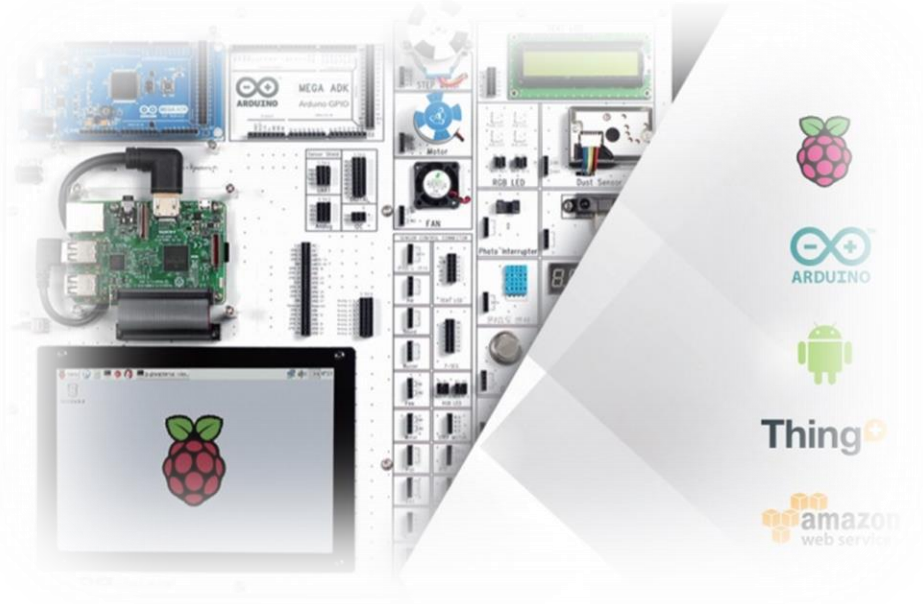


2021년도 2학기

휴먼스마트기기설계



교과목명 : 휴먼스마트기기설계 (01)

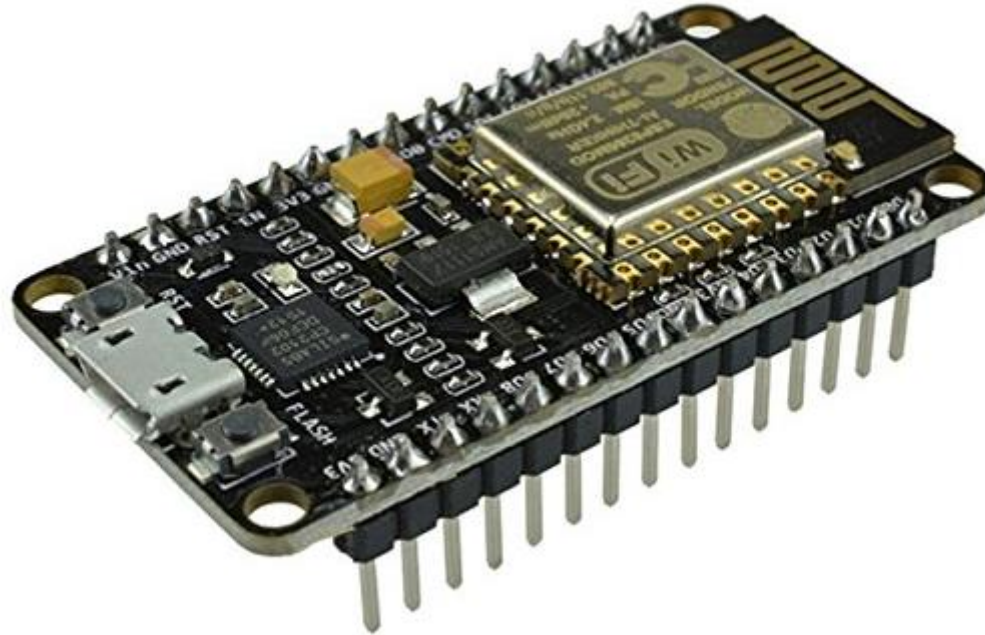
담당교수 : 이 수 형

E-mail : soohyong@uu.ac.kr

교재명 : 스마트기기 개발을 위한 사물인터넷, 이수형. (LINC+ 사업단 배포)

3부 : 네트워크

8. 원격 제어기

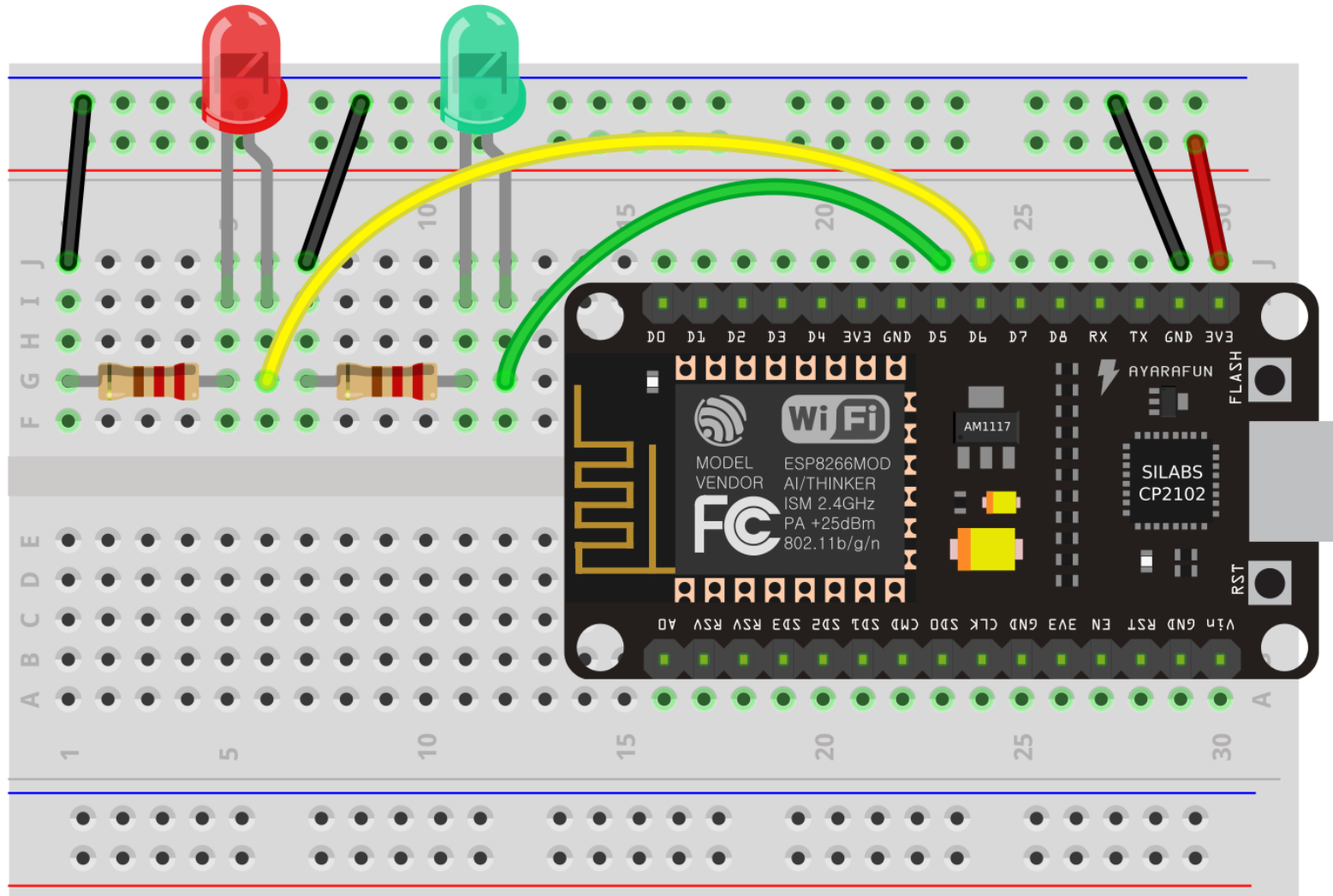


원격 제어기

- 사물인터넷 기기의 용도
 - 센서 측정된 값 $\Rightarrow$ 인터넷 $\Rightarrow$ 서버 $\Rightarrow$ 원격에서 모니터링 (예: 기상 관측소)
 - 원격 제어 S/W $\Rightarrow$ 서버 $\Rightarrow$ 사물인터넷 기기 : 제어
- 원격 제어 방법
 - 원격 제어기 $\rightarrow$ NodeMCU (서버 및 제어기) $\Rightarrow$ 제어 (LED, 모터 등)
 - 원격 제어기 $\rightarrow$ 서버 (MQTT 등) $\rightarrow$ NodeMCU (제어 신호 수신) $\Rightarrow$ 제어

8.1 원격 LED 제어 #1

- 회로 구성



8.1 원격 LED 제어 #2

- 스케치 : 예제 8-1. MQTT를 이용한 원격 LED 제어
 - MQTT 서비스에서 구독(subscribe) 기능으로 구현

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include "ssid.h"

// LED용 핀 정보
#define LED1 D5
#define LED2 D6

// MQTT용 객체
WiFiClient client;
PubSubClient mqtt(client);

#define MQTT_SERVER "broker.hivemq.com"
#define MQTT_PORT 1883

// MQTT용 콜백함수 (메시지 수신시 처리함수)
void callback(char *topic, byte *payload, unsigned int length);
```

8.1 원격 LED 제어 #3

- setup() 함수 설정
 - WiFi 연결 후 MQTT 서버 연결

```
void setup()
{
    // 포트 설정
    pinMode(LED1, OUTPUT);
    pinMode(LED2, OUTPUT);

    // 시리얼 통신 초기화 & 무선 인터넷 시작 (앞 예제 참고)
    // ... 생략 ...

    // MQTT용 클라이언트 초기화 (서버주소 및 포트)
    mqtt.setServer(MQTT_SERVER, MQTT_PORT);
    mqtt.setCallback(callback);
}
```

8.1 원격 LED 제어 #3

- loop() 함수 설정
 - 필요한 토픽 구독 (앱 등을 통해서 토픽 발행)

```
void loop() {  
  // MQTT 서버 접속이 안되었거나 끊어진 경우 재접속  
  if(!mqtt.connected()) {  
    // 접속이 실패하는 경우 2초 후에 다시 시도  
    if(!mqtt.connect("uidukshlee")) {  
      Serial.println("Connect failed !!!");  
      delay(2000);  
      return;  
    }  
    if(!mqtt.subscribe("uiduk/shlee/#")) {  
      Serial.println("Subscribe failed !!!");  
    }  
  }  
  // MQTT 처리  
  mqtt.loop();  
}
```

구독 : MQTT 정보 수신

8.1 원격 LED 제어 #4

- callback() 함수 – MQTT 서버에서 전달한 정보 확인
 - 서버에서 수신된 토픽 / 값을 처리하는 함수
 - 수신된 토픽 / 값을 시리얼 포트를 통해서 PC에 전송

```
void callback(char *topic, byte *payload, unsigned int length) {  
    Serial.print("Message arrived [");  
    Serial.print(topic);  
    Serial.print("] ");  
    for (int i=0;i<length;i++) {  
        Serial.print((char)payload[i]);  
    }  
    Serial.println();  
    // ...  
}
```

8.1 원격 LED 제어 #5

- callback() 함수 – MQTT 서버에서 전달한 정보에 따라 제어

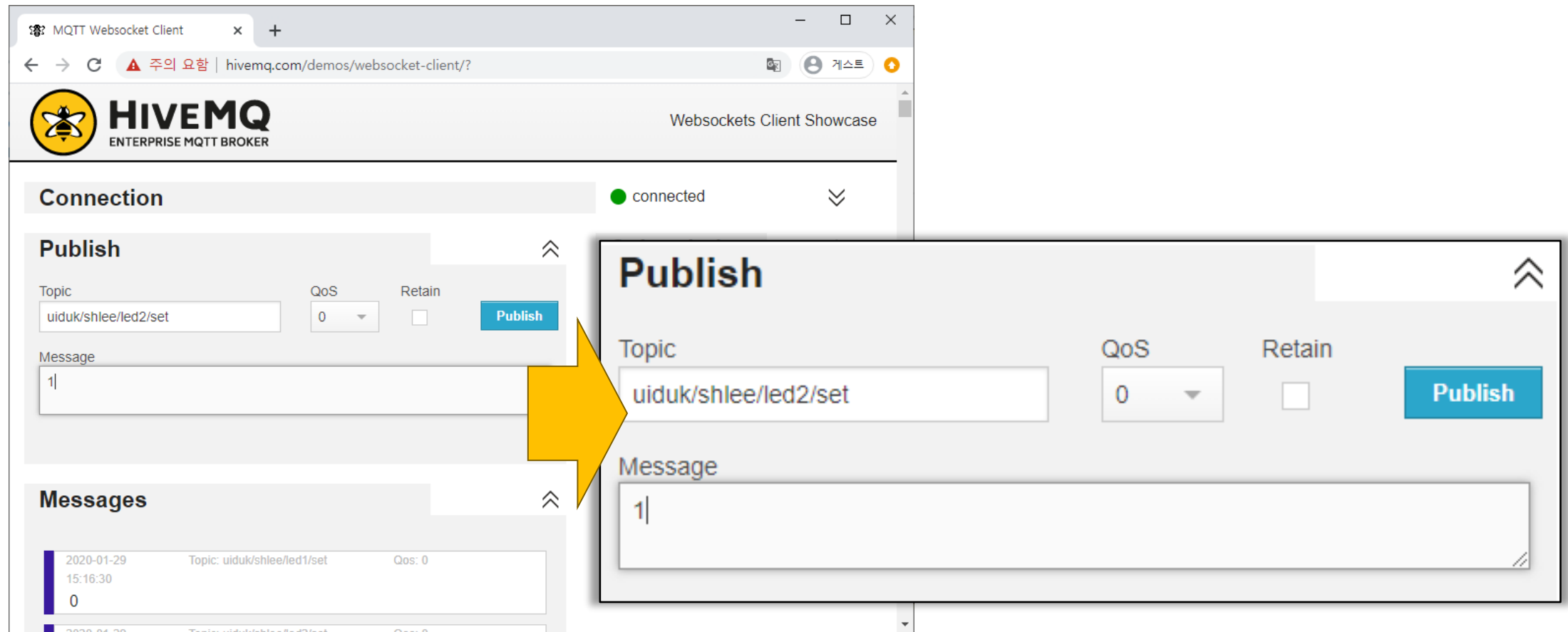
```
// topic에 따라 분석
String top(topic); // == 연산자를 위해서 String으로 변환
// LED1의 제어
if(top == "uiduk/shlee/led1/set") {
    if(payload[0] == '1')
        digitalWrite(LED1, HIGH);
    else
        digitalWrite(LED1, LOW);
}
// LED2의 제어, LED1과 동일한 방식
if(top == "uiduk/shlee/led2/set") {
    digitalWrite(LED2, payload[0] == '1' ? HIGH : LOW);
}
}
```

/uiduk/shlee/led1/set의 값이 “1” 인경우 LED ON

위와 동일한 처리 방식

8.1 원격 LED 제어 #6

- 수행
 - MQTT 서버에 발행 (웹 브라우저 또는 다른 모바일 기기의 앱 사용)



8.1 원격 LED 제어 #7

• 수행 (모바일 기기)

The sequence of screenshots illustrates the setup of remote LED control:

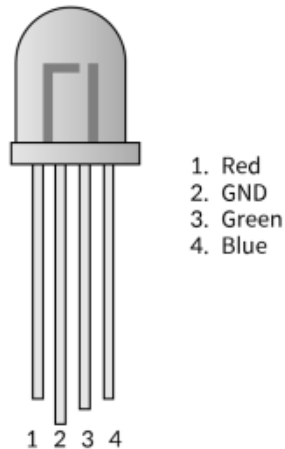
- MQTT Dashboard (Disconnected):** Shows the 'Choose component type' menu. The 'Switch' option is highlighted with a red dashed box and an arrow pointing to the next screen.
- Publication (Type: Switch):** Shows the configuration for 'LED 1'. The 'Topic' is 'uiduk/shlee/led1/set'. The 'Publish value (On)' is set to '1'.
- Publication (Type: Switch):** Shows the configuration for 'LED 2'. The 'Topic' is 'uiduk/shlee/led2/set'. The 'Publish value (On)' is set to '1'.
- MQTT Dashboard (Connected):** Shows the final dashboard with two switch controls. The 'LED 2' switch is in the 'On' position (red), and the 'LED 1' switch is in the 'Off' position (grey).

A yellow callout box points to the 'Publish value (On)' field in the LED 1 configuration screen, containing the text: 스위치 ON인 경우 발행할 값 : "1"

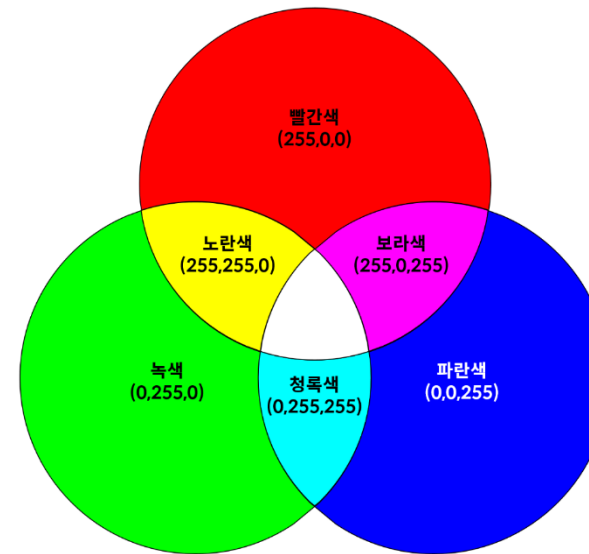
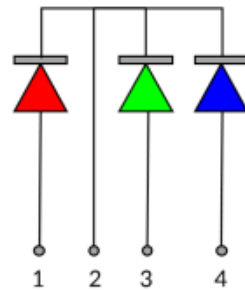
8.2 원격 RGB LED 제어

- RGB LED

- LED내부에 3색의 LED가 들어가있는 소자
- 각각의 색의 밝기를 조절함으로 모든 색을 표현



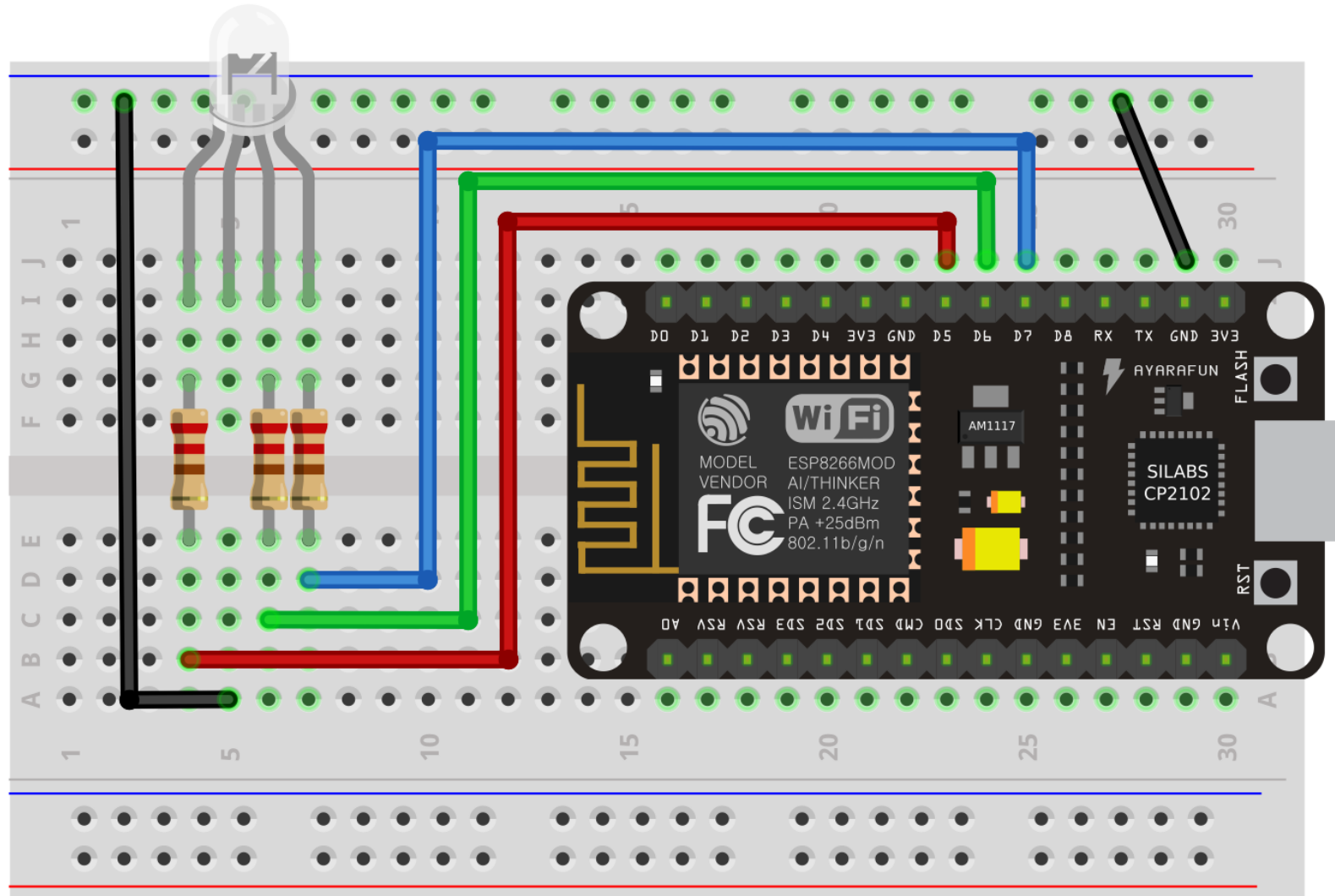
RGB LED의 구조
(common cathode)



빛의 삼원색

예제 8-2. RGB LED의 원격 제어 #1

- 회로 구성



예제 8-2. RGB LED의 원격 제어 #2

- MQTT 서비스에서 구독(subscribe) 기능으로 구현 (예제 8-1 참고)

```
// MQTT를 이용한 원격 RGB LED 제어
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include "ssid.h"

// LED용 핀 정보
#define LEDR D5
#define LEDG D6
#define LEDB D7

// MQTT용 객체
WiFiClient client;
PubSubClient mqtt(client);

#define MQTT_SERVER "broker.hivemq.com"
#define MQTT_PORT 1883
```

예제 8-2. RGB LED의 원격 제어 #3

- setup() 함수
 - WiFi 연결 후 MQTT 서버 연결 (예제 8-1 참고)

```
void setup()
{
    // 포트 설정
    pinMode(LED_R, OUTPUT);
    pinMode(LED_G, OUTPUT);
    pinMode(LED_B, OUTPUT);

    // 시리얼 통신 초기화 & 무선 인터넷 시작 (앞 예제 참고)
    // ... 생략 ...

    // MQTT용 클라이언트 초기화 (서버주소 및 포트)
    mqtt.setServer(MQTT_SERVER, MQTT_PORT);
    mqtt.setCallback(callback);
}
```

예제 8-2. RGB LED의 원격 제어 #4

- loop() 함수 설정 (예제 8-1과 동일)
 - 필요한 토픽 구독 (앱 등을 통해서 토픽 발행)

```
void loop() {  
  // MQTT 서버 접속이 안되었거나 끊어진 경우 재접속  
  if(!mqtt.connected()) {  
    // 접속이 실패하는 경우 2초 후에 다시 시도  
    if(!mqtt.connect("uidukshlee")) {  
      Serial.println("Connect failed !!!");  
      delay(2000);  
      return;  
    }  
    if(!mqtt.subscribe("uiduk/shlee/#")) {  
      Serial.println("Subscribe failed !!!");  
    }  
  }  
  // MQTT 처리  
  mqtt.loop();  
}
```

구독: MQTT 정보 수신

예제 8-2. RGB LED의 원격 제어 #5

- callback() 함수 – MQTT 서버에서 전달한 정보 확인
 - 서버에서 수신된 토픽 / 값을 처리하는 함수
 - 수신된 토픽 / 값을 시리얼 포트를 통해서 PC에 전송 (예제 8-1과 동일)

```
void callback(char *topic, byte *payload, unsigned int length) {  
    Serial.print("Message arrived [");  
    Serial.print(topic);  
    Serial.print("] ");  
    for (int i=0;i<length;i++) {  
        Serial.print((char)payload[i]);  
    }  
    Serial.println();  
    // ...  
}
```

예제 8-2. RGB LED의 원격 제어 #6

- callback() 함수 – MQTT 서버에서 전달한 정보에 따라 제어

```
// 수신 메시지 : #ABCDEF 의 형태로 16진수로 입력됨 -> 16진수 변환
String top(topic); // == 연산자를 위해서 String으로 변환
// 하나만 구독하므로 토픽을 비교할 필요는 없음
```

```
if(top == "uiduk/shlee/color/set") {
    unsigned char r = convertHex(&payload[1]); // 두번째 글자
    unsigned char g = convertHex(&payload[3]); // 두번째 글자
    unsigned char b = convertHex(&payload[5]); // 두번째 글자
```

/uiduk/shlee/color/set의 값을 분석

```
// 해석한 16진수 문자열 확인용
```

```
char buf[20];
sprintf(buf, "# %02X %02X %02X", r, g, b);
Serial.println(buf);
```

#ABCDEF 형식의 16진수 2자리씩 해석

```
// 읽어들이는 색상을 기준으로 PWM 방식으로 출력
```

```
analogWrite(LED_R, r * 4);
analogWrite(LED_G, g * 4);
analogWrite(LED_B, b * 4);
```

#ABCDEF 형식의 색상 정보

- AB : Red 색 성분 16진수 2자리
- CD : Green 색 성분 16진수 2자리
- EF : Blue 색 성분 16진수 2자리

예제 8-2. RGB LED의 원격 제어 #7

- convertHex() 함수 – #ABCDEF형식의 16진수 해석

```
unsigned char convertHex(byte *hex) {  
    // 변환될 숫자  
    int n, number = 0;  
    // 위치별 숫자를 곱하는 숫자, 첫번째문자*16 + 두번째 문자  
    int mul[2] = { 16, 1 };  
  
    for(int i=0; i<2; i++) {  
        int ch = toupper(hex[i]); // 대문자로 변환  
        if(ch >= '0' && ch <= '9') { // 아라비아 숫자인 경우  
            n = ch - '0'; // 숫자기호로 된 문자를 숫자로 변환  
        }  
        else if(ch >= 'A' && ch <= 'F') { // 영문자인 경우  
            n = ch - 'A' + 10;  
        }  
        else { return 0; } // 오류...  
        number += n * mul[i]; // 자리수만큼 곱해서 더하기  
    }  
    return (unsigned char) number;  
}
```

[예] hex = "3b"

ch : '3' (i == 0), ch : 'B' (i == 1)

ch : '3', n : '3' - '0' = 3 (i == 0)

n : 'B' - 'A' + 10 = 10 (i == 1)

(i == 0) number += 3 * 16
(i == 1) number += 10 * 1

number = 3 * 16 + 10 ⇒ 58

예제 8-2. RGB LED의 원격 제어 #8

- 수행 (모바일 기기)

