใช้งาน `pulseIn`

โปรแกรมวัดระยะทาง แสดงผลออก Serial monitor



The speed of sound is 340 m/s
or 29 microseconds per
centimeter

```
void setup()
{
    Serial.begin(9600);
    pinMode(4, OUTPUT);    // 4 = Trig
    pinMode(2, INPUT);     // 2 = Echo
}
```

```
void loop()
{
    digitalWrite(4, HIGH);
    delayMicroseconds(10);
    digitalWrite(4, LOW);

    int pulseWidth = pulseIn(2, HIGH);

    Serial.print("Pulse Width: ");
    Serial.println(pulseWidth);
    long distance = pulseWidth/29/2;
    Serial.print("Distance: ");
    Serial.println(distance);
    delay(1000);
}
```