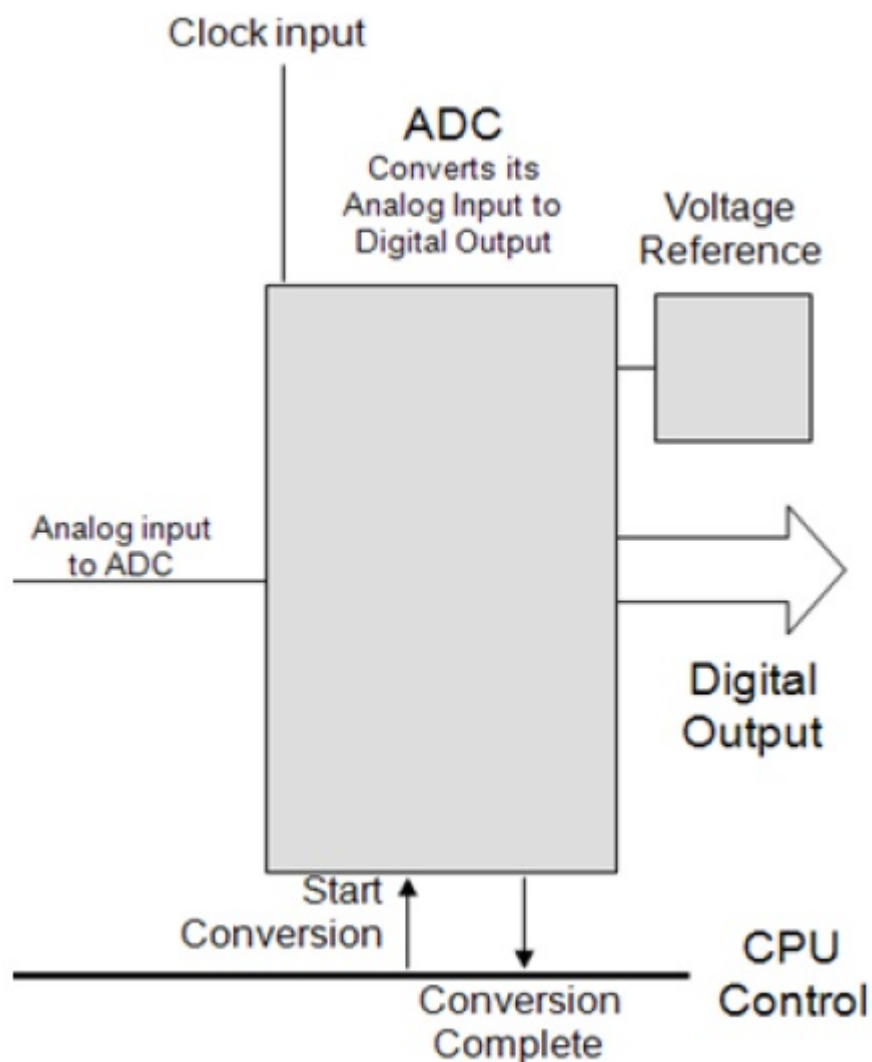


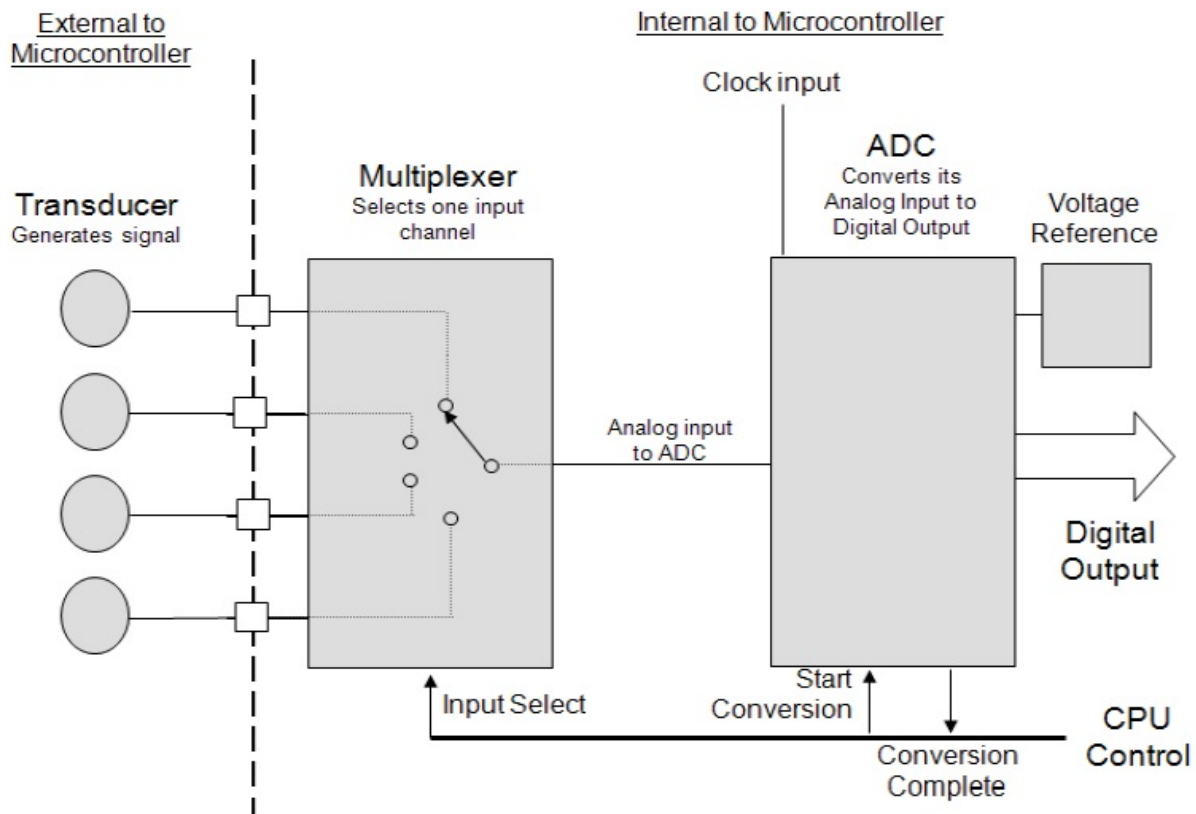
ใบความรู้ที่ 8 การใช้ thingcontrol board กับอินพุตแบบอนาลอก

1. ตัวแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล (Analog-to-Digital Converter)

ตัวแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล จะเป็นวงจรอิเล็กทรอนิกส์ที่ทำการรับค่าอินพุตแบบอนาลอกในรูปแบบแรงดันไฟฟ้าและทำการแปลงเป็นสัญญาณดิจิทัลในรูปเลขฐานสอง ในสัดส่วนของแรงดันไฟฟ้าของอินพุตอนาลอกที่รับเข้ามา ในการแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล เราจำเป็นต้องรู้แรงดันอ้างอิง (Voltage Reference) เพื่อใช้ในการเปรียบเทียบกับค่าอินพุตแบบอนาลอกในรูปแบบแรงดันไฟฟ้า และทำการแปลงเป็นค่าตัวเลขในรูปเลขฐานสอง



โดยทั่วไปตัวแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล จะทำงานในจำนวนช่องทางการการรับข้อมูลจำนวนมาก จนบางครั้งเราเรียกว่าระบบการรับข้อมูล (Data Acquisition System) ซึ่งโดยทั่วไปจะมีรูปแบบการทำงานดังนี้



2.ความสัมพันธ์ของสัญญาณอนาลอกอินพุตกับสัญญาณดิจิทัลเอาต์พุตมีค่าดังนี้

$$D = V_i / V_r \times 2^n$$

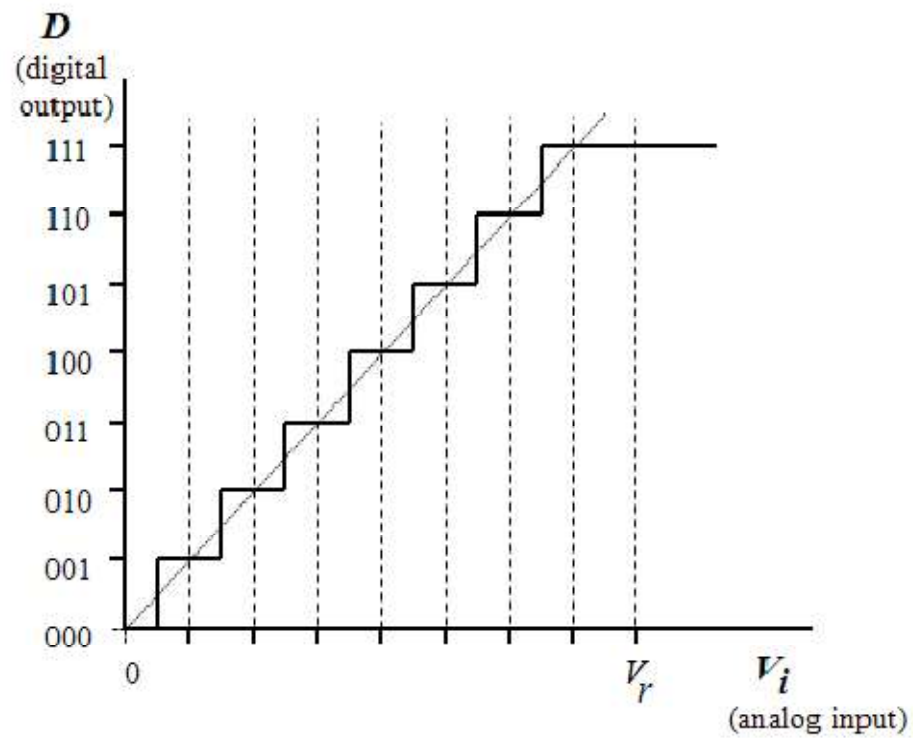
D = Digital output

V_i = Analog input

V_r = Voltage Reference

n = n-bit of Digital output (resolution)

3.คุณสมบัติของตัวแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล



4.ตัวอย่าง

Analog input

$$D = 0/3.3 \times 2^3 - 1 = 0$$

$$= 0.4714/3.3 \times 2^3 - 1 = 1$$

$$= 0.9429/3.3 \times 2^3 - 1 = 2$$

$$= 1.4142/3.3 \times 2^3 - 1 = 3$$

$$= 1.8857/3.3 \times 2^3 - 1 = 4$$

$$= 2.3571/3.3 \times 2^3 - 1 = 5$$

$$= 2.8285/3.3 \times 2^3 - 1 = 6$$

$$= 3.3/3.3 \times 2^3 - 1 = 7$$

Digital output

0 = 000

1 = 001

2 = 010

3 = 011

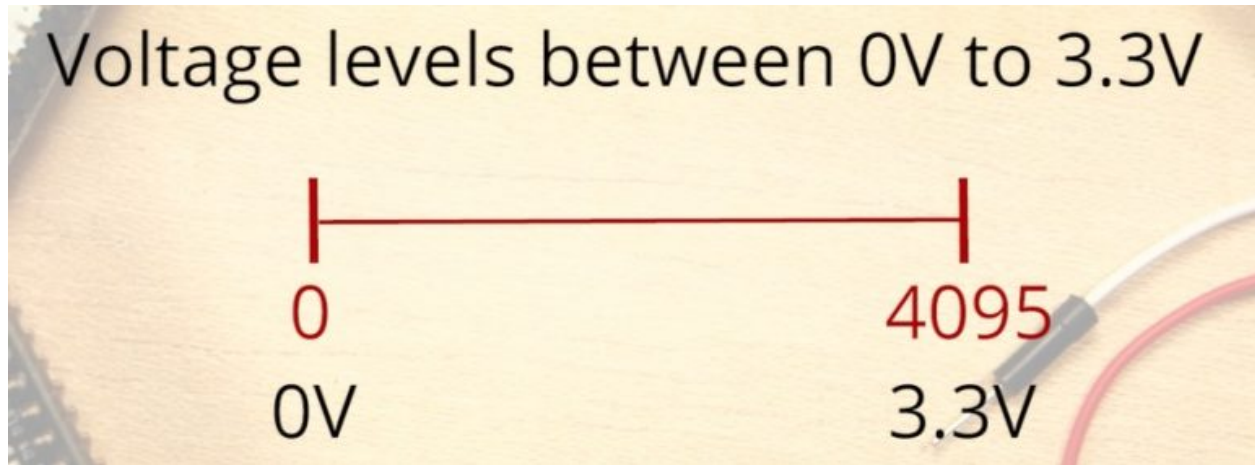
4 = 100

5 = 101

6 = 110

7 = 111

Thingcontrol board ใช้ ESP32-WROOM-32 เป็น MCU ที่มี ADC มาให้ 2 ชุด คือ ADC1 และ ADC2 โดยทั้ง 2 ชุดนี้เป็น ADC ที่มีความละเอียด 12 bit (0 ถึง 4095 = 4096 ระดับ) แบบ SAR (Successive Approximation Register) ADCs และสามารถใช้งานได้ทั้งหมด 16 channel โดยใช้ไฟเลี้ยงที่ 3.3 V ซึ่งเท่ากับ Voltage Reference ดังนั้นความละเอียดของแต่ละขั้น $3.3/4096 = 0.8 \text{ mV}$



5.รายละเอียดของ ADC ของ ESP32-WROOM-32

ADC1 จะมีทั้งหมด 8 Channel (0 – 7) และหมายเลข GPIO ดังนี้

ADC1_CH0 – GPIO36

ADC1_CH1 – GPIO37

ADC1_CH2 – GPIO38

ADC1_CH3 – GPIO39

ADC1_CH4 – GPIO32

ADC1_CH5- GPIO33

ADC1_CH6 – GPIO34

ADC1_CH7 – GPIO35

ADC2 จะมีทั้งหมด 10 Channel (0 – 9) และหมายเลข GPIO ดังนี้

ADC2_CH0 – GPIO4

ADC2_CH1 – GPIO0

ADC2_CH2 – GPIO2

ADC2_CH3 – GPIO15

ADC2_CH4 – GPIO13

ADC2_CH5 – GPIO12

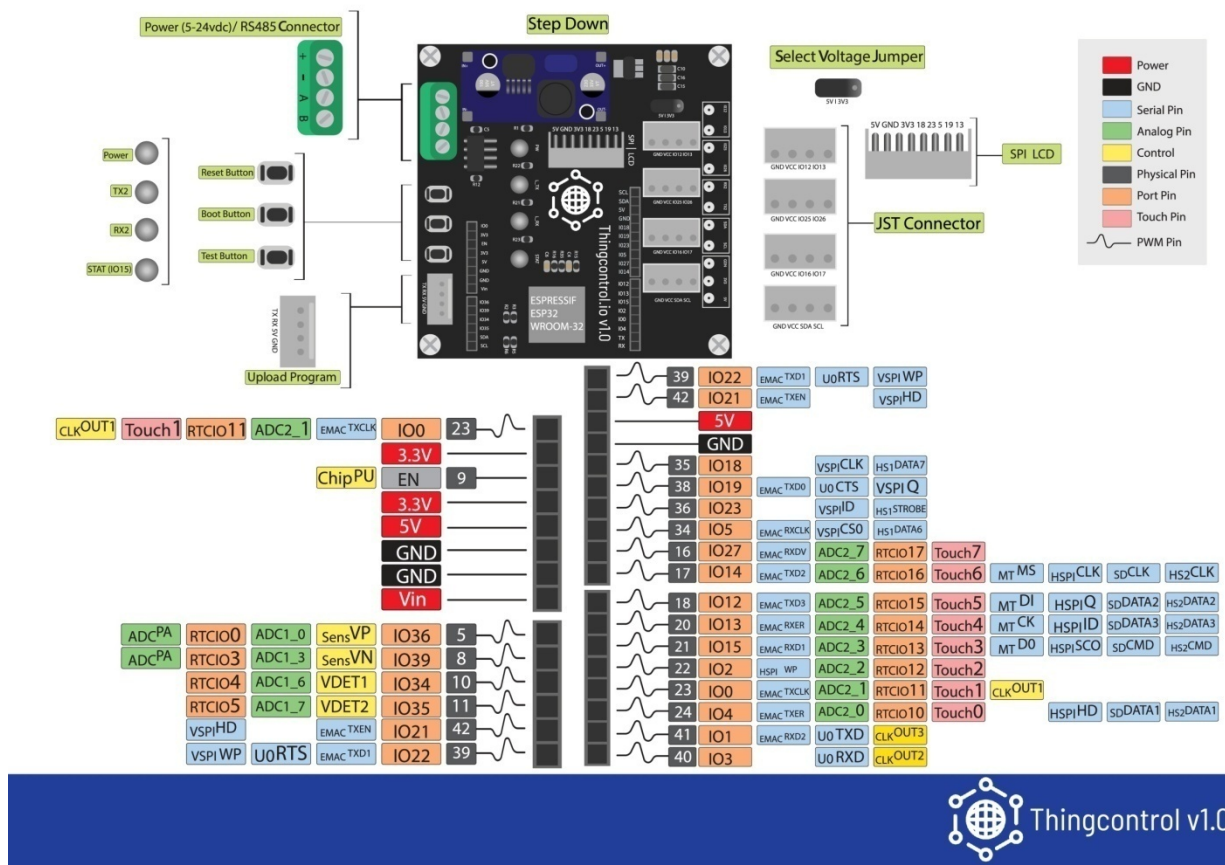
ADC2_CH6 – GPIO14

ADC2_CH7 – GPIO27

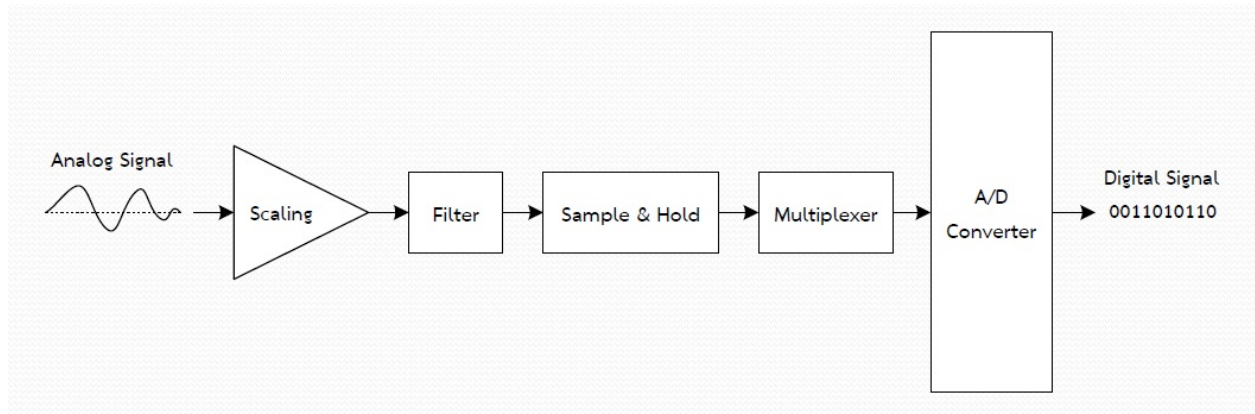
ADC2_CH8 – GPIO25

ADC2_CH9 – GPIO26

6.รายละเอียดของ GPIO pin

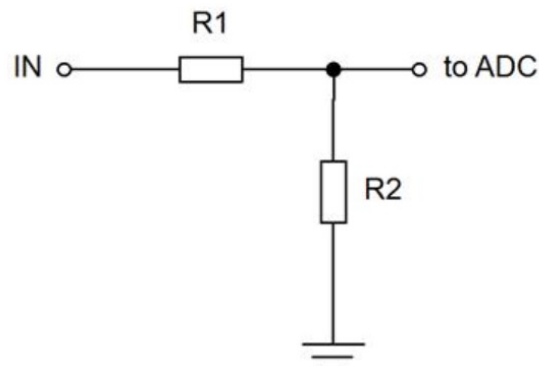


7. ขั้นตอนการแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัลของ thingcontrol board



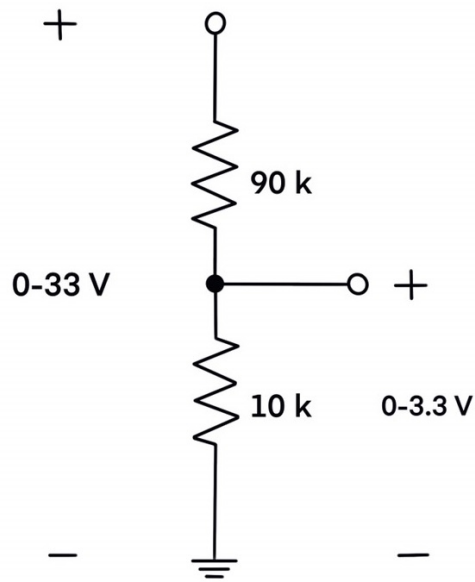
1. ขั้นตอนการทำ Scaling ด้วยการต่อความต้านทานแบ่งแรงดัน

เนื่องจากไมโครคอนโทรลเลอร์ไม่สามารถรับค่าแรงดันไฟฟ้าเกิน 3.3 V ดังนั้นเราจำเป็นต้องใช้วงจรแบ่งแรงดัน (Voltage Dividers) มาช่วย ดังรูป

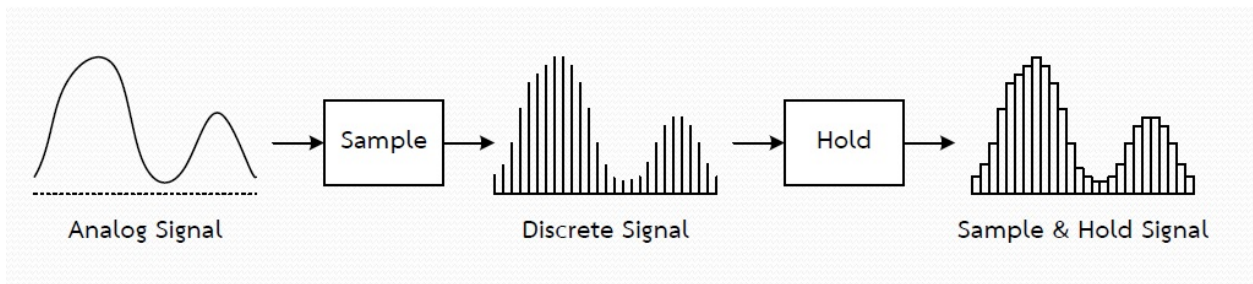


โดยวงจรจะประกอบด้วยตัวความต้านทาน 2 ตัวต่ออนุกรมกัน ซึ่งจะเห็นได้ว่าแรงดัน ADC คือแรงดันที่ตกคร่อม R_2 สามารถได้จากสมการนี้

$$V_{R2} = (R2/R1 + R2) \times V_{IN}$$

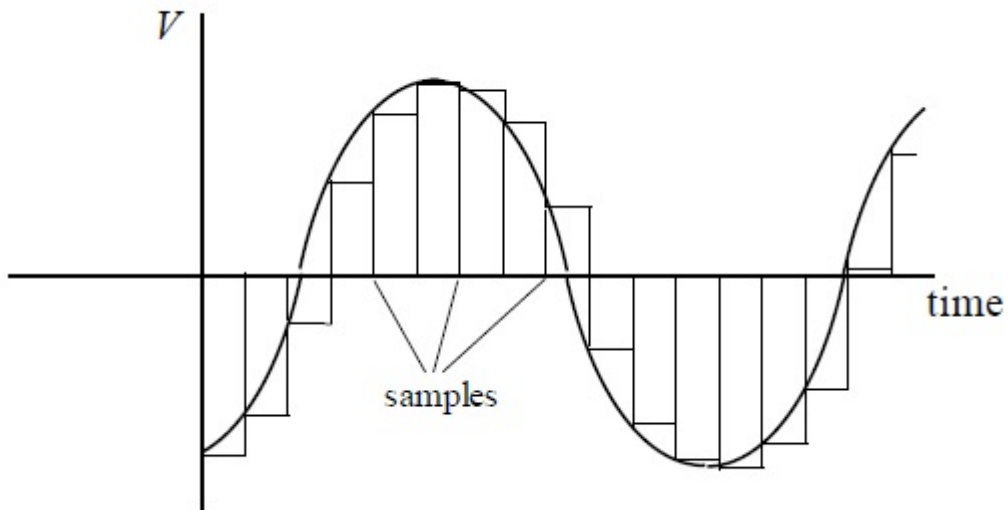


2. ขั้นตอนการ Sample และ Hold



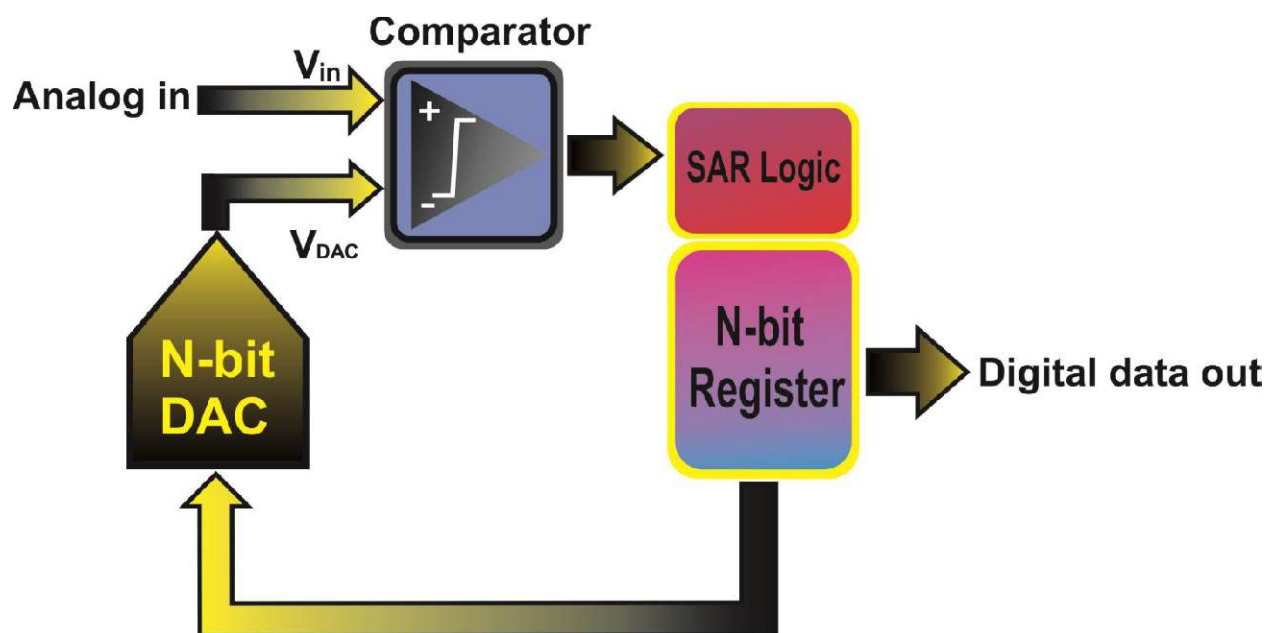
การสุ่มตัวอย่าง (Sampling) คือการแปลงสัญญาณที่มีการเปลี่ยนแปลงต่อเนื่องทางเวลา (continuous signal) ให้อยู่ในรูปไม่ต่อเนื่องทางเวลา (discrete signal) ด้วยการสุ่มเก็บตัวอย่าง (sample) ของสัญญาณนั้น ๆ ในช่วงเวลาที่เท่า ๆ กัน ซึ่งช่วงเวลาดังกล่าวก็คือค่าอัตราสุ่มสัญญาณ (sampling rate) นั่นเอง

การแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิทัล จะเป็นการแบ่งสัญญาณอนาลอกเป็นเป็นขั้นเล็ก ๆ ที่เป็นตัวอย่าง (Sample) ของสัญญาณ ถ้าจำนวนตัวอย่างยิ่งมาก ก็จะทำให้ข้อมูลสัญญาณดิจิทัลมีความแม่นยำสูง



ค่าความถี่ของการสุ่มสัญญาณเรียกว่า Nyquist Frequency การสุ่มตัวอย่าง (sampling theorem) จึงระบุให้ใช้ความถี่ของอัตราสุ่มมากเป็น 2 เท่าของความถี่ของสัญญาณต้นฉบับ เพื่อให้สัญญาณ alias ที่เกิดขึ้นไม่ไปทับซ้อนกับสัญญาณต้นฉบับ

8.N-bit Successive-approximation A/D



9. คำสั่งในอินพุตแบบอนาลอก ADC

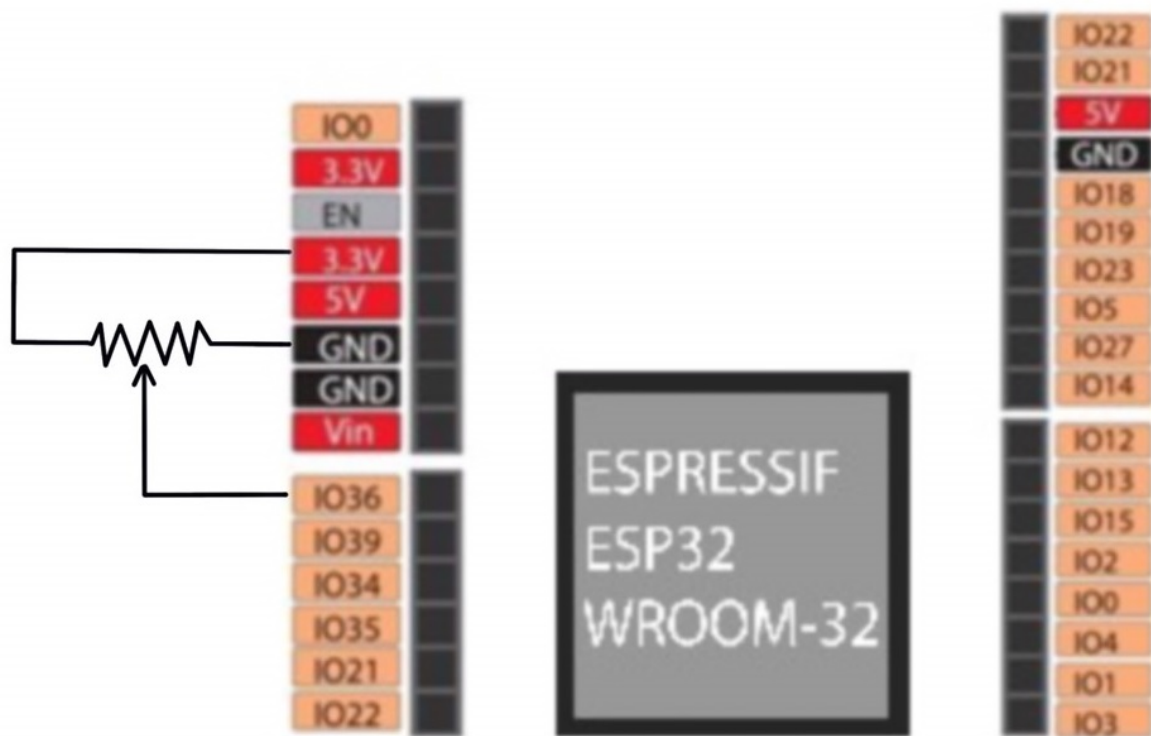
`analogRead(pin)` เป็นการอ่านค่า Analog ของขาพอร์ทที่ระบุไว้

`pin` - หมายเลขของ GPIO ของ thingcontrol board

`delay(value)` เป็นการหน่วงเวลาของโปรแกรมตามเวลาที่กำหนดมีหน่วยเป็นมิลลิวินาที

`value` - มีค่าเป็น, milliseconds (1 second = 1000 milliseconds)

10. รูปแบบการต่อใช้งานของ ADC



11.ตัวอย่างโปรแกรม

```
int sensorPin = 36; // select the input pin for the potentiometer

int sensorValue = 0; // variable to store the value coming from the sensor

float voltageValue = 0;

void setup() {

  Serial.begin(9600);

}

void loop() {

  // read the value from the sensor:

  sensorValue = analogRead(sensorPin);

  Serial.print("ADC Value : ");

  Serial.print(sensorValue);

  voltageValue = (sensorValue* 3.3 ) / (4095);

  Serial.print(" Volatge value : ");

  Serial.println(voltageValue);

  delay(1000);

}
```

คำถามท้ายใบความรู้ที่ 8

1. จงอธิบายหลักการทำงานของตัวแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอล (Analog to Digital Converter)
2. จงอธิบายขั้นตอนการแปลงสัญญาณอนาลอกเป็นสัญญาณดิจิตอลของ thingcontrol board