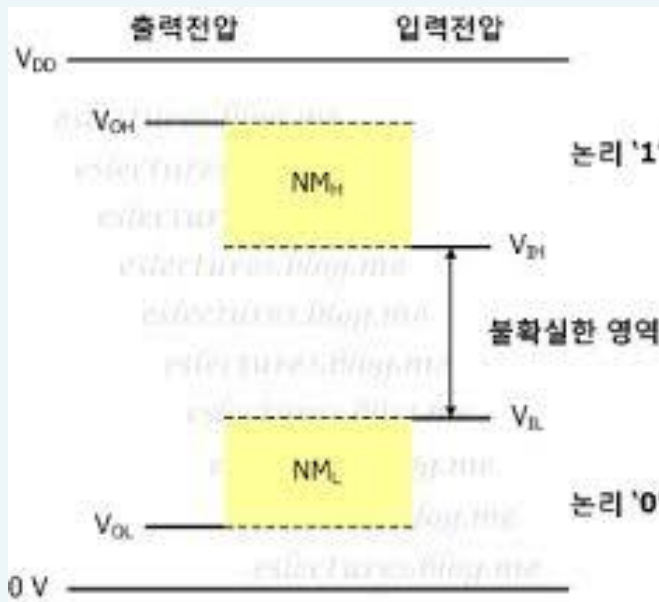
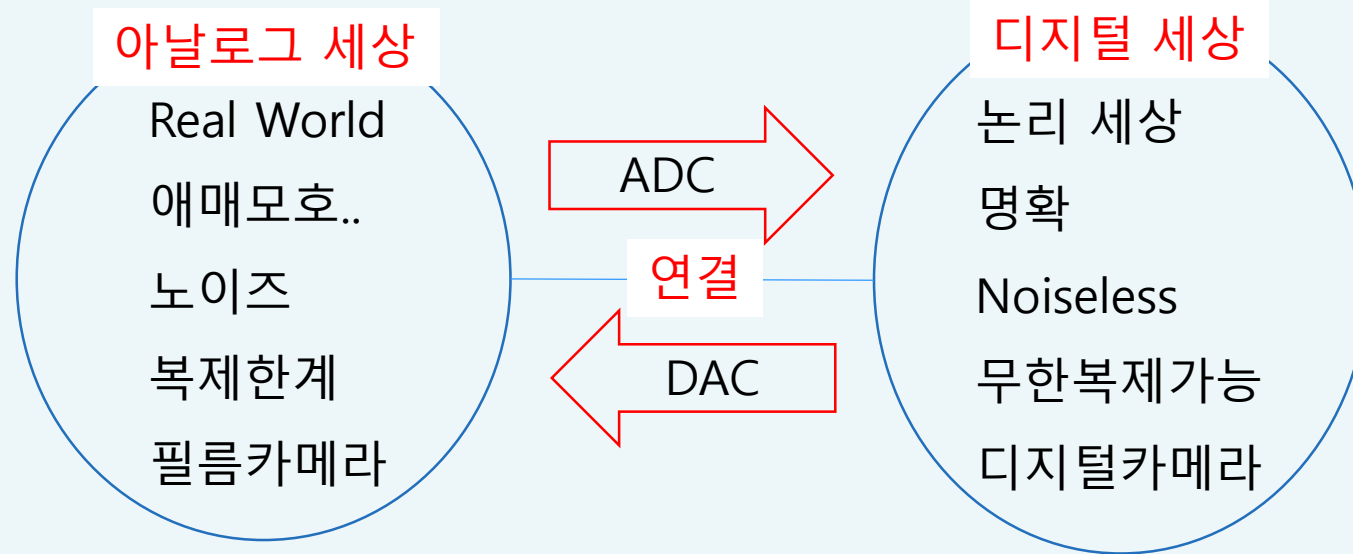


Internet of Things class 3

ESP32 GPIO, Digital Input / Output

디지털의 이해



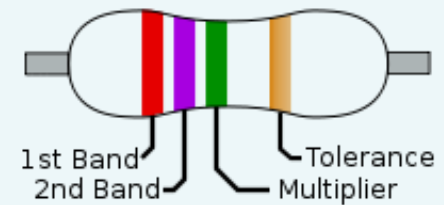
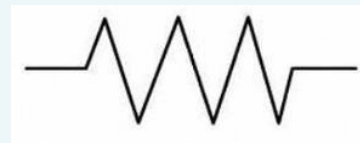
* 입력신호변환: 정보화@3.3v

아날로그	디지털	
전기신호	논리값	코드값
2.3~3.3V	1 (HIGH)	0 이외의 값
0~1V	0 (LOW)	0

0 ~ $V_{cc} * 1/3$: LOW, $V_{cc} * 2/3$ ~ V_{cc} : HIGH

Resistors

- 저항: 전기의 흐름을 제한하는 정도 (단위: 오옴, Ω)

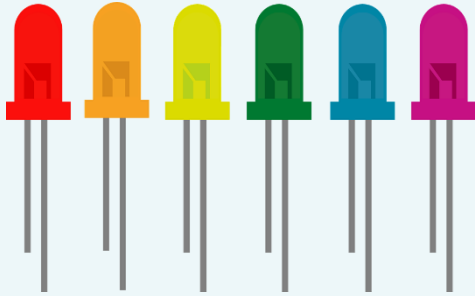


<div> <div>0 1 2 3 4 5 6 7 8 9</div> <div> <div>0 Black</div> <div>1 Brown</div> <div>2 Red</div> <div>3 Orange</div> <div>4 Yellow</div> <div>5 Green</div> <div>6 Blue</div> <div>7 Purple</div> <div>8 Grey</div> <div>9 White</div> </div> <div> <div>$\pm 1\%$ Brown</div> <div>$\pm 2\%$ Red</div> <div>$\pm 5\%$ Gold</div> <div>$\pm 10\%$ Silver</div> </div> </div>	<div> <div>$\pm 1\%$</div> <div>$\pm 2\%$</div> <div>$\pm 5\%$</div> <div>$\pm 10\%$</div> </div> <div> </div> <div> <div>0 $\times 1$</div> <div>1 1 $\times 10$</div> <div>2 2 $\times 100$</div> <div>3 3 $\times 1000$</div> <div>4 4 $\times 10000$</div> <div>5 5 $\times 100000$</div> <div>6 6 $\times 1000000$</div> <div>7 7 $\div 10$</div> <div>8 8 $\div 100$</div> <div>9 9</div> </div>	<div> <div>$\pm 1\%$</div> <div>$\pm 2\%$</div> <div>$\pm 5\%$</div> <div>$\pm 10\%$</div> </div> <div> </div> <div> <div>0 0 $\times 1$</div> <div>1 1 1 $\times 10$</div> <div>2 2 2 $\times 100$</div> <div>3 3 3 $\times 1000$</div> <div>4 4 4 $\times 10000$</div> <div>5 5 5 $\div 10$</div> <div>6 6 6 $\div 100$</div> <div>7 7 7</div> <div>8 8 8</div> <div>9 9 9</div> </div>	<div> <div>$\pm 1\%$</div> <div>$\pm 2\%$</div> <div>$\pm 5\%$</div> <div>$\pm 10\%$</div> </div> <div> <div>100 50</div> <div>25 15</div> <div>10 5</div> <div>1</div> </div> <div> </div> <div> <div>0 0 $\times 1$</div> <div>1 1 1 $\times 10$</div> <div>2 2 2 $\times 100$</div> <div>3 3 3 $\times 1000$</div> <div>4 4 4 $\times 10000$</div> <div>5 5 5 $\div 10$</div> <div>6 6 6 $\div 100$</div> <div>7 7 7</div> <div>8 8 8</div> <div>9 9 9</div> </div>
Color Codes	4 Band Resistors	5 Band Resistors	6 Band Resistors

출처: <http://www.diyaudioandvideo.com/Electronics/ResistorColorCodes/>

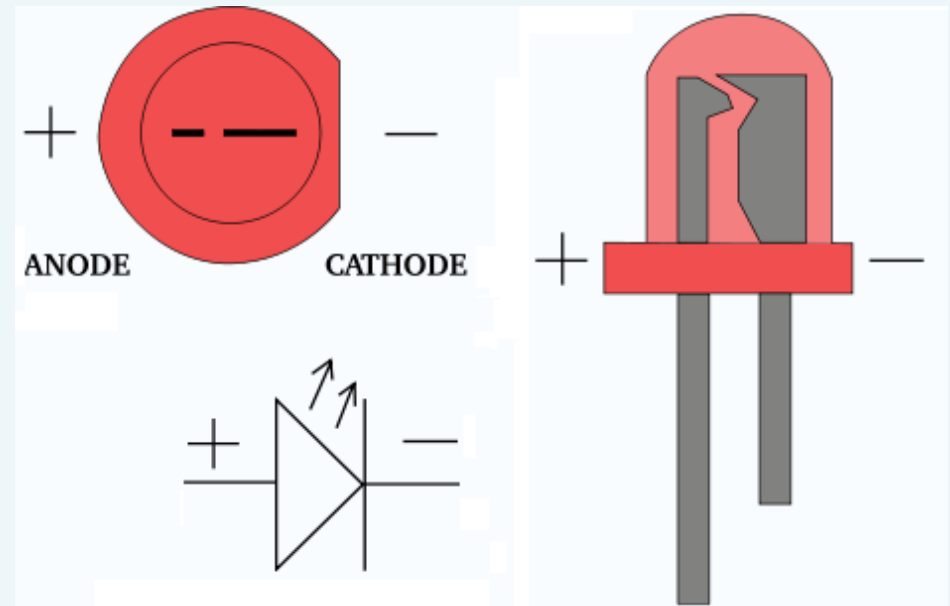
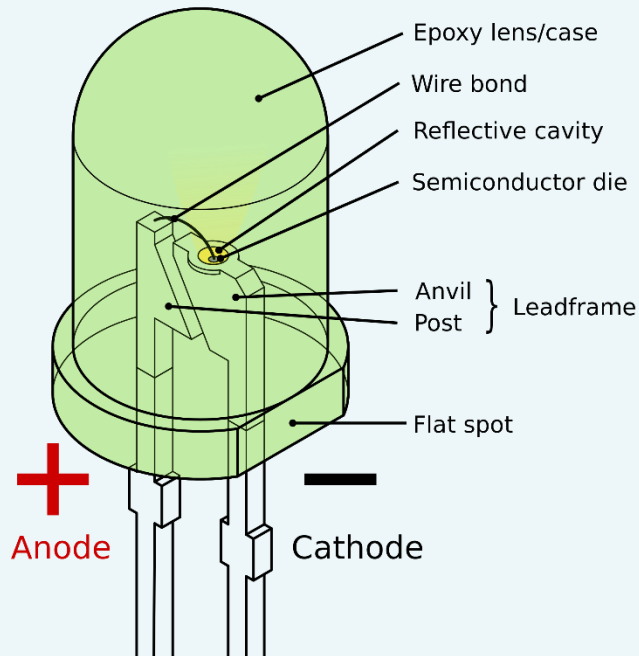
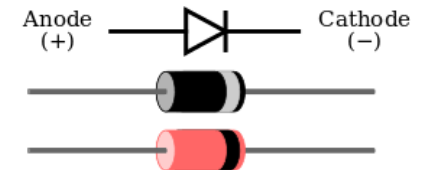
LED (Light-emitting diode)

- 발광다이오드: 순방향 전압에 빛을 내는 반도체 소자




**** 다이오드:** 한쪽 방향으로 전류가 흐르도록 제어하는 반도체




ko.wikipedia.org



LED (Light-emitting diode)



LED - Basic Red 5mm

COM-09590 ROHS   

Description: LEDs - those blinky things. A must have for power indication, pin status, opto-electronic sensors, and fun blinky displays.



This is a very basic 5mm LED with a red lens. It has a typical forward voltage of 2.0V and a rated forward current of 20mA.

Features:

- 1.8-2.2VDC forward drop
- Max current: 20mA
- Suggested using current: 16-18mA
- Luminous Intensity: 150-200mcd

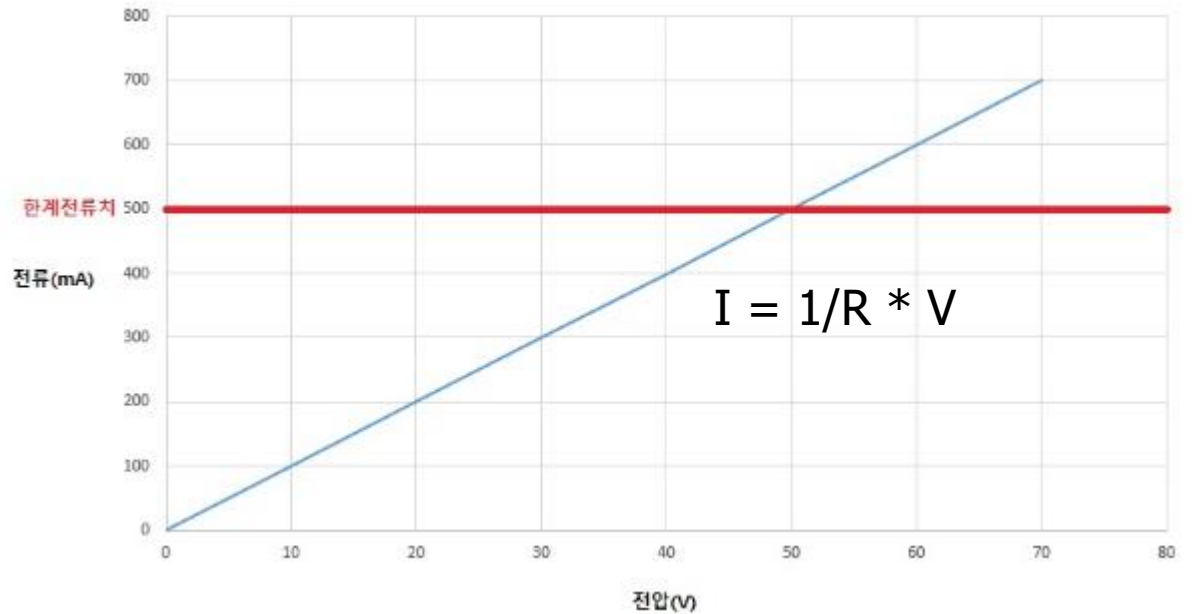
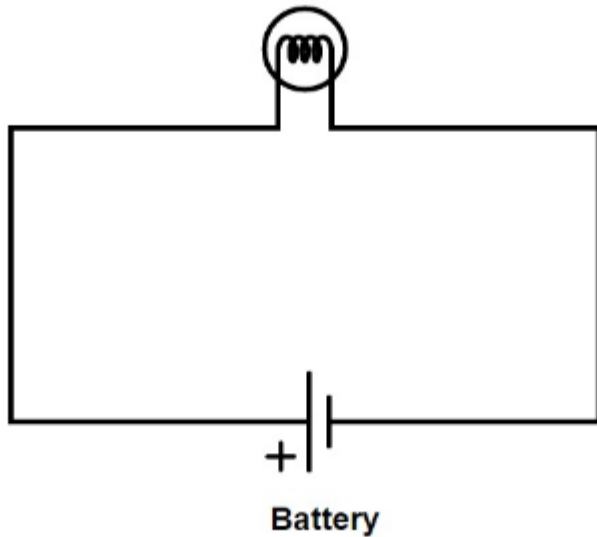
Documents:

- [Datasheet](#)
- [LED Tutorial](#)



ITEMS	Symbol	Absolute Maximum Rating	Unit
Forward Current	I_F	20	mA
Peak Forward Current	I_{FP}	30	mA
Suggestion Using Current	I_{su}	16-18	mA
Reverse Voltage ($V_R=5V$)	I_R	10	uA
Power Dissipation	P_D	105	mW
Operation Temperature	T_{OPR}	-40 ~ 85	°C
Storage Temperature	T_{STG}	-40 ~ 100	°C
Lead Soldering Temperature	T_{SOL}	Max. 260°C for 3 Sec. Max. (3mm from the base of the epoxy bulb)	

옴의 법칙 : $V = I * R$



0) 전지에 은박지를 연결할 때 왜 불이 붙을까?

1) 전구의 저항 = 100Ω

Battery의 전압 = $10V$ 이면

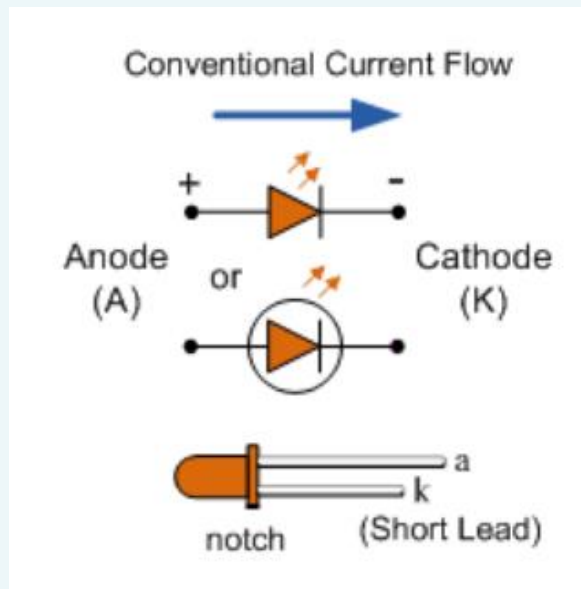
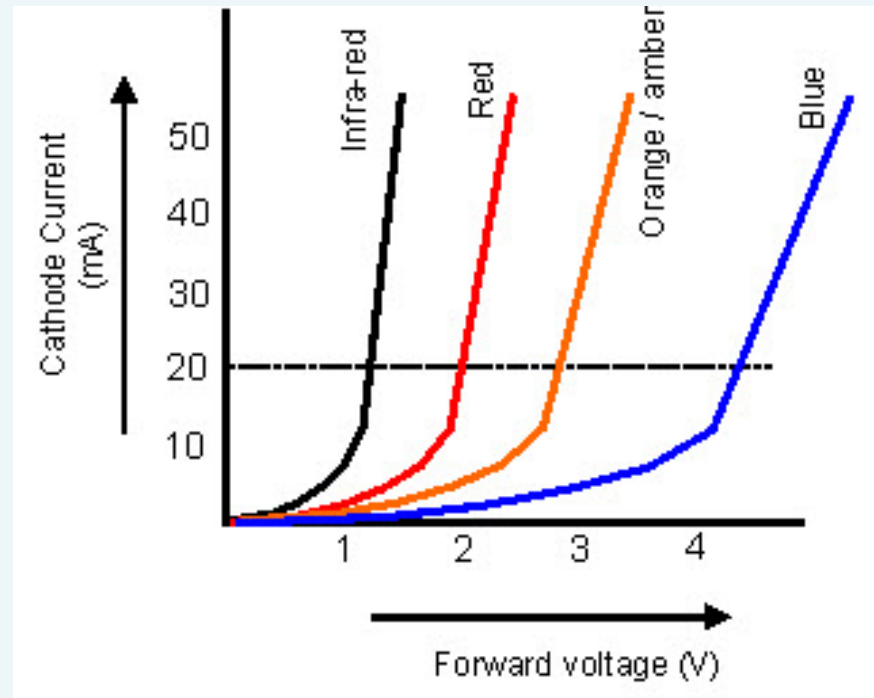
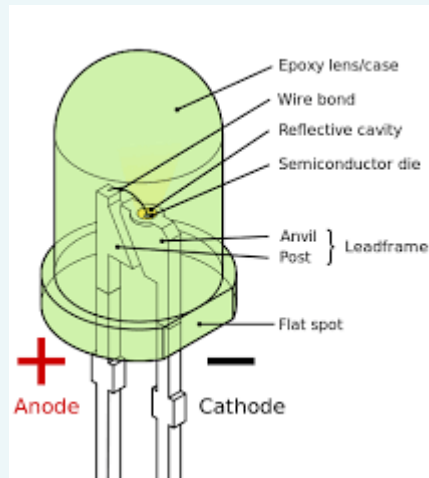
전류는 얼마인가? ($I = V / R$)

2) 전구의 한계 전류값 = $500mA$ 라면

허용할 수 있는 최대 전압은? ($V = I * R$)



옴의 법칙 : LED (발광 다이오드)



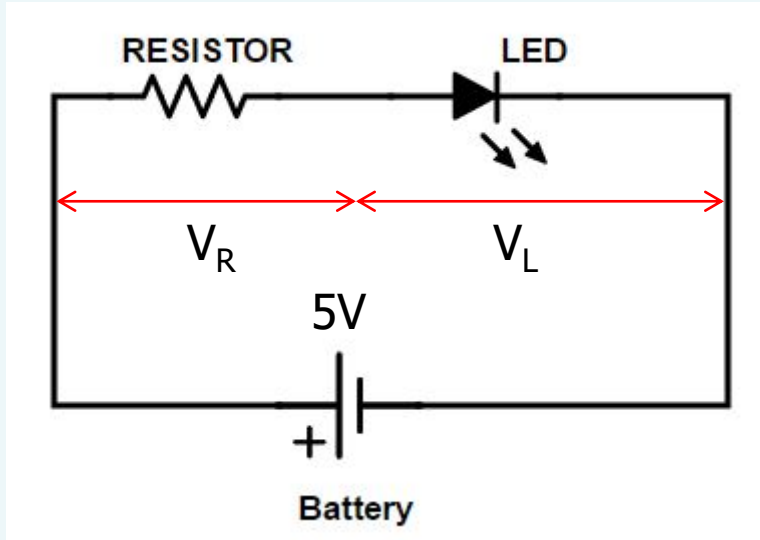
LED의 특성은 비선형임

1) 최대 허용전류가 20mA 이라면

LED 종류별로 최대 허용 전압은 얼마인가?

2) LED 사용상 주의할 점이 무엇인가?

옴의 법칙 : 전압의 분배



5mm Round LEDs	Forward Voltage		Luminous Intensity	
Emitting Color	(V) IF=20mA		(mcd) IF=20mA	
	TYP	MAX	TYP	MAX
Red	1.8	2.3	4000	5000
Yellow	1.8	2.3	4000	5000
Orange	1.8	2.3	5000	6000
Blue	3.2	3.4	6000	8000
Green	3.2	3.4	18000	20000
White	3.2	3.4	18000	20000
Warm White	3.2	3.4	13000	14000
Pink	3.2	3.4	8000	9000
UV/Purple	3.2	3.4	1000	1500

전원의 분배.. $5V = V_R + V_L$

LED 특성 "순방향 전압".. 빛을 내기 위해 필요한 전압을 의미 (Red:1.8V, White: 3.2V)

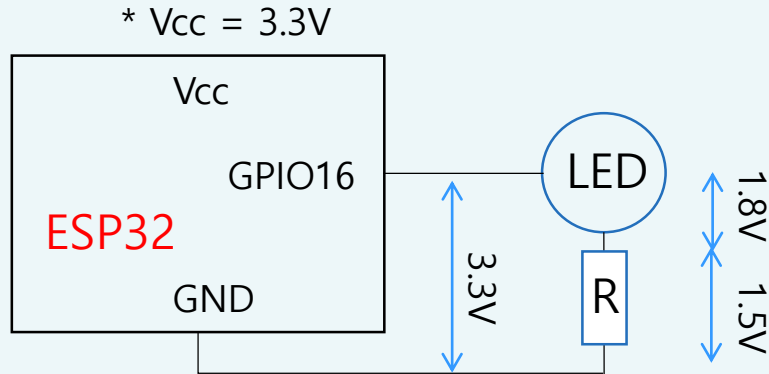
1) 적색 LED에 18mA가 흐르게 하려면 저항값을 얼마로 해야 할까?

$$(R = (Batt \text{ 전압} - LED \text{ 순방향전압}) / I)$$

2) 이 저항을 백색 LED에 사용할때 전류는? ($I = V / R$)

3) 전압이 3.3V 일때의 전류는?

ESP32 GPIO.OUT - LED



GPIO16	전압	LED
HIGH	3.3V	ON
LOW	0V	OFF

* 옴의 법칙: $V = I * R$

** Port 에서 흘릴 수 있는 전류는?

- 보통 최대 20mA
- 안전하게 5~10mA 정도로 정함

** LED ON을 위해서는 ?

“순방향 전압: 1.8V,

전류: 5~10mA 일 때”

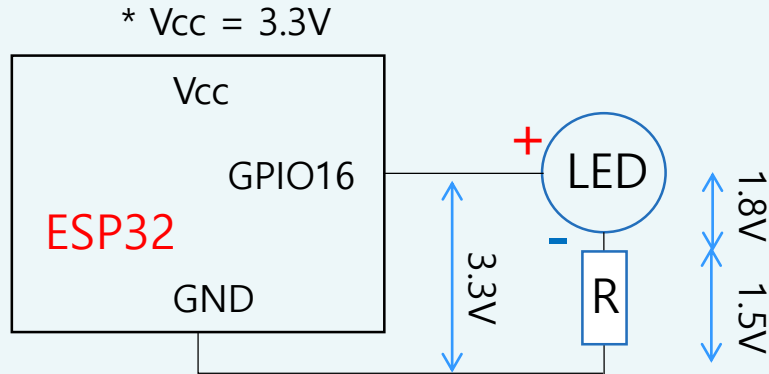
R 값은 ?

$$= V / I$$

$$= (3.3 - 1.8) / (0.005 \sim 0.010)$$

$$= 150\Omega \sim 300\Omega$$

ESP32 GPIO.OUT - LED



** Port 에서 흘릴 수 있는 전류는?

- 보통 최대 20mA
- 안전하게 5~10mA 정도로 정함

** LED ON을 위해서는 ?

"순방향 전압: 1.8V,

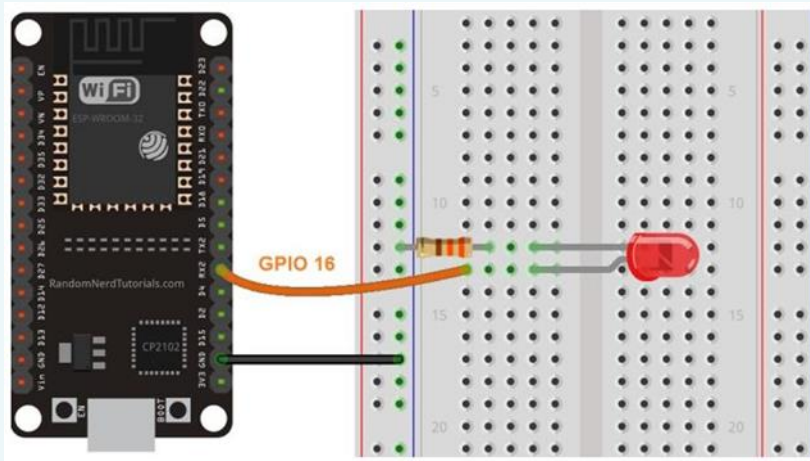
전류: 5~10mA 일 때"

R 값은 ?

$$= V / I$$

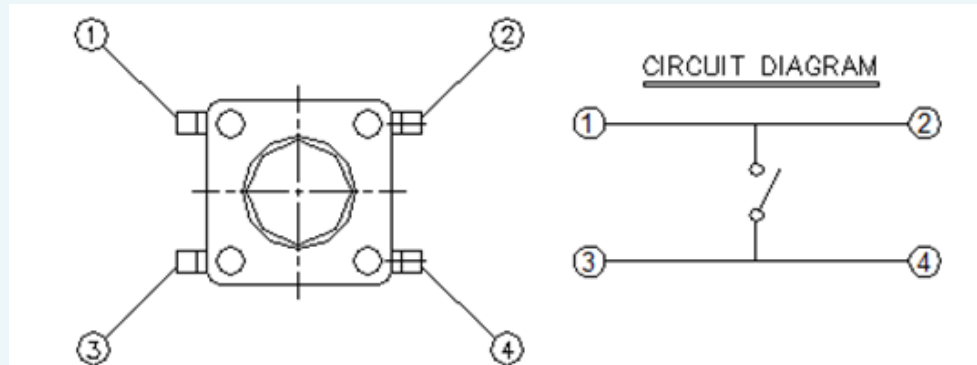
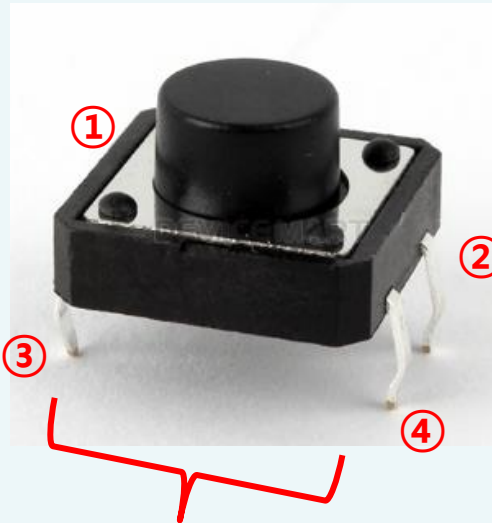
$$= (3.3 - 1.8) / (0.005 \sim 0.010)$$

$$= 150\Omega \sim 300\Omega$$



Digital Tact Switch

- **Tact** 스위치: Push Button으로 전류의 흐름을 제어
 - 평소에는 회로가 끊어져 전류가 통하지 않음
 - 버튼을 누르면 회로가 연결되어 전류가 통함

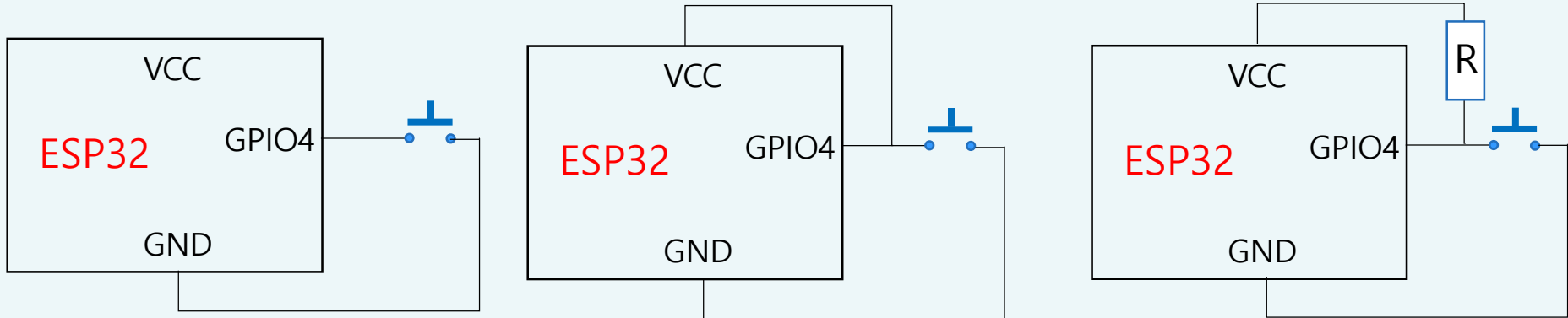


거리가 먼 단자끼리 연결되어 있다
평소: **1-2**, **3-4** 가 각각 연결되어 있음
버튼을 누르면: **1-2-3-4** 가 모두 연결됨

Digital Input Switch 1

■ Tact Switch 사용법 – Pull Up

```
pinMode(4, INPUT);
digitalRead(4);
```



스위치	GPIO4
Press	LOW
Release	?



스위치	GPIO4
Press	!!!
Release	HIGH



스위치	GPIO4
Press	LOW
Release	HIGH

? = Floating

!!! = Short

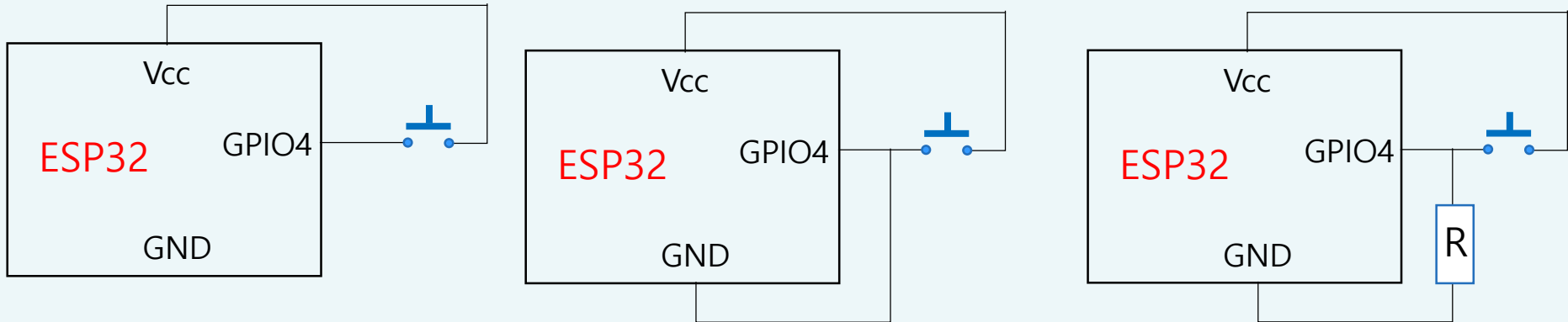
R = Pull-Up 저항 값 ?
 $R = V / I = 3.3 / 0.0003 = \text{약 } 10 \text{ K}\Omega$

* pinMode(4, INPUT_PULLUP);

Digital Input Switch 2

■ Tact Switch 사용법 – Pull Down

```
pinMode(4, INPUT);
digitalRead(4);
```



스위치	GPIO4
Press	HIGH
Release	?

? = Floating

스위치	GPIO4
Press	!!!
Release	LOW

!!! = Short

스위치	GPIO4
Press	HIGH
Release	LOW

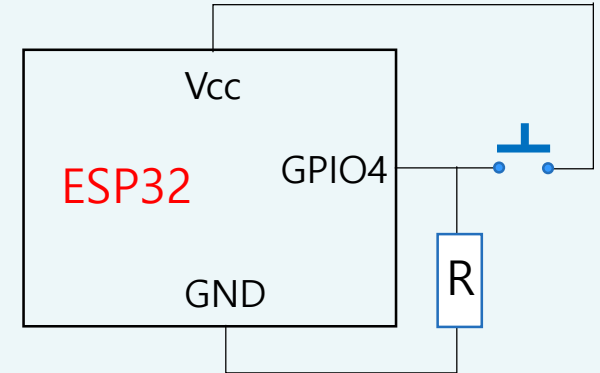
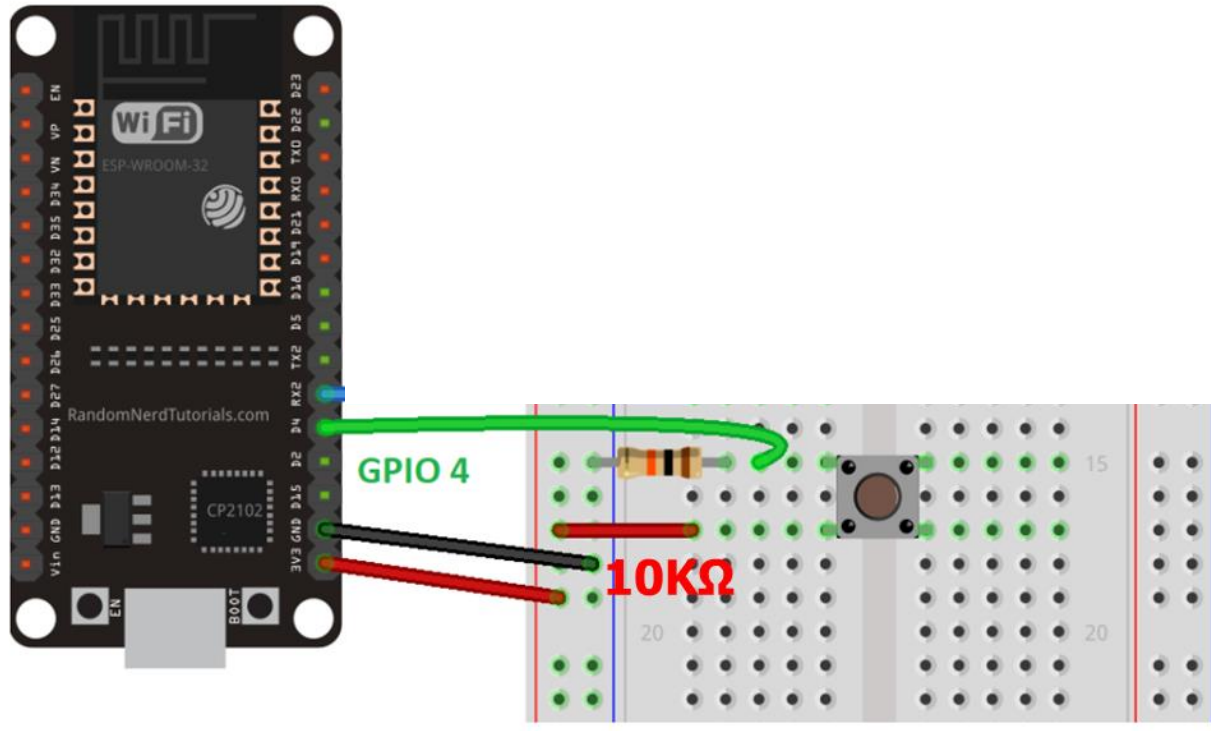
R = Pull-Down 저항 값 ?
 $R = V / I = 3.3 / 0.0003 = \text{약 } 10 \text{ K}\Omega$

* `pinMode(4, INPUT_PULLDOWN);`

Digital Input Switch 2

■ Tact Switch 사용법 – Pull Down

```
pinMode(4, INPUT);  
digitalRead(4);
```



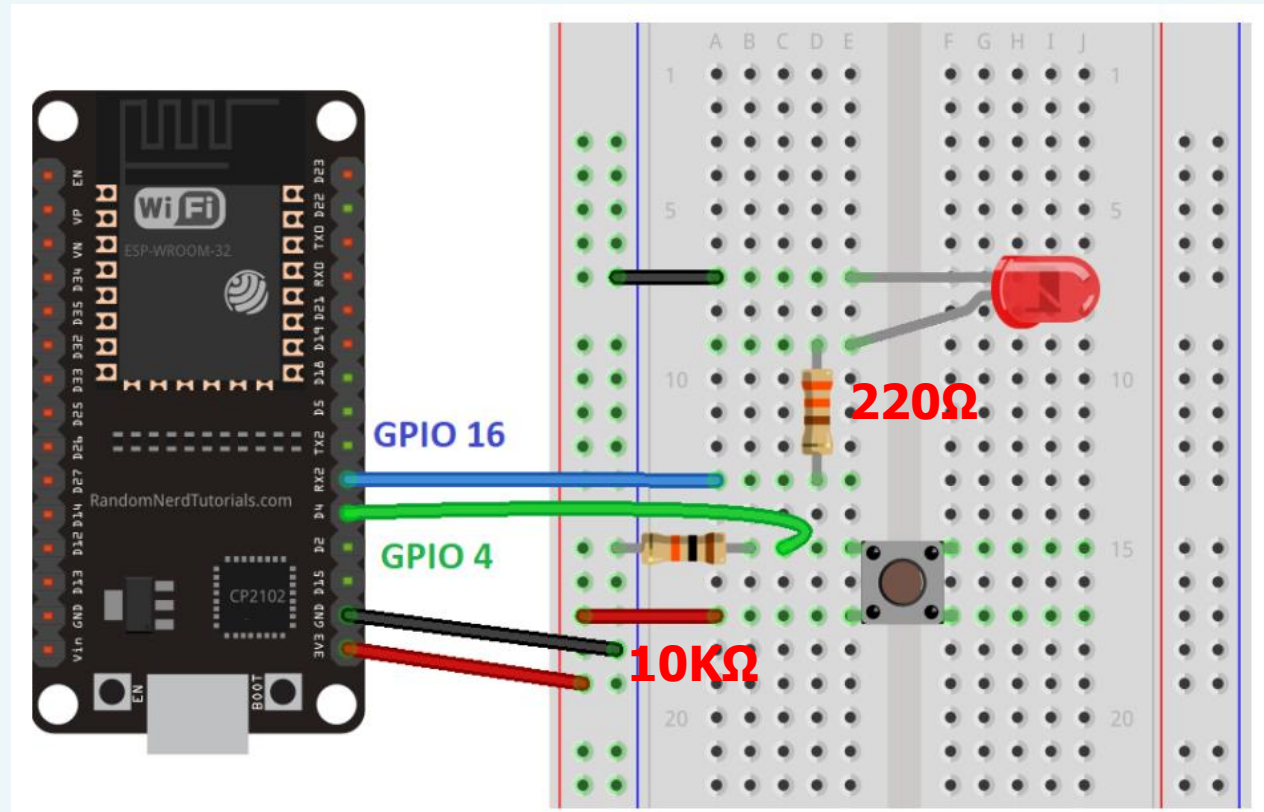
스위치	GPIO4
Press	HIGH
Release	LOW

R = Pull-Down 저항 값 ?
 $R = V / I = 5 / 0.0005 = 10 \text{ K}\Omega$

ESP32 Digital Inputs and Outputs

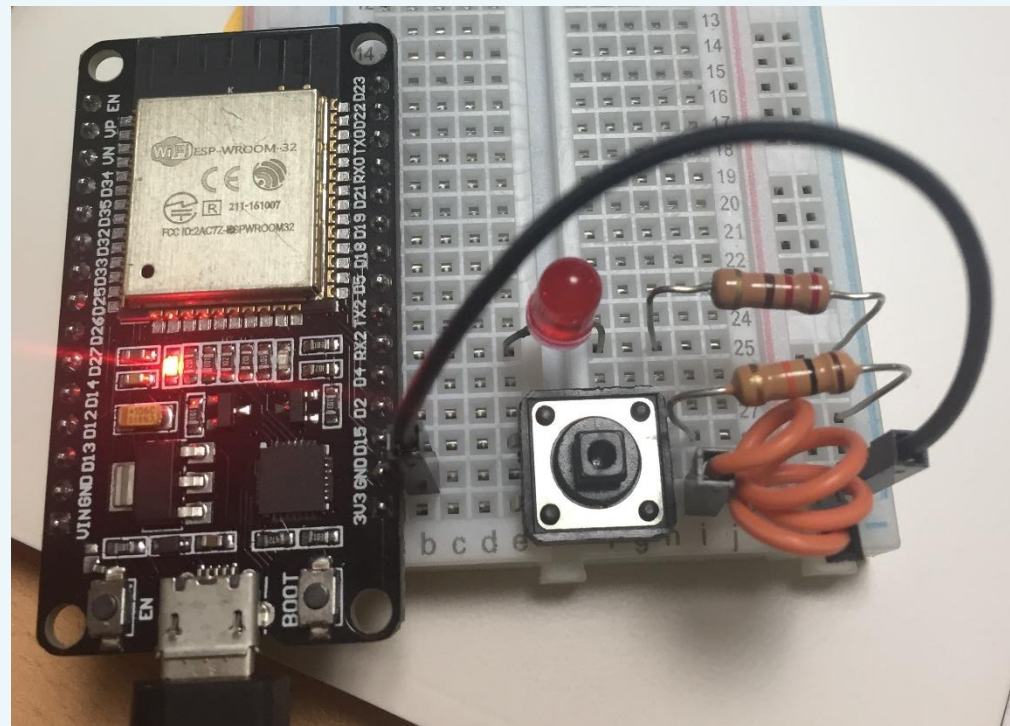
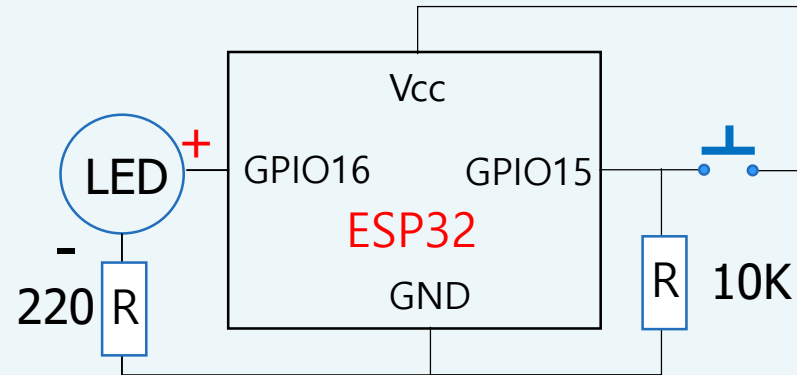
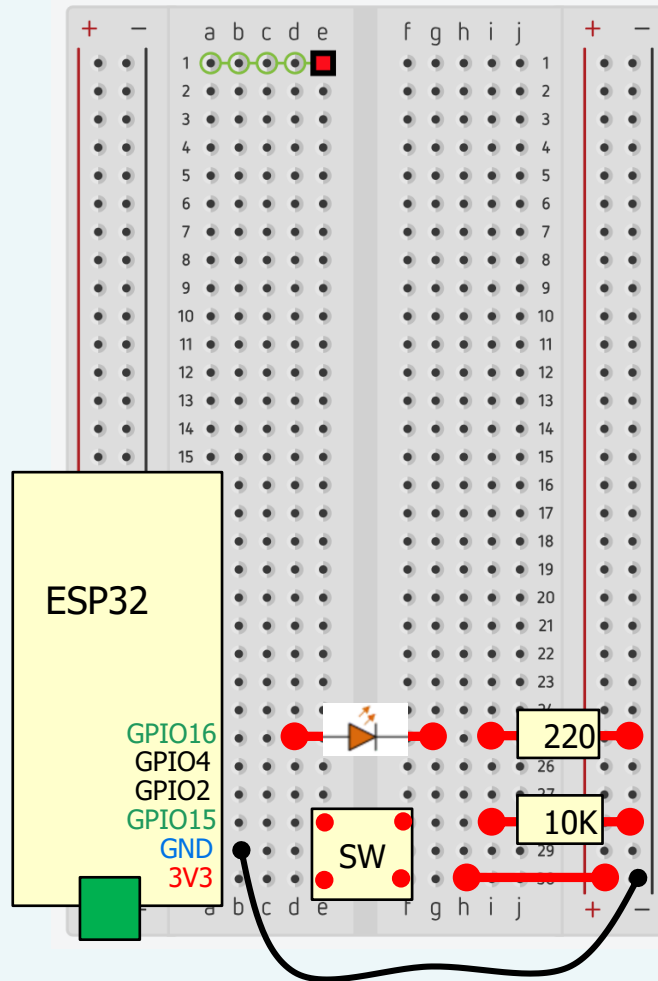
- `digitalWrite(GPIO, STATE)`
 - GPIO = GPIO No. STATE = HIGH or LOW
- `digitalRead(GPIO)`
- `pinMode(GPIO, IO)`
IO = INPUT or OUTPUT

- Example
Read Button
(GPIO:4)
Light LED
(GPIO:16)



ESP32 Digital Inputs and Outputs

** Use **GPIO15** to connect switch



ESP32 Digital Inputs and Outputs

<Task03-1>

```
// <Task 03-1>
// set pin numbers
const int buttonPin = 15; // the number of the pushbutton pin
const int ledPin = 16;    // the number of the LED pin
// variable for storing the pushbutton status
int buttonState = 0;

void setup() {
  Serial.begin(115200);
  // initialize the pushbutton pin as an input
  pinMode(buttonPin, INPUT);
  // initialize the LED pin as an output
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // read the state of the pushbutton value
  buttonState = digitalRead(buttonPin);
  Serial.println(buttonState);
  // check if the pushbutton is pressed.
  // if it is, the buttonState is HIGH
  if (buttonState == HIGH) {
    // turn LED on
    digitalWrite(ledPin, HIGH);
  } else {
    // turn LED off
    digitalWrite(ledPin, LOW);
  }
}
```

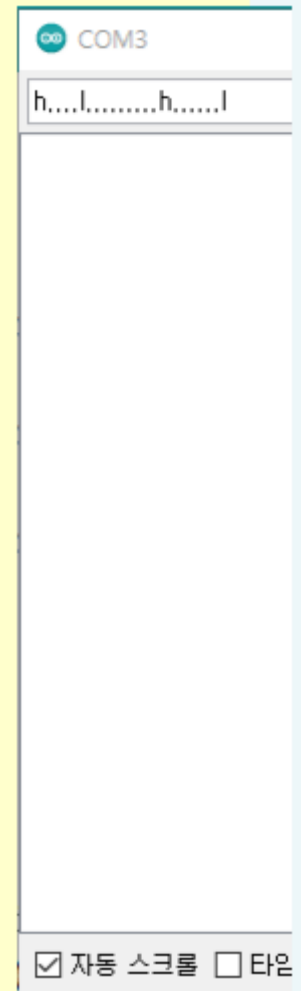
Serial Input

<Task03-2> 시리얼모니터에서 'h' 와 'l'을 입력하여 LED를 On, Off 하라
각 입력시 마다 250ms Delay 하라

```
// <Task 03-2>
// set pin numbers
const int ledPin = 16;    // the number of the LED pin
// set variables
char serState = NULL;

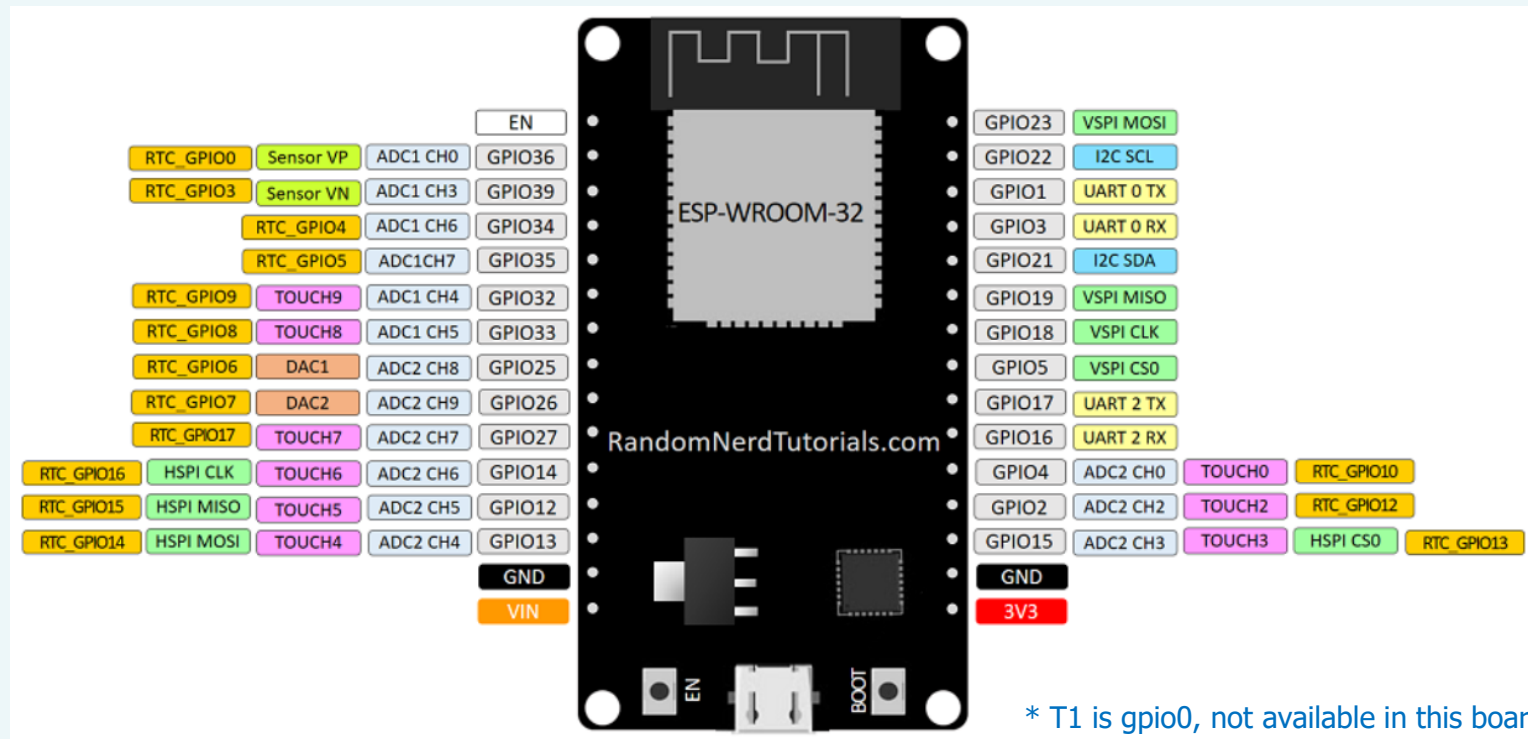
void setup() {
  Serial.begin(115200);
  // initialize the LED pin as an output
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // read the state of Serial port
  if (Serial.available() > 0)
    serState = Serial.read();
  // check which keyboard 'h' or 'l' is
  // pressed.
  if (serState == 'h') {
    // turn LED on
    digitalWrite(ledPin, HIGH);
  } else if (serState == 'l') {
    // turn LED off
    digitalWrite(ledPin, LOW);
  }
  delay(250);
}
```



ESP32 Touch Sensors

- ESP32 has 10 capacitive touch GPIOs
 - Can sense variations in electrical charge
 - Human skin, touching the GPIOs with a finger
 - Mechanical buttons

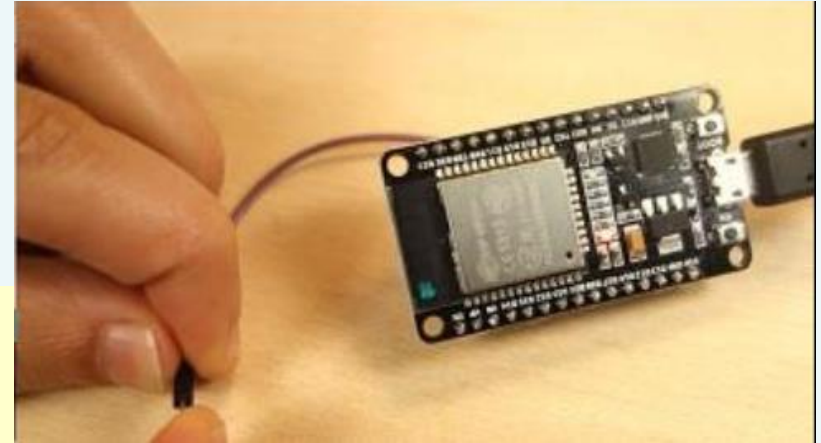


ESP32 Touch Sensors

- touchRead(GPIO)
 - Example: ESP32 > Touch > Touch Read
- Serial Monitor or Serial Plotter

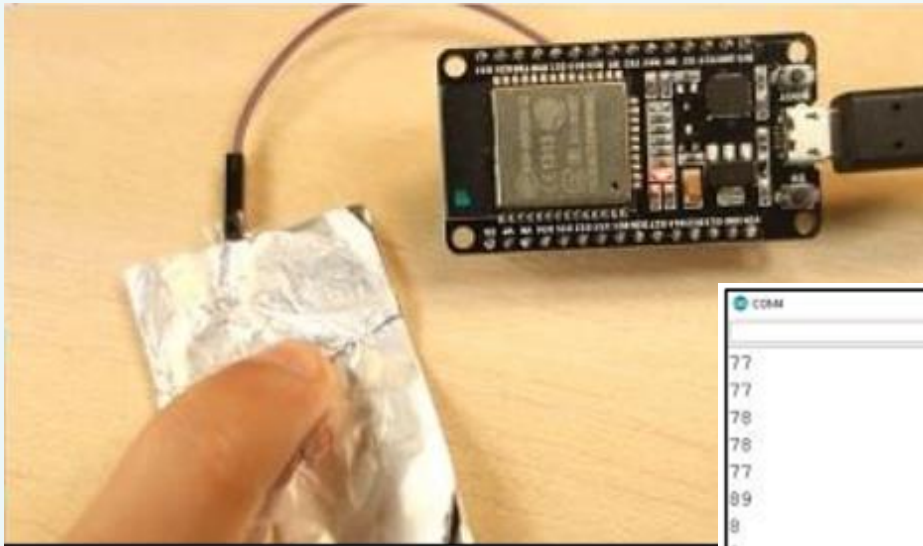
<Task03-3>

```
// <Task 03-3>
// ESP32 Touch Test
// Just test touch pin - Touch0 is T0 which is on GPIO 4.
void setup()
{
  Serial.begin(115200);
  delay(1000); // give me time to bring up serial monitor
  Serial.println("ESP32 Touch Test");
}
void loop()
{
  Serial.println(touchRead(4)); // get value of Touch 0 pin = GPIO 4
  delay(1000);
}
```



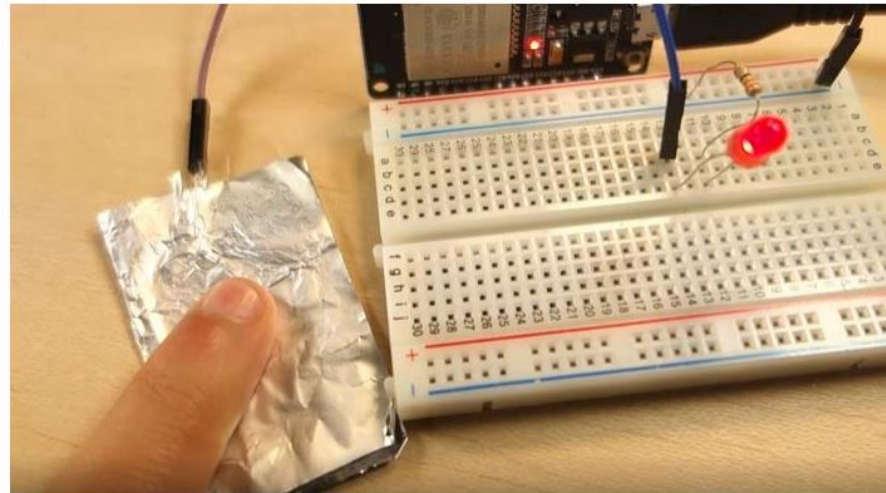
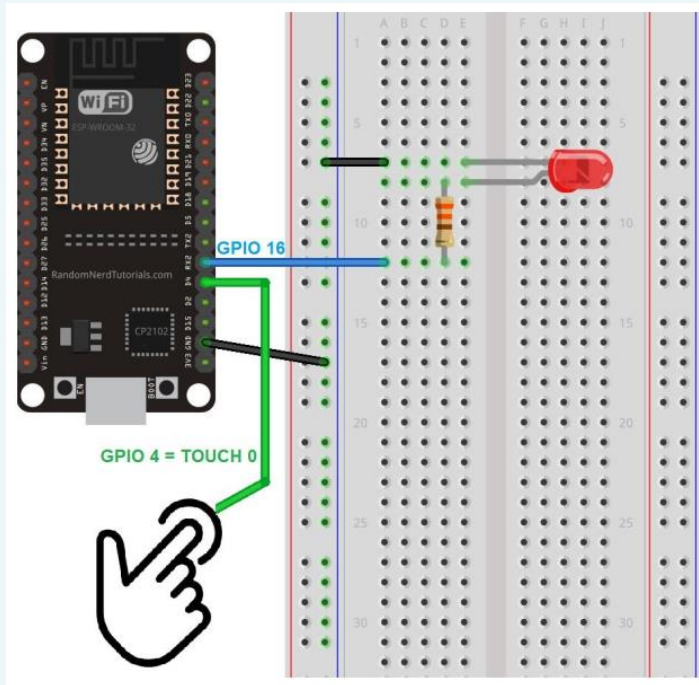
Touch Sensitive LED

- Finding the threshold
 - Touch the aluminum foil, values changing again.



Touch Sensitive LED

- Finding the threshold
 - Change the code to turn on/off by touch
 - Considering the threshold..



Touch Sensitive LED

<Task03-4>

```
// <Task 03-4>
// set pin numbers
const int touchPin = 4;
const int ledPin = 16;
// change with your threshold value
const int threshold = 20;
// variable for storing the touch pin value
int touchValue;

void setup() {
  Serial.begin(115200);
  delay(1000); // time to bring up serial monitor
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // read the touch value:
  touchValue = touchRead(touchPin);
  Serial.print(touchValue);
  // check if the touchValue below the
  // threshold
  // if it is, set ledPin to HIGH
  if(touchValue < threshold) {
    // turn LED on
    digitalWrite(ledPin, HIGH);
    Serial.println(" - LED on");
  }
  else {
    // turn LED off
    digitalWrite(ledPin, LOW);
    Serial.println(" - LED off");
  }
  delay(500);
}
```

COM3		
27	-	LED off
13	-	LED on
12	-	LED on
12	-	LED on
69	-	LED off
71	-	LED off
70	-	LED off
69	-	LED off
70	-	LED off
70	-	LED off
70	-	LED off
70	-	LED off
69	-	LED off
68	-	LED off
<input checked="" type="checkbox"/> 자동 스크롤 <input type="checkbox"/> 타임-		