



Armstrong

School Program 2023-2024

Lesson 6



Armstrong

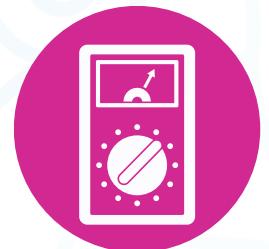
entertainment meets education



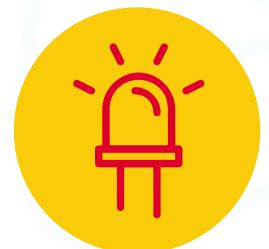
Lesson Content



Revision on Digital signals



Revision on Analog signals



Revision on PWM



Revision on Serial monitor



Revision on Potentiometer



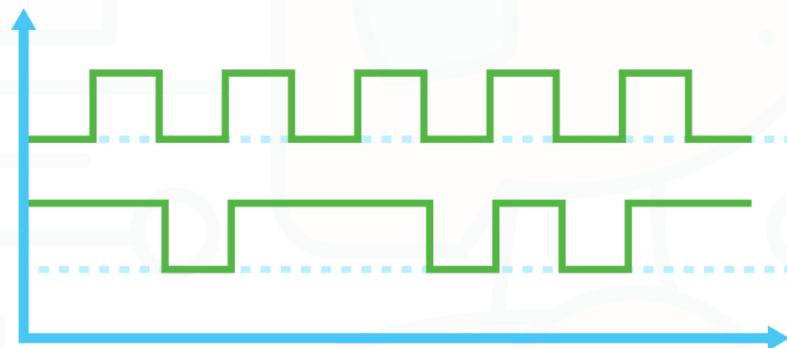
Examples



Remember signals



Digital input :

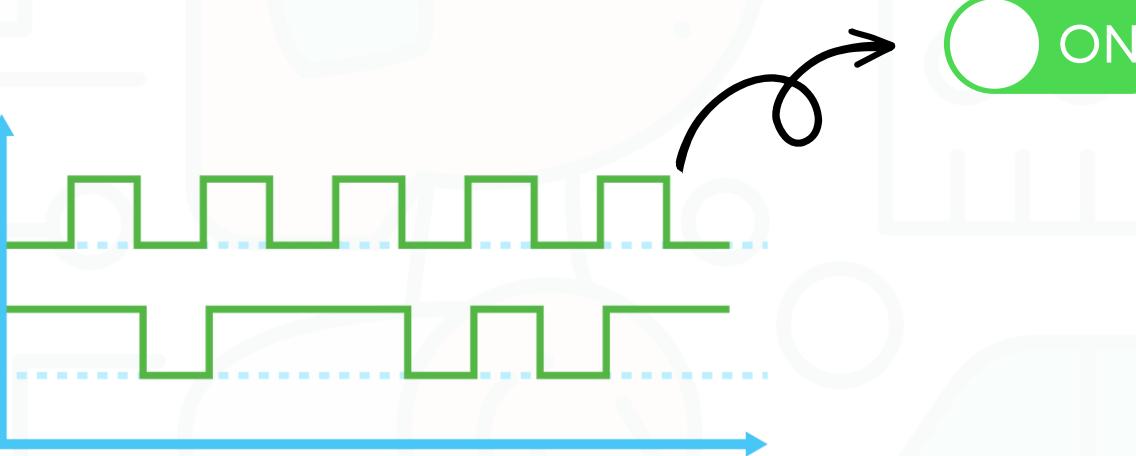


Digital signals are transmitted in the form of 1s and 0s, and it is entirely on or off there is no in-between.



Remember signals

Digital input :



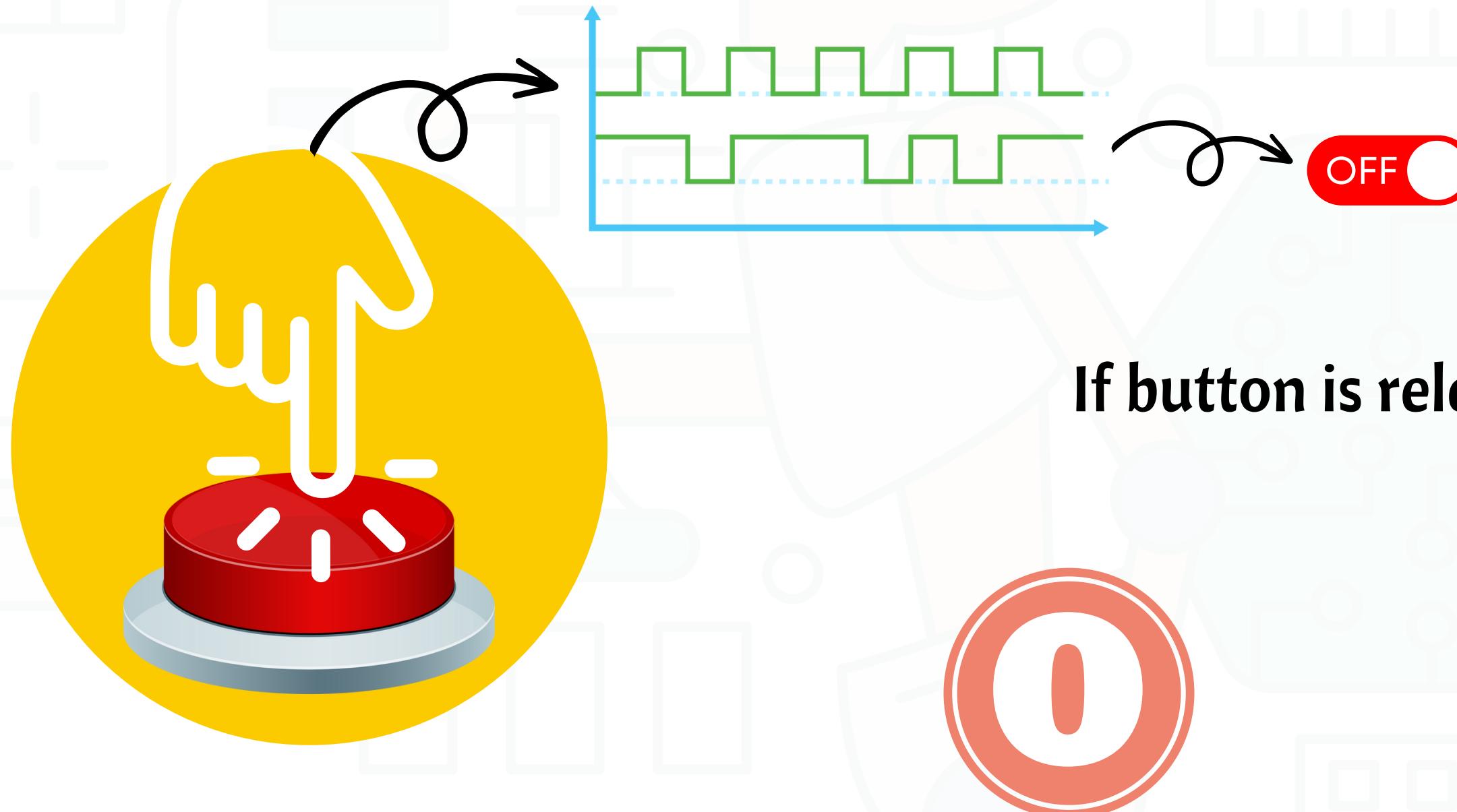
If button is pressed then its "high".

1



Remember signals

Digital input :

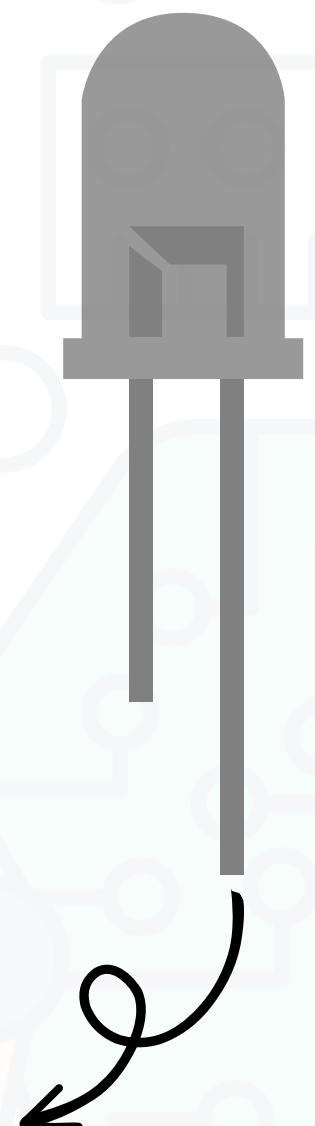


If button is released then its "low".

Remember signals

Digital output :

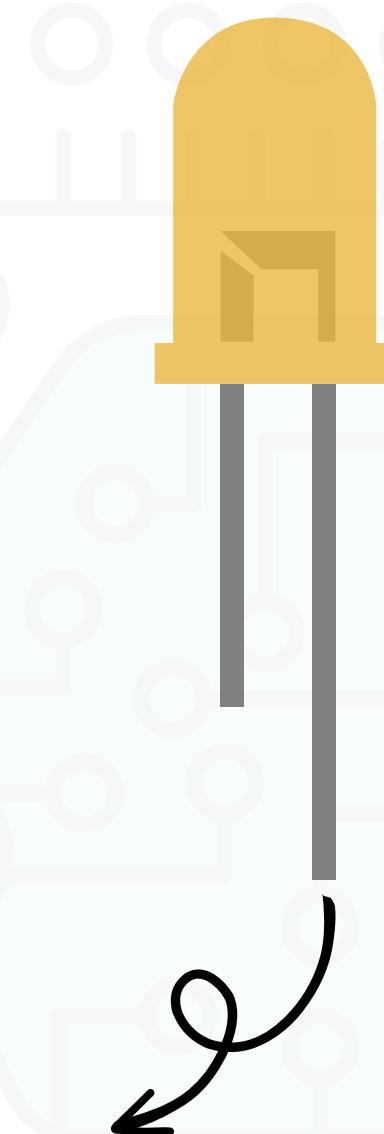
when the led light is off its digital "low"



Remember signals

Digital output:

when the led light is on its digital "high"

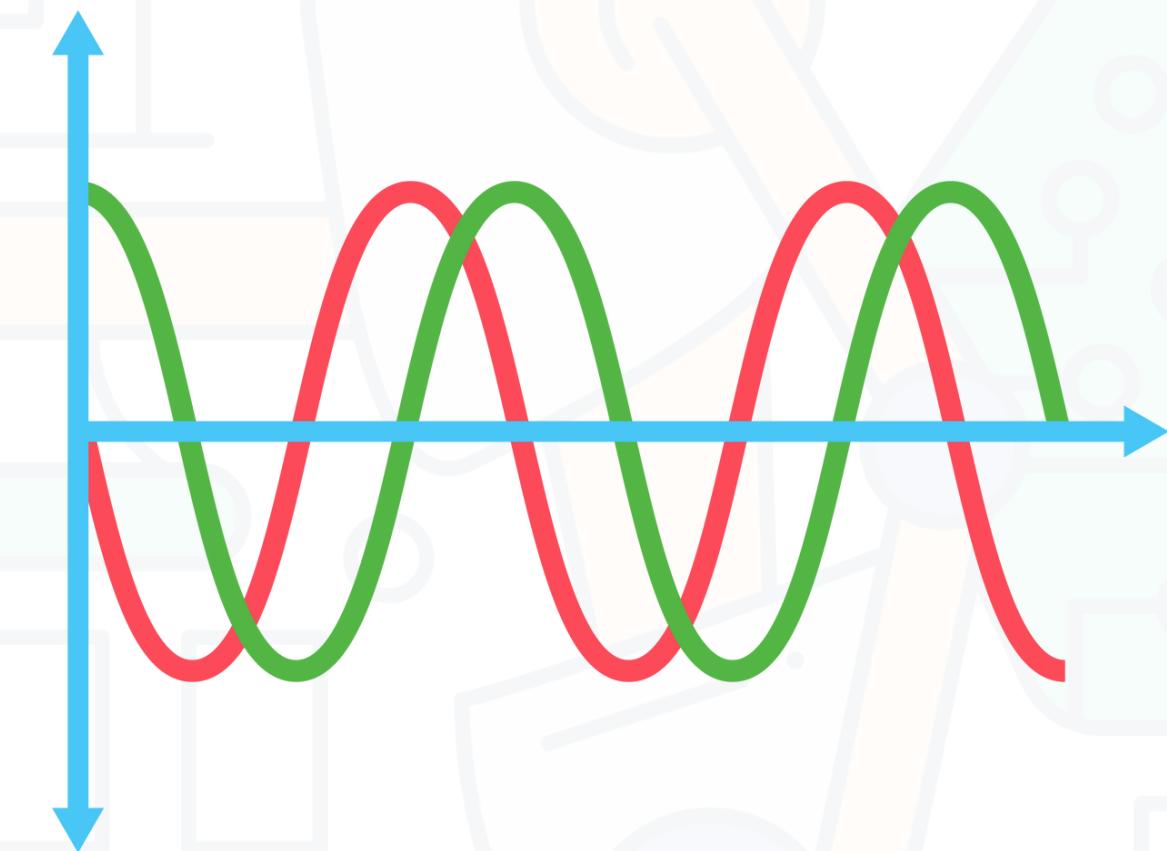




Remember Signals

Digital vs Analog :

Analog signals are waves with a range of values and have gradual changes by time.

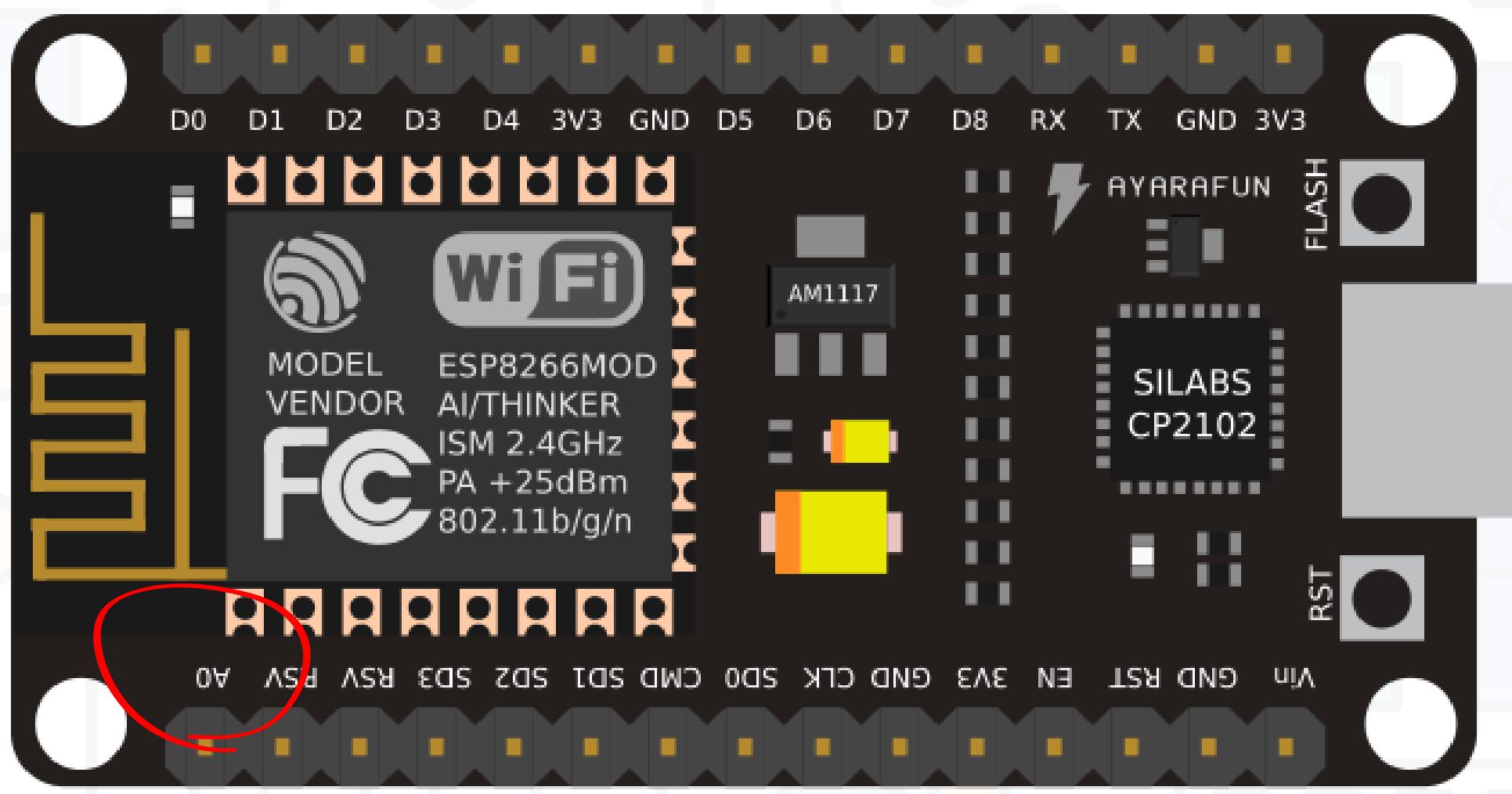




Remember ESP8266

Analog signal:

It has only one analog input pin (A0)

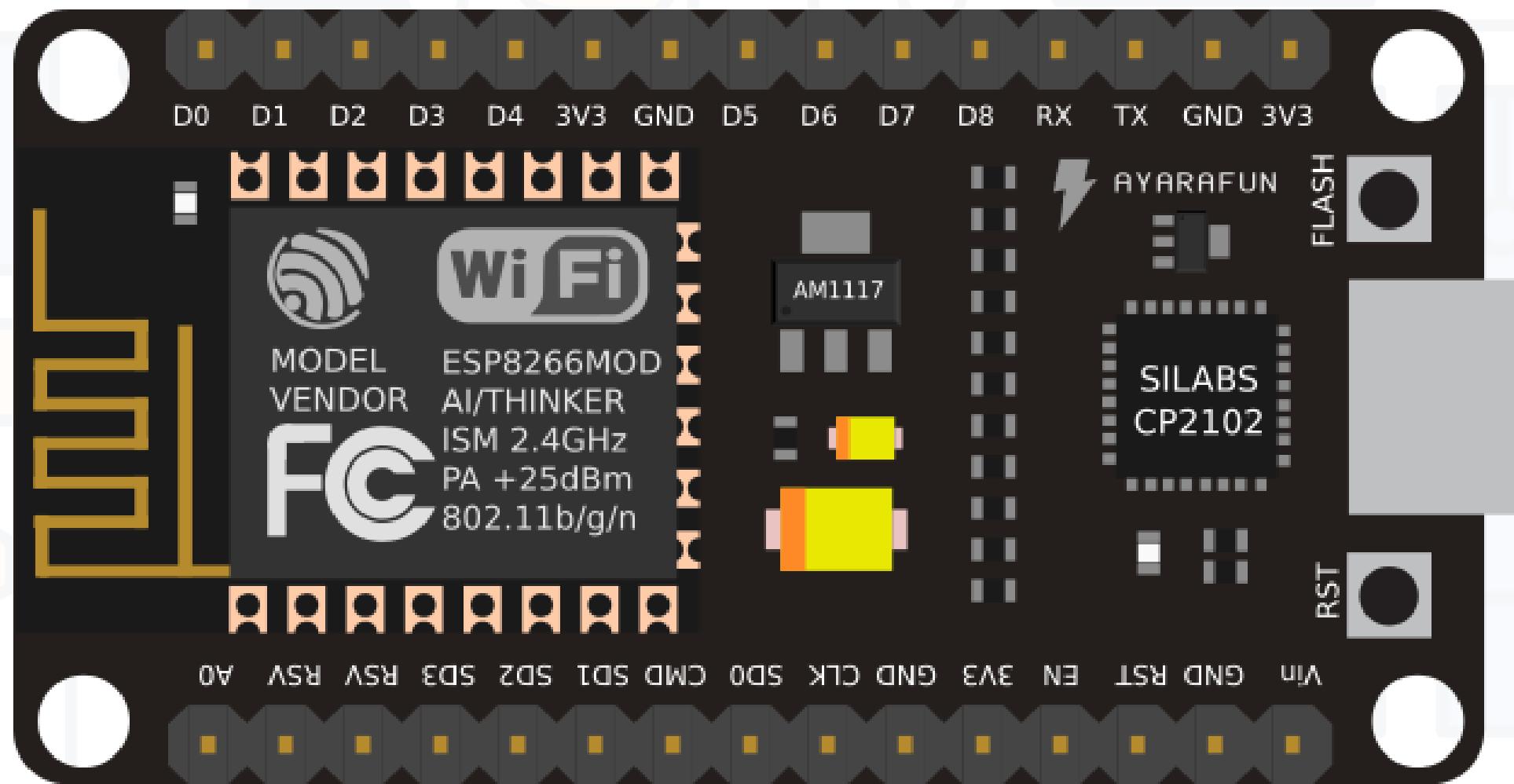




Remember ESP8266

Analog signal:

It has only 9 analog output pins (D0-D8)

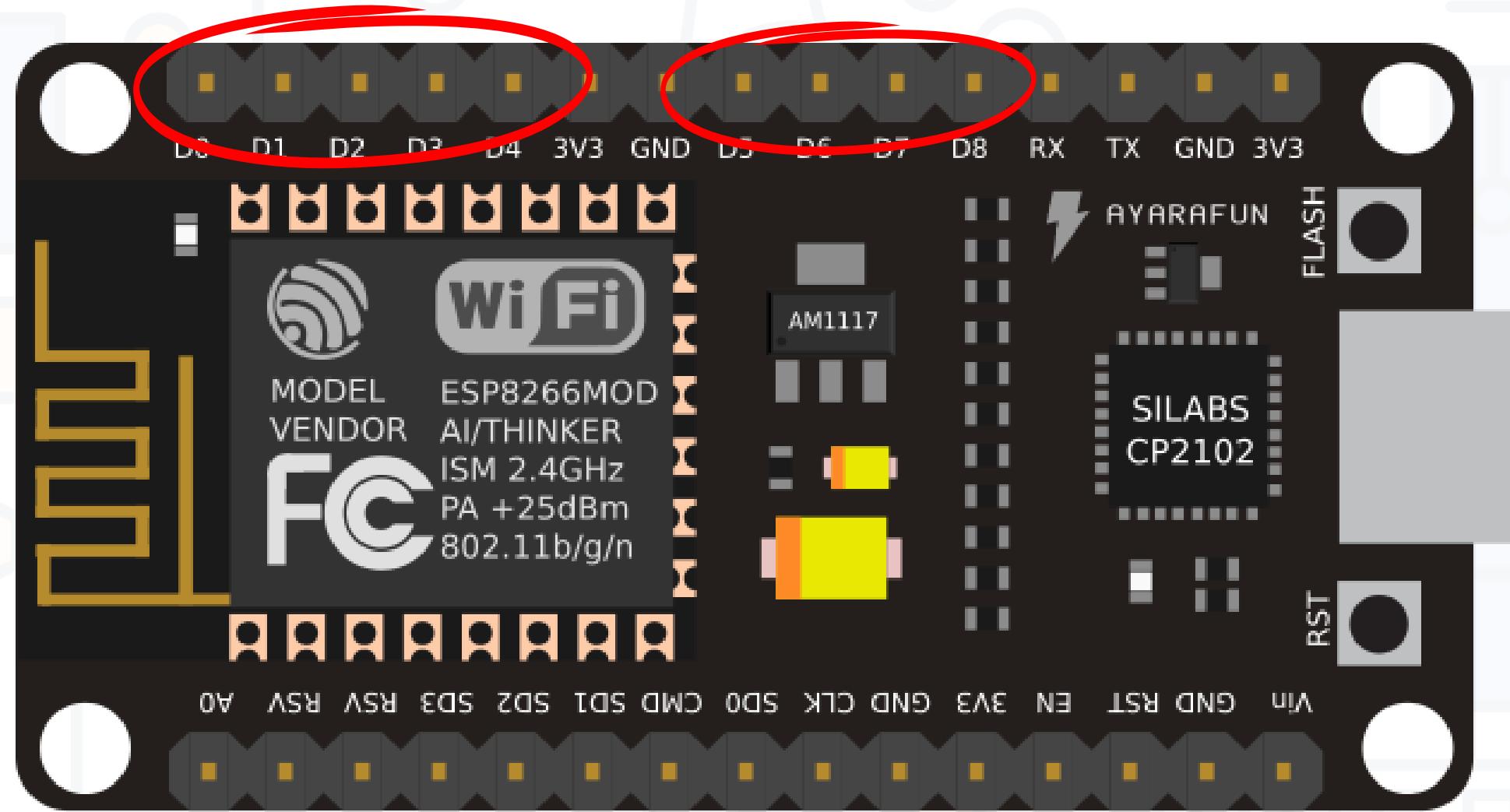




Remember ESP8266

Analog signal:

It has only 9 analog output pins (D0-D8)

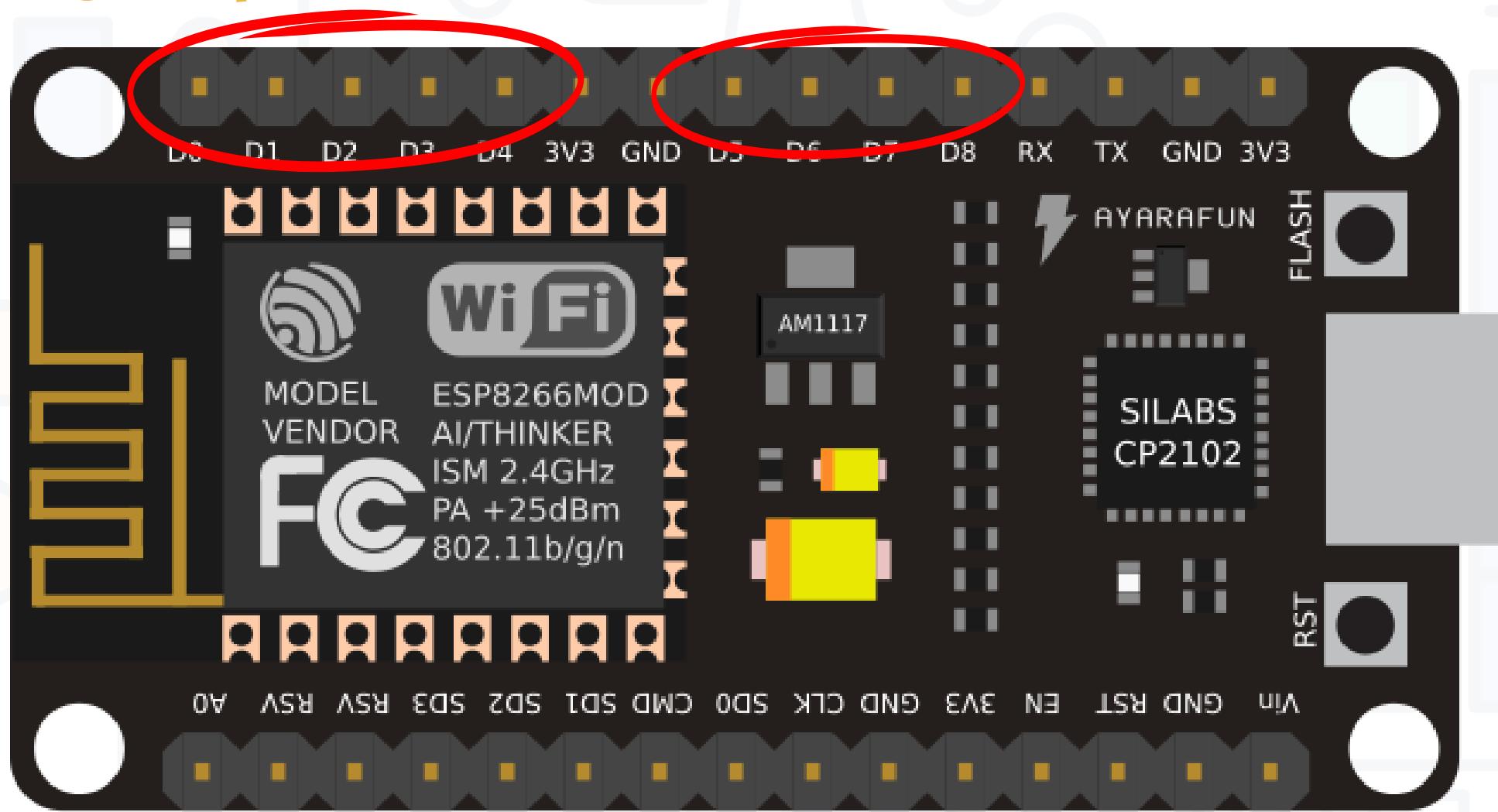




Remember ESP8266

pins:

Those pins can be used as digital input ,digital output
and also analog output





Remember

PWM

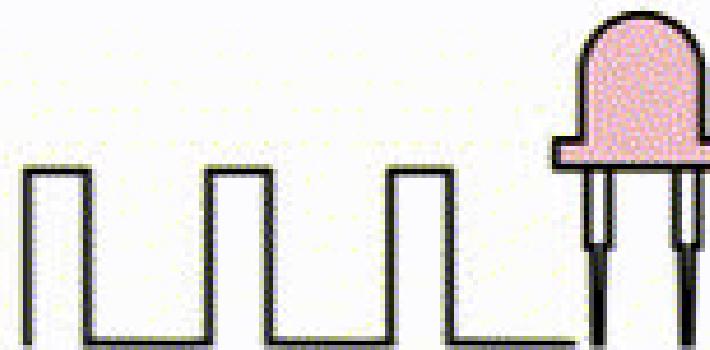


What does PWM stands for?

Pulse width modulation.

What is PWM?

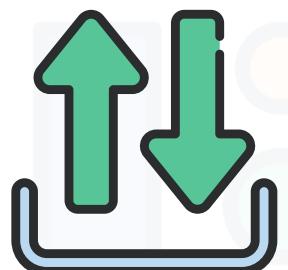
PWM is a way to control the amount of electricity that goes to a device by turning it on and off very quickly, like a water faucet.



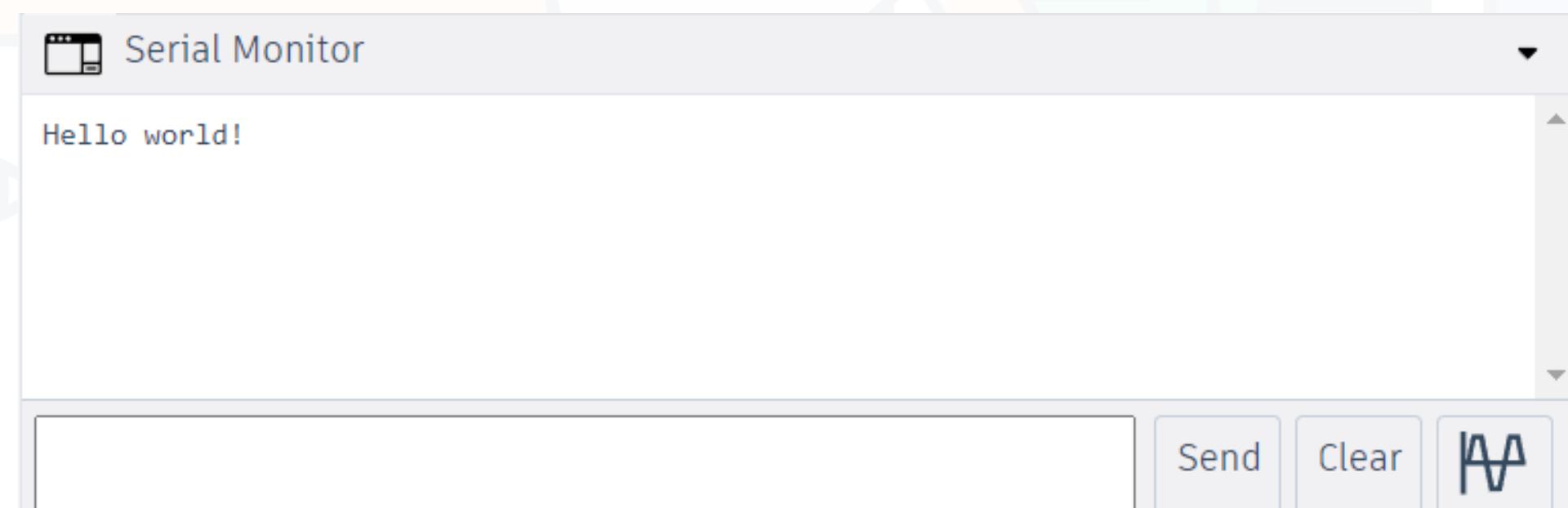
Remember Serial monitor

what is a serial monitor?

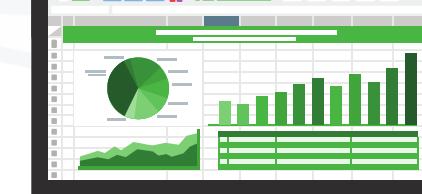
Its the partition on the screen where you display and print outputs or send inputs



For example we can print a simple message:

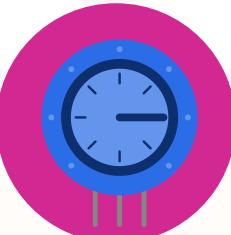


```
Serial Monitor
Hello world!
Send Clear
```





Remember potentiometer



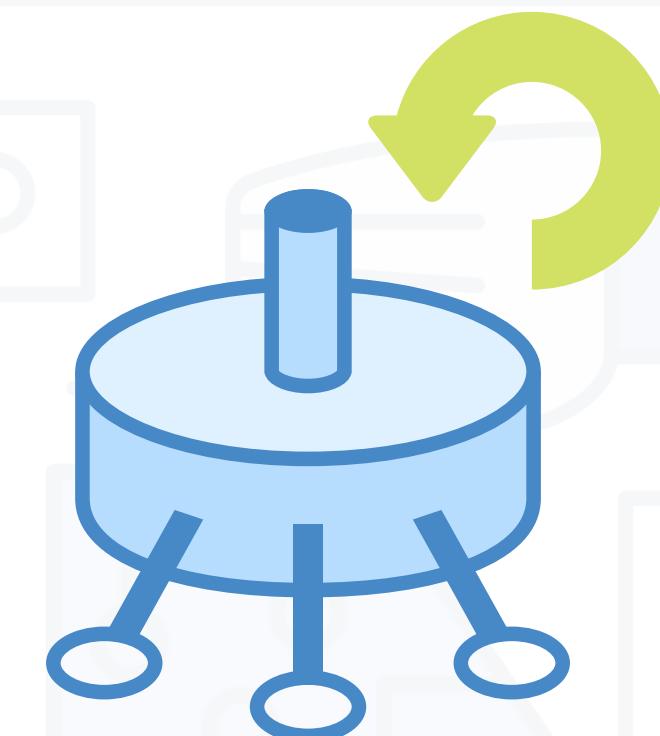
what is the potentiometer?

Its a variable resistance that can be changed manually, it has no polarity and is a good example for analog input

How does it work?

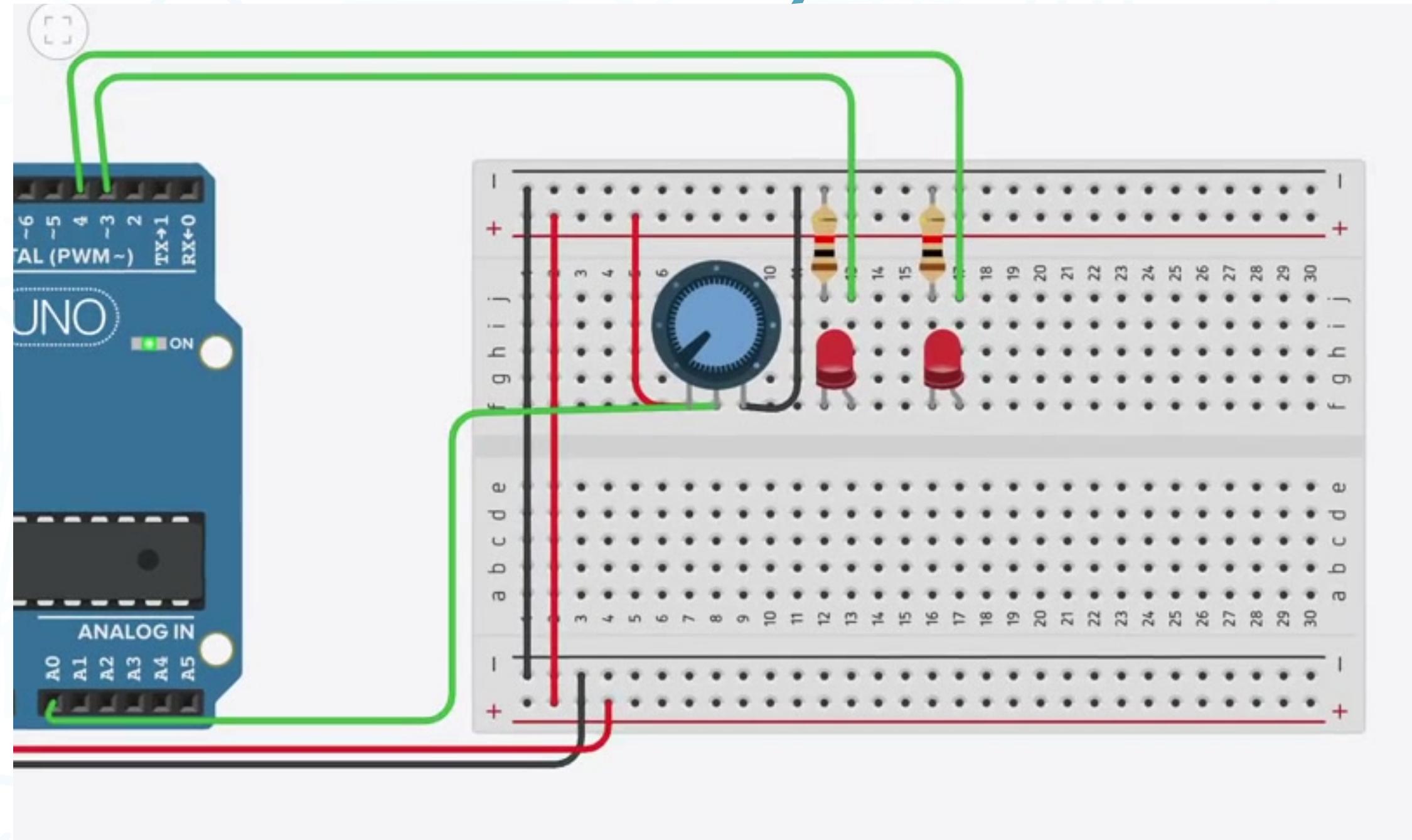
It has three terminals, the first and third resistances are fixed and the middle one is variable

when the knob is turned the resistance between the middle terminal and the other two terminals changes it decreases in the direction of rotation

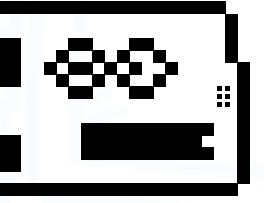


Example(control LEDs intensity)

Example: control the intensity LEDs via Potentiometer



Let's try it on Arduino IDE



Experiment: control LEDs intesity

Try to partition the code yourself



Example(control LEDs intensity)

Initialize our three variables

```
1 int led1pin = D3; // Pin for LED 1:  
2 int led2pin = D4; // Pin for LED 2  
3 int potPin = A0; // Pin for the potentiometer  
4  
5  
6 void setup()  
7 {  
8     pinMode(led1pin, OUTPUT);  
9     pinMode(led2pin, OUTPUT);  
10    // No need to set up the potentiometer pin; it is an input by default  
11 }  
12
```

non need to set up
the potentiometer

Setting up the pins

Example(LEDs and Potentiometer)

```
int potValue=analogRead(potPin);
```

reads the analog value from
the potentiometer and store it
in a variable caled potValue

Note:

There is difference between declaring variables inside and outside the void loop

```
int led1pin = D3; // Pin for LED 1:  
int led2pin = D4; // Pin for LED 2  
int potPin = A0; // Pin for the potentiometer
```

- Called **global variables**
- they are accessible from any part of the code, including inside functions.
- They can be accessed, modified, and used throughout the entire program.

```
void loop()  
{  
    // Read the potentiometer value (0 to 1023)  
    int potValue = analogRead(potPin);
```

- Called **local variables**
- They are **only accessible and valid within the function in which they are declared.**
- They cannot be accessed or modified from outside the function.



Map()

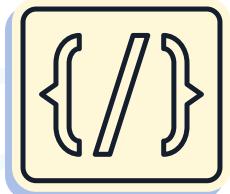
Definition

Re-maps a number from one range to another. That is, a value of `fromLow` would get mapped to `toLow`, a value of `fromHigh` to `toHigh`, values in-between to values in-between, etc.



Map()

Parameters



`map(value, fromLow, fromHigh, toLow, toHigh)`

value: the number to map.

fromLow: the lower bound of the value's current range.

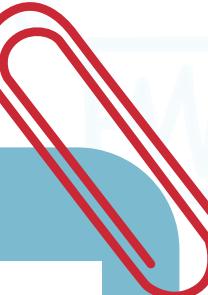
fromHigh: the upper bound of the value's current range.

toLow: the lower bound of the value's target range.

toHigh: the upper bound of the value's target range.

what is map ()?

```
int brightness = map(potValue, 0, 1023, 0, 255);
```



maps the potentiometer value to a brightness value ranging from 0 to 255. This brightness value will be used to control the intensity of the LEDs.

Example(control LEDs intensity)

```
analogWrite(led1pin, brightness);  
analogWrite(led2pin, brightness);
```

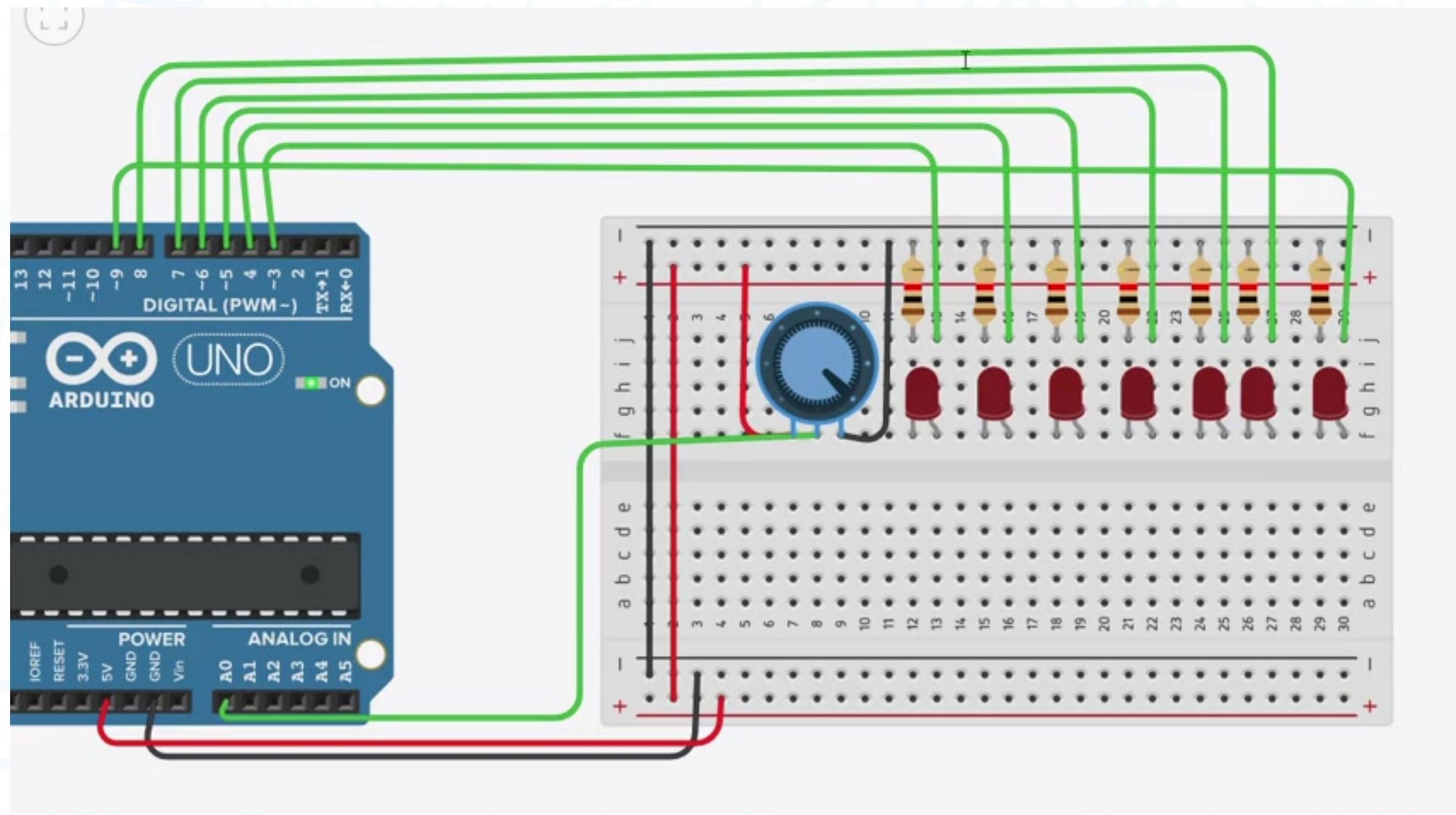
use PWM to control the brightness of
both LED 1 and LED 2. T

Example(control LEDs intensity)

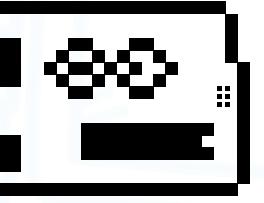
```
1 int led1pin = D3; // Pin for LED 1:  
2 int led2pin = D4; // Pin for LED 2  
3 int potPin =A0; // Pin for the potentiometer  
4  
5  
6 void setup()  
7 {  
8     pinMode(led1pin, OUTPUT);  
9     pinMode(led2pin, OUTPUT);  
10    // No need to set up the potentiometer pin; it is an input by default  
11 }  
12  
13  
14 void loop()  
15 {  
16     // Read the potentiometer value (0 to 1023)  
17     int potValue=analogRead(potPin);  
18     //Map the potentiometer value to the LED brightness range (0 to 255)  
19     int brightness = map(potValue, 0, 1023, 0, 255);  
20     // Get the LED brightness using the PWM technique  
21     analogWrite(led1pin, brightness);  
22     analogWrite(led2pin, brightness);  
23 }
```

Example(control multiple LEDs)

Example: control the multiple LEDs via Potentiometer



Let's try it on Arduino IDE



Experiment: control multiple LEDs using
potentiometer

Try to partition the code yourself



Example(control multiple LEDs)

```
int potPin = A0;  
int led1Pin= D0;  
int led2Pin= D3;  
int led3Pin= D4;  
int led4Pin= D5;  
int led5Pin= D6;  
int led6Pin= D7;  
int led7Pin= D8;
```

Initialize our
variables

Example(control multiple LEDs)

Set LED pins as outputs

```
void setup()
{
    pinMode(potPin, INPUT);
    pinMode(led1Pin, OUTPUT);
    pinMode(led2Pin, OUTPUT);
    pinMode(led3Pin, OUTPUT);
    pinMode(led4Pin, OUTPUT);
    pinMode(led5Pin, OUTPUT);
    pinMode(led6Pin, OUTPUT);
    pinMode(led7Pin, OUTPUT);
}
```

Set potentiometer pin as input

Example(control multiple LEDs)

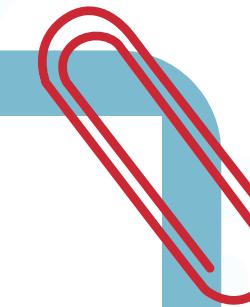
```
int potValue=analogRead(potPin);
```

reads the analog value from
the potentiometer and store it
in a variable caled potValue

Example(control multiple LEDs)

Check potValue range and control LEDs accordingly

```
if (potValue > 0 && potValue <= 128) {  
    digitalWrite(led1Pin,LOW);  
    digitalWrite(led2Pin,LOW);  
    digitalWrite(led3Pin,LOW);  
    digitalWrite(led4Pin,LOW);  
    digitalWrite(led5Pin,LOW);  
    digitalWrite(led6Pin,LOW);  
    digitalWrite(led7Pin,LOW);  
} else if (potValue > 128 && potValue <= 256) {  
    digitalWrite(led1Pin,HIGH);
```



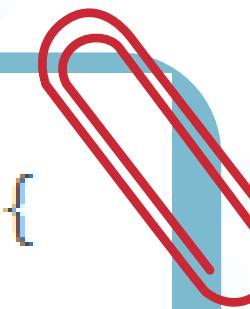
Turn off all LEDs

turn on LED1

Example(control multiple LEDs)

repeat for all the LEDs

```
digitalWrite(led2Pin,HIGH);  
} else if (potValue > 384 && potValue <= 512) {  
    digitalWrite(led3Pin,HIGH);  
} else if (potValue > 512 && potValue <= 640) {  
    digitalWrite(led4Pin,HIGH);  
} else if (potValue > 640 && potValue <= 768) {  
    digitalWrite(led5Pin,HIGH);  
} else if (potValue > 768 && potValue <= 896) {  
    digitalWrite(led6Pin,HIGH);  
} else if (potValue > 896 && potValue <= 1024) {  
    digitalWrite(led7Pin,HIGH);  
}
```



Example(control multiple LEDs)

```
int potPin = A0;      // Pin for the potentiometer
int led1Pin= D0;
int led2Pin= D3;
int led3Pin= D4;
int led4Pin= D5;
int led5Pin= D6;
int led6Pin= D7;
int led7Pin= D8;

void setup()
{
    pinMode(potPin, INPUT);
    pinMode(led1Pin, OUTPUT);
    pinMode(led2Pin, OUTPUT);
    pinMode(led3Pin, OUTPUT);
    pinMode(led4Pin, OUTPUT);
    pinMode(led5Pin, OUTPUT);
    pinMode(led6Pin, OUTPUT);
    pinMode(led7Pin, OUTPUT);
}
```



Example(control multiple LEDs)

```
void loop()
{
    int potValue = analogRead(potPin);

    if (potValue > 0 && potValue <= 128) {
        digitalWrite(led1Pin,LOW);
        digitalWrite(led2Pin,LOW);
        digitalWrite(led3Pin,LOW);
        digitalWrite(led4Pin,LOW);
        digitalWrite(led5Pin,LOW);
        digitalWrite(led6Pin,LOW);
        digitalWrite(led7Pin,LOW);
    } else if (potValue > 128 && potValue <= 256) {
        digitalWrite(led1Pin,HIGH);
    } else if (potValue > 256 && potValue <= 384) {
        digitalWrite(led2Pin,HIGH);
    } else if (potValue > 384 && potValue <= 512) {
        digitalWrite(led3Pin,HIGH);
    } else if (potValue > 512 && potValue <= 640) {
        digitalWrite(led4Pin,HIGH);
    } else if (potValue > 640 && potValue <= 768) {
        digitalWrite(led5Pin,HIGH);
    } else if (potValue > 768 && potValue <= 896) {
        digitalWrite(led6Pin,HIGH);
    } else if (potValue > 896 && potValue <= 1024) {
        digitalWrite(led7Pin,HIGH);
    }
}
```

