

아두이노(Arduino) 소개



아두이노 소개

- **오픈소스를 기반으로 한 단일 보드 마이크로 컨트롤러**
- **작은 컴퓨터**
 - 80년대 중후반에 나온 80286PC(16MHz) 정도의 속도로 동작하는 컴퓨터
 - 메모리나 저장공간, 입출력 장치가 매우 제한적이거나 없는 대신 다른 LED, 모터, 입력장치, 출력장치 등등을 제어하는데 특화된 컴퓨터
- **설계도가 모두 공개되어 있는 오픈 하드웨어**
 - 정식 아두이노 보드와 아두이노 호환 보드

2

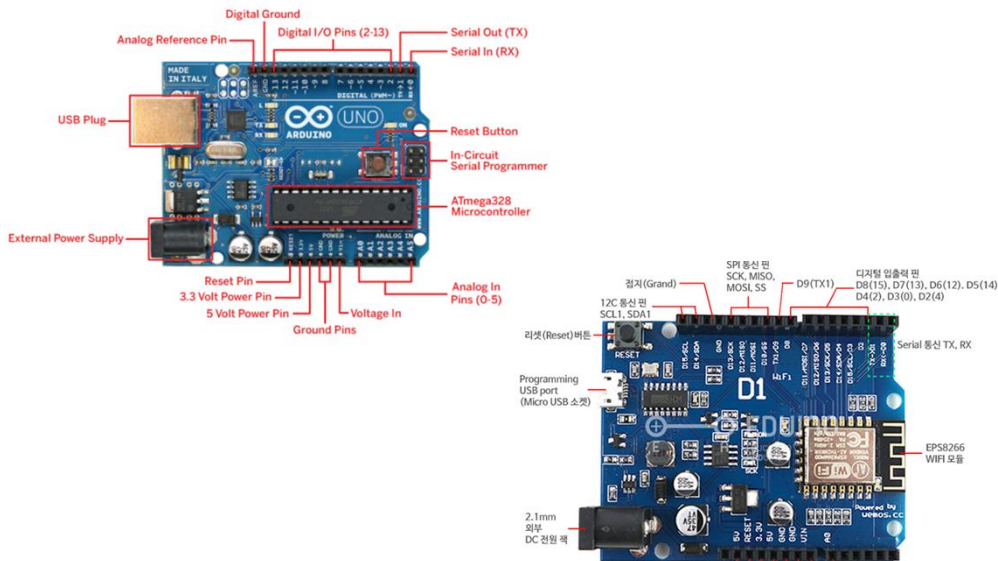
아두이노(이탈리아어: Arduino 아르두이노)는 오픈소스를 기반으로 한 단일 보드 마이크로컨트롤러로 완성된 보드(상품)와 관련 개발 도구 및 환경을 말한다. 2005년 이탈리아의 IDI(Interaction Design Institutelvera)에서 하드웨어에 익숙지 않은 학생들이 자신들의 디자인 작품을 손쉽게 제어할 수 있게 하려고 고안되었다.

아두이노는 다수의 스위치나 센서로부터 값을 받아들여, LED나 모터와 같은 외부 전자 장치들을 통제함으로써 환경과 상호작용이 가능한 물건을 만들어 낼 수 있다.

임베디드 시스템 중의 하나로 쉽게 개발할 수 있는 환경을 이용하여, 장치를 제어할 수 있다.

아두이노 통합 개발 환경(IDE)을 제공하며, 소프트웨어 개발과 실행코드 업로드도 제공한다.

아두이노 UNO와 D1 R1(WiFi)



3

아두이노 우노는 가장 많이 사용되는 기본적인 아두이노 보드이다.

아두이노 우노는 **ATmega328** 기반의 **마이크로 컨트롤러 보드**로 14 개의 디지털 입출력 핀 (6 개는 PWM 출력으로 사용할 수 있음), 6 개의 아날로그 입력, 16MHz 세라믹 공진기, USB 연결, 전원 잭, ICSP 헤더 및 재설정 버튼이 있다.

아두이노 WIFI D1 R1 보드는 아두이노우노보드와 ESP8266 WIFI 모듈을 혼합한 보드이다.

ESP8266 모듈은 2014년 처음 등장한 WIFI 통신모듈이다.

아두이노 WIFI D1 R1 보드는 우노와 형태는 같지만 ESP-8266EX프로세서로 동작하는 다른 보드이다.

하지만 기존의 아두이노 IDE에서 프로그래밍이 가능하고 아두이노 실드들과 호환이 가능하도록 만들어진 보드이다.

특히, **3.3V에서 동작하며 80Mhz 클럭 속도로 매우 빠른** 프로세싱이 가능하고 플래시메모리를 4M byte를 보유하고 있다.

아두이노 UNO Pinout

DC Power Jack
7-12VDC Input
2.1mm x 5.5mm
Male Center Positive

USB-B Port
To Computer

Reset Button

UNO Pinout

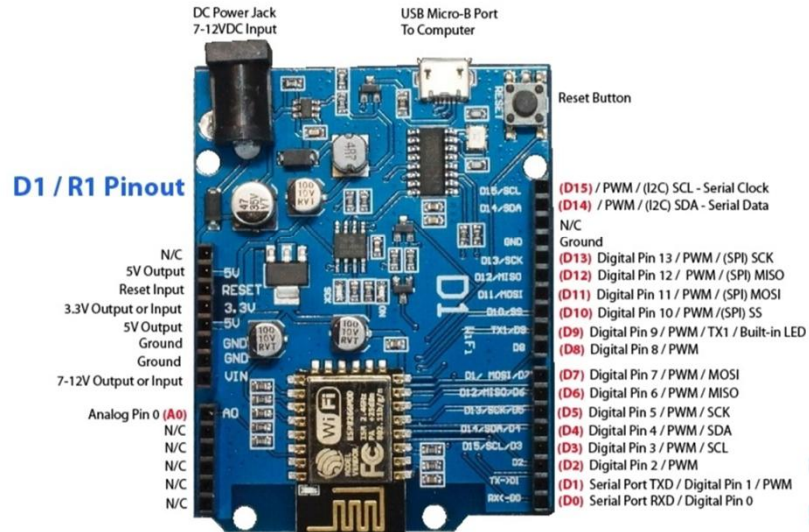
No Connection
I/O Reference Voltage for shields
Reset Input
3.3V Output @ 50mA
5V Output or Input
Ground
Ground
7-12V Output or Input

Analog Pin 0 (A0)
Analog Pin 1 (A1)
Analog Pin 2 (A2)
Analog Pin 3 (A3)
Analog Pin 4 (A4)
Analog Pin 5 (A5)

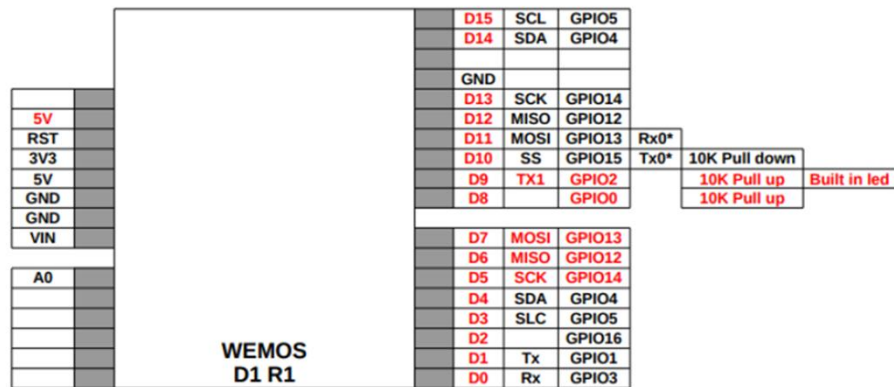
(I2C) SCL - Serial Clock
(I2C) SDA - Serial Data
Analog Reference Voltage
Ground
(13) Digital Pin 13 / (SPI) SCK / Connected to on-board LED
(12) Digital Pin 12 / (SPI) MISO
(11) Digital Pin 11 / PWM / (SPI) MOSI
(10) Digital Pin 10 / PWM / (SPI) SS
(9) Digital Pin 9 / PWM
(8) Digital Pin 8
(7) Digital Pin 7
(6) Digital Pin 6 / PWM
(5) Digital Pin 5 / PWM
(4) Digital Pin 4
(3) Digital Pin 3 / PWM / Ext Int 1
(2) Digital Pin 2 / Ext Int 0
(1) Serial Port TXD / Digital Pin 1
(0) Serial Port RXD / Digital Pin 0



D1 R1 Pinout



D1 R1



GPIO란 General Purpose Input/Output 의 줄임말로, 마이크로 컨트롤러의 핀을 입력 또는 출력 용도로 사용할 수 있게 하는 기능이다.
GPIO핀을 제어해서 핀의 전압 값을 읽어 들일 수도 있고, 핀이 연결된 회로에 전압을 인가해 줄 수도 있다.

아두이노를 동작시키기 위한 과정



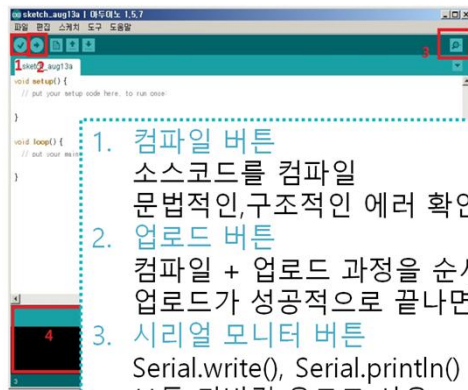
7

[아두이노 동작과정]

1. 아두이노와 각종 장치들(LED, 모터, 센서, 모듈 등등)을 연결
2. 아두이노 개발환경(Integrated Development Environment, 통합개발환경)을 실행
3. 아두이노에 연결된 장치들을 원하는 대로 동작하도록 스케치(Sketch, 소스코드) 작성
4. 아두이노 개발환경에서 스케치 컴파일
 - 아두이노에 넣어도 문제가 없는지, 문법이 틀리지는 않았는지 검사하고 아두이노에 넣을 수 있는 형태로 바꾸는 과정이다.
 - 컴파일 과정에 문제가 없으면 아두이노의 마이크로 컨트롤러에 프로그램을 넣게 된다.
5. 아두이노 개발환경에서 컴파일된 바이너리를 아두이노 보드에 업로드
6. 아두이노가 스케치대로 동작

아두이노 개발환경

- <https://www.arduino.cc/> 다운로드



1. 컴파일 버튼
소스코드를 컴파일
문법적인, 구조적인 에러 확인
2. 업로드 버튼
컴파일 + 업로드 과정을 순서대로 실행
업로드가 성공적으로 끝나면 4번 영역에 '업로드 완료' 메시지가 표시
3. 시리얼 모니터 버튼
Serial.write(), Serial.println() 등의 함수를 사용해서 serial 출력
보통 디버깅 용도로 사용
4. 메시지 영역
알림 메시지나 컴파일, 업로드 결과 등등 메시지가 표시되는 영역

8

[아두이노 IDE설치]

1. 인터넷 창을 열고, <https://www.arduino.cc/> 주소로 접근한다.
2. 다운로드(Download) 메뉴를 클릭하고 내 컴퓨터의 운영체제에 맞는 소프트웨어를 선택한다.
3. 소프트웨어를 설치한다.

[아두이노 IDE와 아두이노 연결하기]

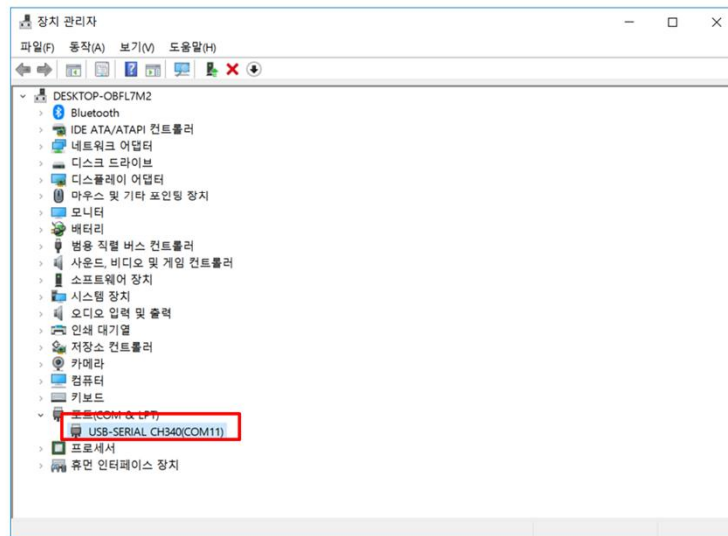
1. 아두이노 보드와 컴퓨터를 USB 케이블로 연결한다.
2. 일반적으로 아두이노 IDE 프로그램을 설치하면 드라이버 프로그램도 함께 설치된다.
3. 장치관리자에서 아두이노가 몇 번 포트에 연결되어 있는지 확인한다.
4. 아두이노 IDE 프로그램의 '도구' 메뉴를 선택하고 '시리얼 포트'를 . 방금 본 장치관리자의 시리얼포트 번호와 같은지 확인해 보고, 다르다면 같은 숫자로 바꾸어준다.

[자동으로 연결되지 않을 경우]

- 시작 메뉴 클릭 > 컴퓨터 오른쪽 마우스 클릭 > 속성 클릭 > 장치관리자 클릭
- 포트(Ports COM & LPT)에 아두이노(Arduino)가 보이는지 확인
- 오른쪽 마우스를 클릭하고 드라이버 소프트웨어 업데이트를 선택
- 드라이버를 선택(아두이노 IDE 프로그램 폴더에서 선택)

아두이노 UNO WiFi D1 R1 환경구축

• 장치관리자에서 확인



[보드인식 확인]

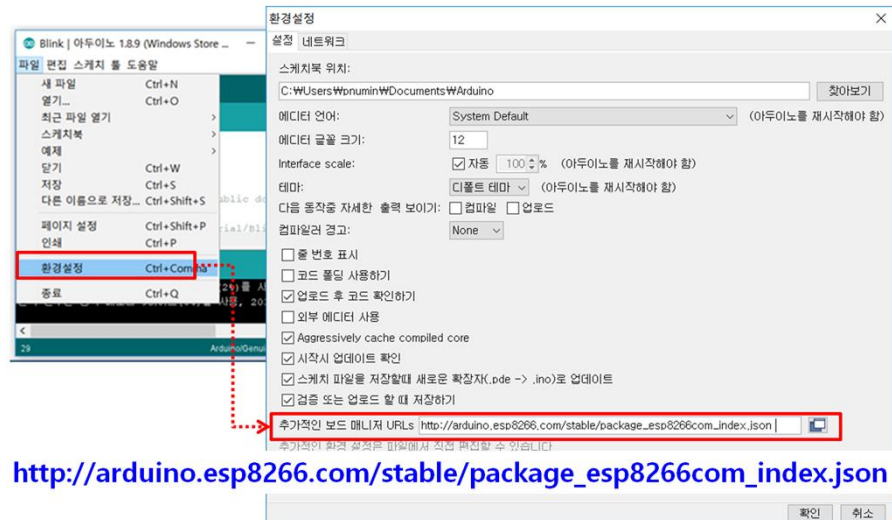
-장치관리자의 포트 확인

-포트 확인 시 드라이버가 설치 되어 있지 않으면 아래의 사이트에 접속하여 드라이버를 설치한다.

-http://www.wch.cn/download/CH341SER_ZIP.html

아두이노 UNO WiFi D1 R1 환경구축

• 아두이노 IDE에서 보드 추가



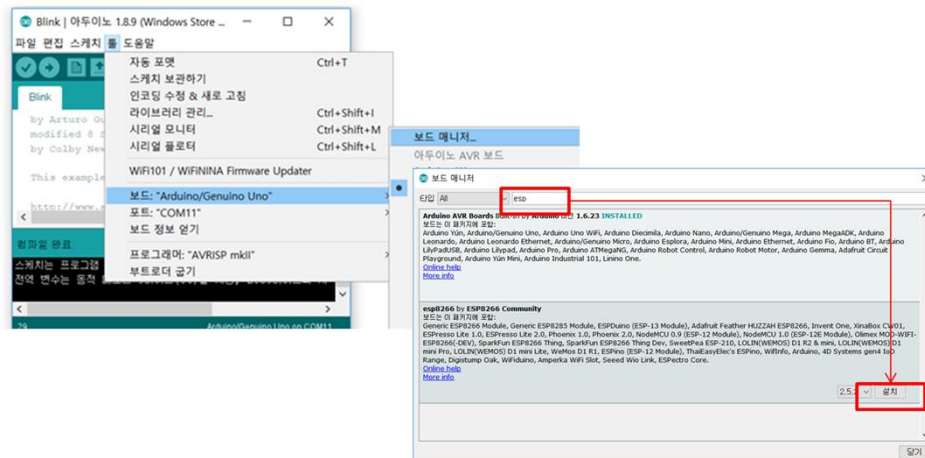
[환경설정]

-추가적인 보드 매니저

-http://arduino.esp8266.com/stable/package_esp8266com_index.json

아두이노 UNO WiFi D1 R1 환경구축

• 아두이노 IDE에서 보드 추가

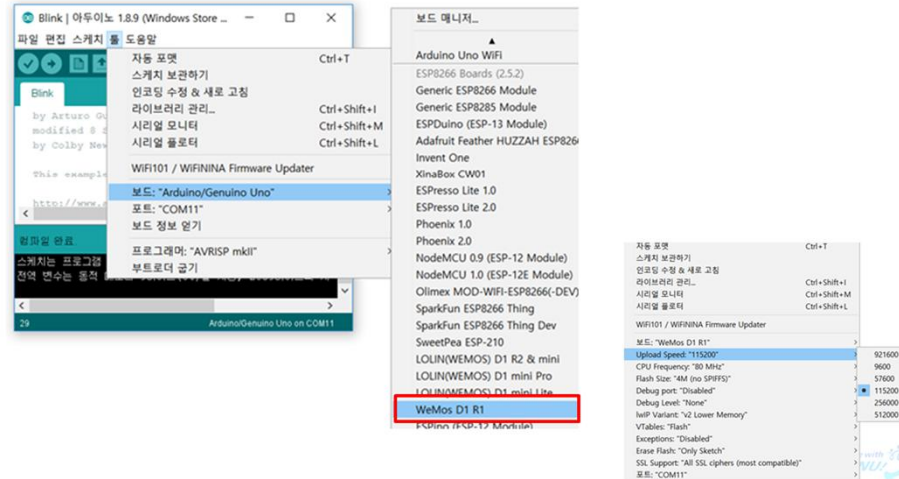


11

[보드 매니저]
-Esp8266 추가

아두이노 UNO WiFi D1 R1 환경구축

• 아두이노 IDE에서 보드 설정



[보드변경]

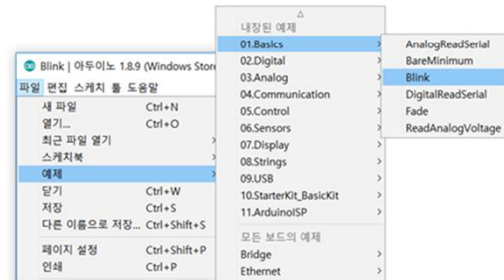
툴-> 보드 : Webmos D1 R1

툴-> Upload Speed : 115200

아두이노 UNO WiFi D1 R1 환경구축

• 설정 테스트

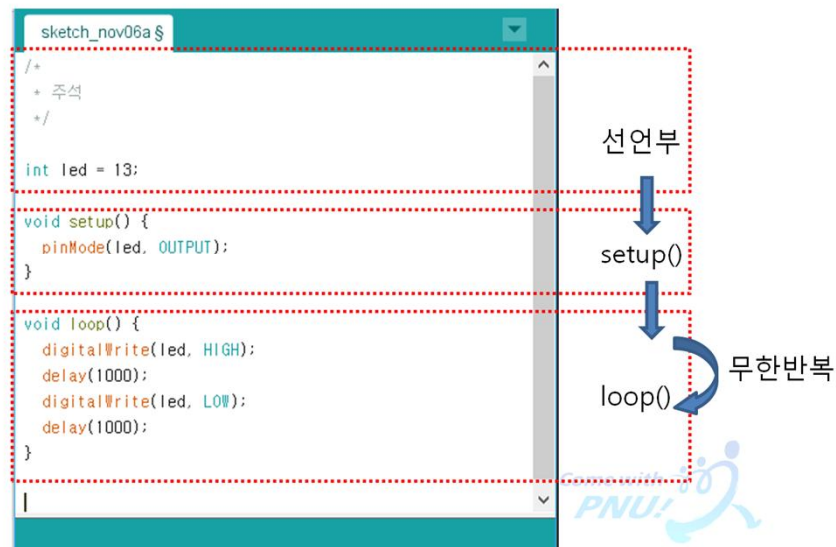
- 파일 > 예제 > 01.Basics > Blink
- 컴파일하고 업로드



[스케치(Sketch)]

- 아두이노 보드에서 동작하는 프로그램을 만들기 위해 사용되는 프로그래밍 언어이다.
- 아두이노 개발환경(IDE)에서 작성하고 컴파일한다.
- 컴파일 과정이 완료되면 아두이노 보드에 업로딩 한다.

아두이노 소스의 기본구조



[프로그램 기본 구조]

1. 주석 및 전역변수 선언 영역

- 보통 전역변수나 라이브러리 헤더 include.
- 클래스 인스턴스 선언하는 공간

2.setup() 함수

- 아두이노에 전원이 들어오거나 reset 되었을 때 처음 1회만 실행되는 함수
- 아두이노가 가진 pin 들을 초기화하거나 소스코드 내에서 사용되는 변수 및 각종 하드웨어를 초기화

3.loop() 함수

- setup이 실행된 후 무한 반복해서 호출되는 함수
- 아두이노가 동작하는 동안 실행 될 핵심 동작코드가 위치

[변수]

- 변수는 메모리 공간에 저장하는 영역으로 변수선언을 통해서 공간을 확보

[변수선언]

- 데이터타입 변수명 = 값
int led = 13 ;

[데이터타입]

- 문자 : char (1byte) 아스키코드 값
- 정수 : int(2byte), byte(1byte), long(4byte)

- 실수 : float(4byte), double(8byte)

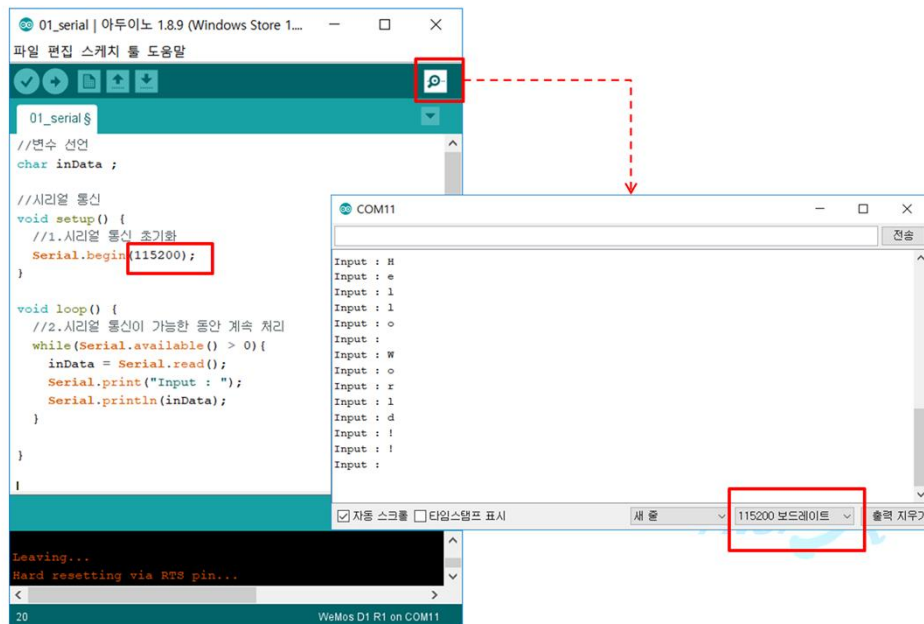
시리얼 통신



[시리얼(Serial)통신]

- 일련의 데이터를 비트 단위로 순차적으로 전송하는 통신규약
- 아두이노에서 시리얼 통신은 보드와 PC간의 통신을 목적으로 사용
- RX와 TX를 통해 USB로 연결된 PC와 시리얼 통신 수행
- 시리얼 통신을 위해서는 통신설정(보드레이트:Baudrate)을 맞추어야 함
: 단위(BPS:Bit Per Second)

시리얼 통신



[시리얼통신]

Serial.begin(speed)

- 시리얼 통신을 사용하기 위해 사용하고자 하는 통신 포트를 초기화(레지스터 값 설정)

Serial.available()

- 시리얼 포트로부터 수신받은 데이터가 있는지 확인할 때 사용
- 수신받은 데이터가 0 초과라면, 즉 데이터가 존재한다면 존재함을 의미

Serial.read()

- 수신받은 데이터를 읽는데 사용

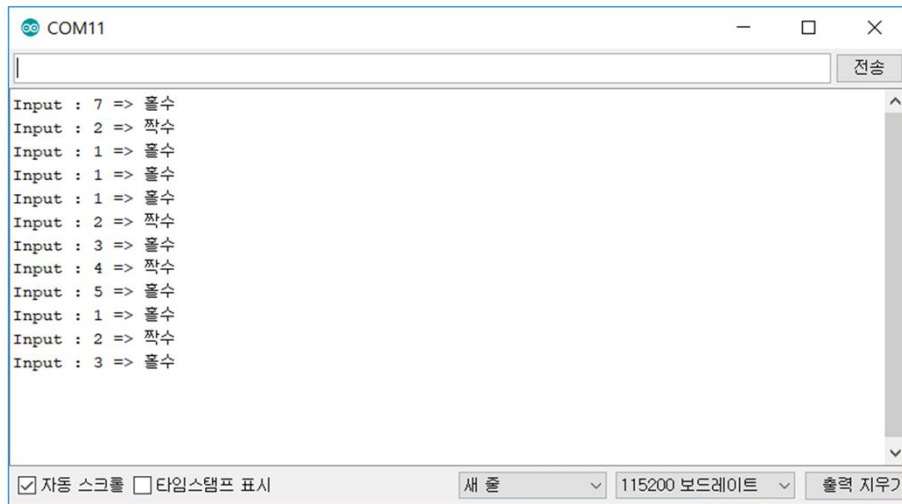
Serial.print(), Serial.println()

- 시리얼 포트를 통하여 데이터를 출력시키는데 사용
- Serial.print() 는 줄바꿈을 하지 않고 Serial.println() 는 줄바꿈 함

[반복문 : while]

- 조건이 참일 동안 계속 반복

해결문제



해결문제 : 시리얼통신을 통해서 입력 받은 숫자를 짝수와 홀수로 구분하시오.

[연산자]

- 사칙연산 : +, -, *, /
- 나머지연산자 : %
- 비교연산자 : >, <, >=, <=, ==, !=
- 논리연산자 : && (and), ||(or), !(not)

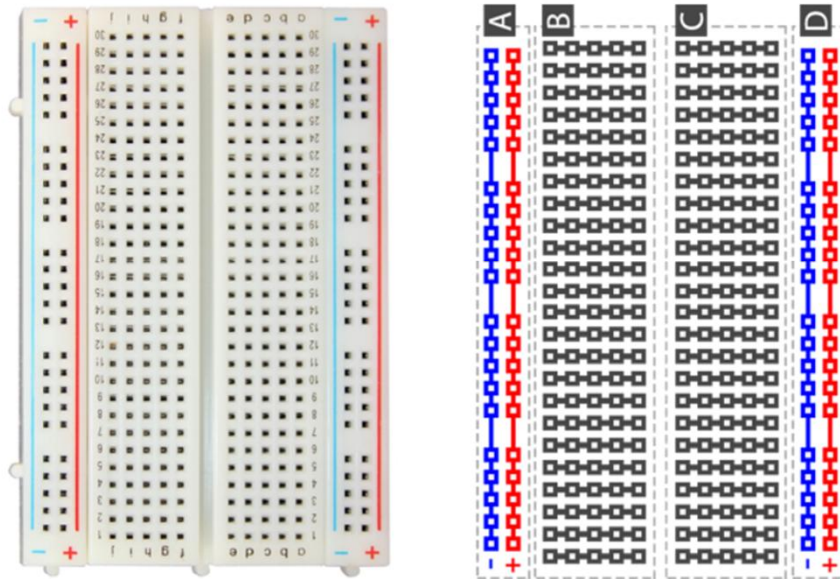
[조건문: if..else]

- 조건에 따라 수행문 결정

LED 제어하기 (digital output)



아두이노 부품-브레드보드



[브레드보드(breadboard)]

- 0.25센티미터 간격으로 구멍이 나있는 플라스틱으로 된 틀 아래에 전류가 흐를 수 있는 라인이 배치된 납땀이 필요 없는 형태의 기판
- 전기 및 전자 회로의 실험에서 기판에 납땀을 하지 않고도 회로를 구성할 수 있는 도구
- 브레드보드는 세로로 파진 홈인 중앙선의 좌우에 위치한 수평 방향의 단자 띠와 수직 방향의 버스 띠로 구성
- 단자 띠는 부품간의 연결을 위한 것이고 버스 띠는 전원공급

아두이노 부품-LED



- 매우 작은 전류를 사용하며 빛을 발산하는 부품
- LED를 전원소스에 직접 연결하면 LED는 타버리게 됨

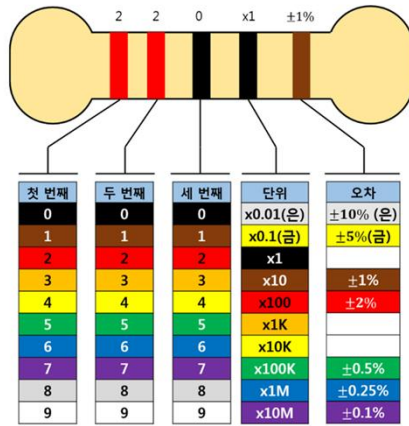
- 저항을 이용하여 LED로 들어오는 전류의 양을 제한



[LED]

- 전류를 빛으로 전환하는 기본적인 반도체 소자로 발광 다이오드
- (+)단자와 (-)단자로 구성
- 다리가 긴 쪽이 (+) 단자, 짧은쪽이 (-)단자
- 전원을 입력해줄 때는 단자를 맞춰 입력 해줘야하며, 적정 전압을 입력해줘야 함
- 극을 반대로 연결하거나 높은 전압을 인가할 경우 LED가 터질 수 있으니 주의

아두이노 부품-저항



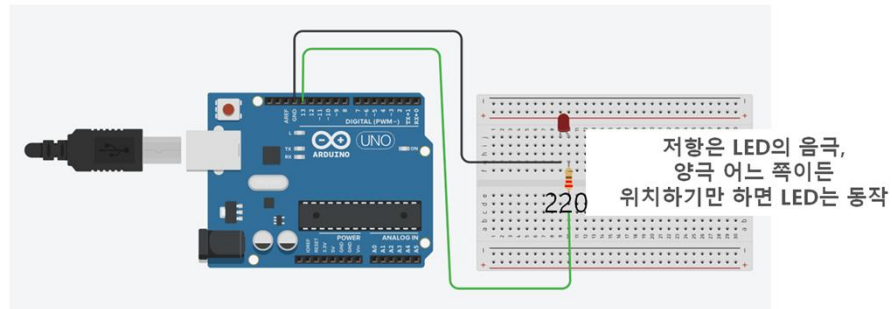
< 220 Ω 저항 >

| 색 | 값 |
|-----|------|
| 검정색 | 0 |
| 갈색 | 1 |
| 빨강색 | 2 |
| 주황색 | 3 |
| 노란색 | 4 |
| 초록색 | 5 |
| 파란색 | 6 |
| 보라색 | 7 |
| 회색 | 8 |
| 하얀색 | 9 |
| 은색 | ±10% |
| 금색 | ±5% |

[저항]

- 저항은 전기의 흐름을 조절 해서 전자 부품이 과전류로 인해 손상되는 것을 방지
- 아두이노에서 기본적으로 내보내는 5V를 직접적으로 LED에 보내게 되면 LED가 손상되기 때문에 저항과 함께 사용해야 함
- 저항에는 극성이 없음
- 저항의 값은 저항에 색칠되어 있는 컬러로 확인

LED 불 깜빡이기



```
void setup() {
  // initialize digital pin LED as an output.
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);            // wait for a second
  digitalWrite(13, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);            // wait for a second
}
```



pinMode()

- 특정한 핀을 입력으로 쓰지 출력으로 쓰지를 설정

digitalWrite()

- GPIO output 기능을 수행

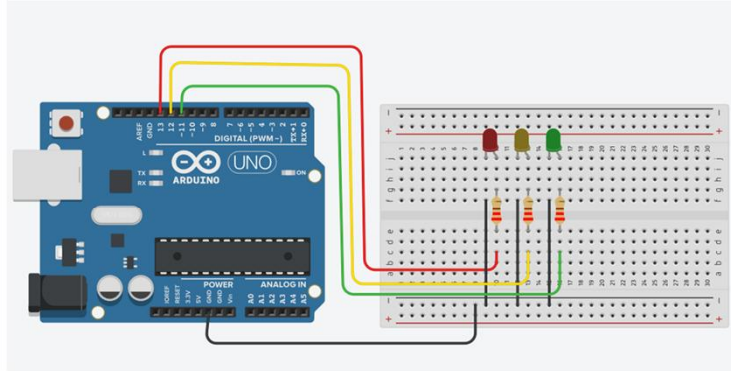
digitalWrite(pin number, HIGH)

- 해당되는 핀에 5V를 출력

digitalWrite(pin number, LOW)

- 해당되는 핀에 0V를 출력하게 된다.

해결문제



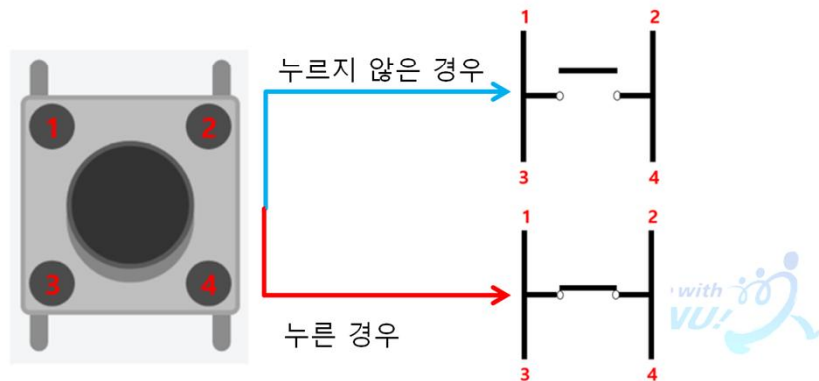
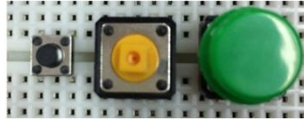
Come with
PNU!

해결문제 : 1초에 한번씩 3색의 LED가 하나씩 불이 켜지도록 작성하세요.

버튼제어하기 (digital input)



아두이노 부품-push 버튼



[푸시버튼]

- 푸시버튼은 회로도 상에서 1번과 3번, 2번과 4번 단자는 서로 연결
- 푸시버튼을 누르면 1번, 2번, 3번, 4번이 모두 연결된 상태
- 저항은 VCC단자에 달아주는 것이 중요

[풀업(PULL-UP)과 풀다운(PULL-DOWN)]

풀업과 풀다운은 방법은 같고 VCC과 GND와 같이 서로 반대이다.

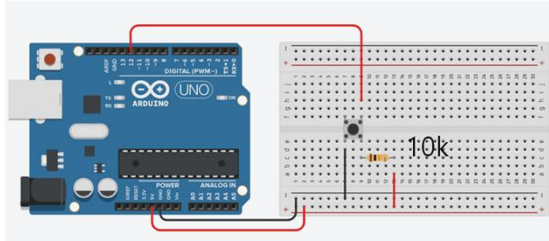
풀업(PULL-UP)

- ON : 0(LOW), OFF : 1(HIGH)

풀다운(PULL-DOWN)

- ON : 1(HIGH), OFF : 0(LOW)

버튼 입력



```
//디지털 입력 저장변수  
int din = 0;  
//디지털 입력 핀 번호  
int btnpin = 12;
```

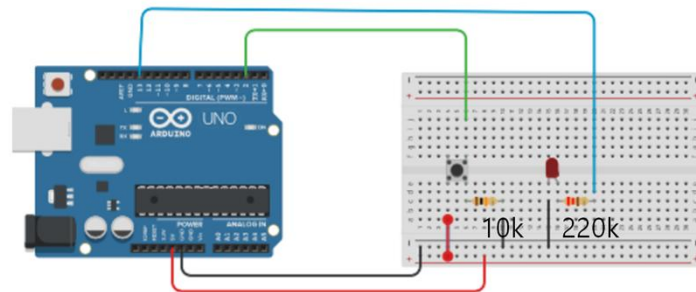
```
void setup() {  
  Serial.begin(115200);  
  //입력핀 설정  
  pinMode(btnpin, INPUT);  
}
```

```
void loop() {  
  //디지털 입력  
  din = digitalRead(btnpin);  
  //입력값 시리얼 통신  
  Serial.println(din);  
  delay(100);  
}
```

digitalRead()

- GPIO의 입력기능 구현

해결문제

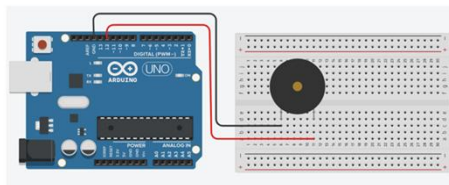


해결문제 : 버튼을 눌렀을 때 불이 켜지도록 작성하시오.

수동부저



아두이노 부품 - 피에조 부저



```
//음정의
#define Do 262
#define Re 294
#define Mi 330
#define Fa 349
#define Sol 392
#define Ra 440
#define Si 494
#define Do2 523
```

```
//부저 핀번호
int buzzerPin = 12;
```

```
//노래
int play[] = {Sol,Sol,Ra,Ra,Sol,Sol,Mi,Sol,Mi,Mi,Re} ;
int i;
```

```
void setup() {
  pinMode(buzzerPin, OUTPUT);
}
void loop() {
  for(i=0; i<12; i++){
    tone(buzzerPin, play[i], 200);
    delay(500);
  }
  noTone(buzzerPin);
}
```



29

[피에조 부저 (Piezo Buzzer)]

- 얇은 금속판을 주기적으로 떨리게하여서 소리를 출력하는 소형 부품
- 극성이 있음

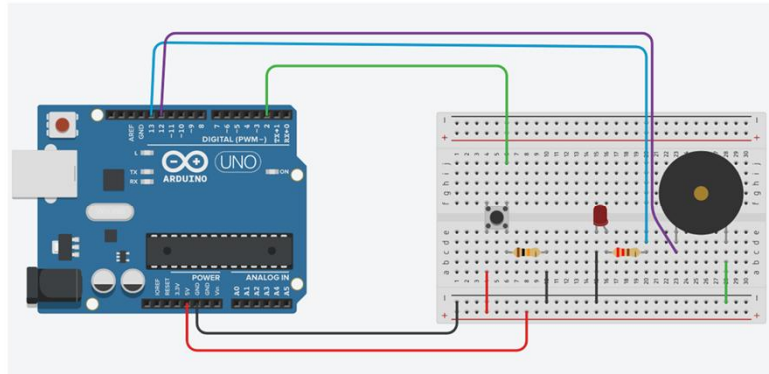
tone(핀번호, 음의 높낮이, 음의 지속시간) 함수

- 지정포트로 지정 주파수 신호를 길이(1/1000초단위)로 출력하여 소리가 나도록 함

noTone(포트)함수

- 지정하는 포트의 출력 끄기

해결문제



Come with
PNU!

30

해결문제 : 버튼을 누르면 LED에 불이 오고 부저 소리를 내시오.

적외선 모션 감지 센서



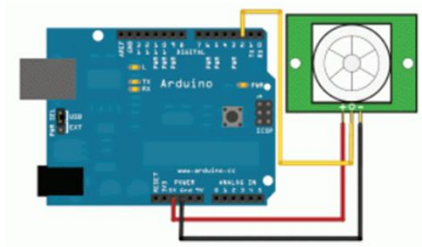
아두이노 부품-모션감지센서



[모션감지센서]

- 적외선 PIR센서(PIR, Passive Infrared Sensor)
- 수동적외선센서로써 적외선을 통해 **사람의 움직임(모션, motion)**을 감지하는 센서
- 센서 앞에 사물이 지나가면 적외선의 변화가 생기며 이것을 감지
- 감지각도범위 안에 적외선(빛)의 변화가 있을 시 High(1) 신호를, 없을 시 Low(0) 값을 출력하는 센서

적외선 모션 감지 센서



```
//핀번호
int pirPin = 5;

//입력 변수
int din;

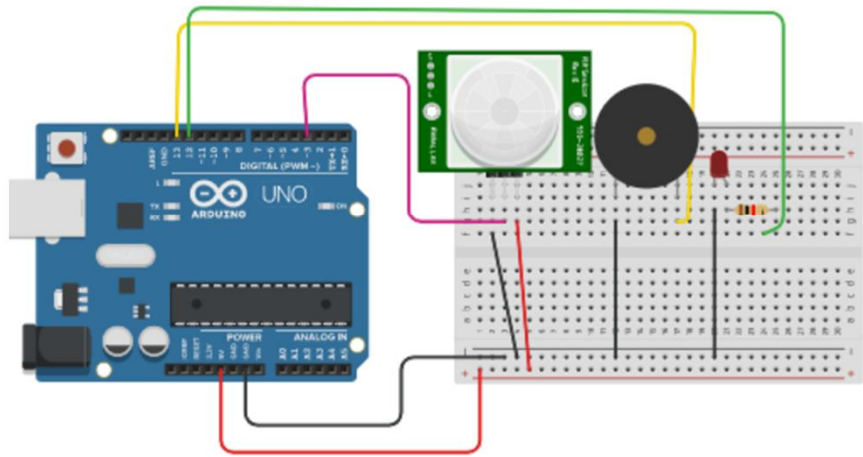
void setup() {
  Serial.begin(115200);
  pinMode(pirPin, INPUT);
}

void loop() {
  din = digitalRead(pirPin);
  Serial.println(din);
  delay(100);
}
```



모션 감지 센서는 감지각도범위 안에 적외선(빛)의 변화가 있을 시 High(1) 신호를, 없을 시 Low(0) 값을 출력함으로 **digitalRead()** 함수를 사용하여 변화를 감지

해결문제



34

해결문제 : 움직임이 감지되었을 때 부저 소리를 내고 LED에 불을 켜보세요.

온도 감지 센서




라이브러리 추가

- 라이브러리

- 특정 목적을 위해 사용되는 파일들을 모아놓은 집합
- 새로운 센서를 쓸 때 우리가 직접 컨트롤 하기에는 너무 복잡하기 때문에 이런 일을 쉽게 할 수 있도록 해주는 라이브러리들을 사용

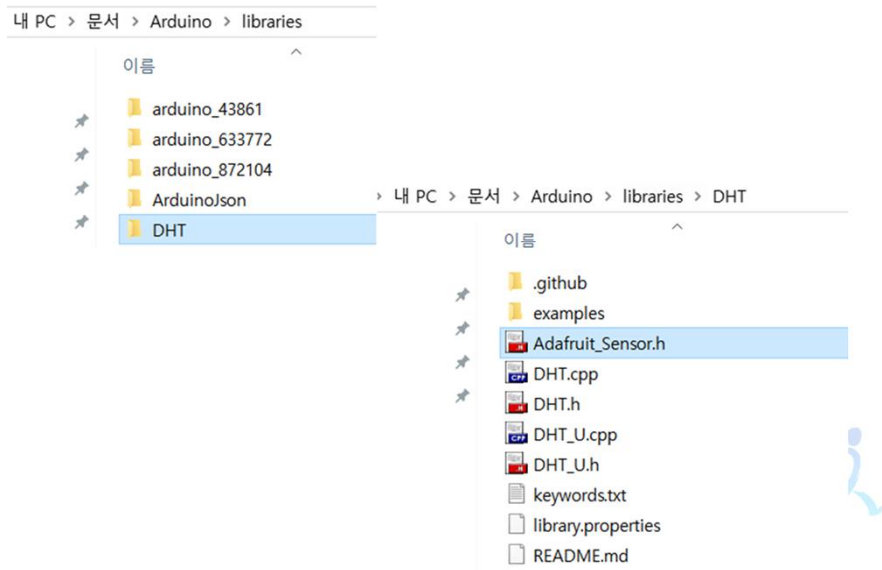
- 라이브러리 설치 방법

- [스케치 - 라이브러리 가져오기] 메뉴로 설치
 - [스케치 - 라이브러리 가져오기 - 라이브러리 추가...] 메뉴를 선택합니다. 그리고 다운로드 받은 라이브러리 ZIP 파일의 위치를 지정
- 직접 설치
 - 아두이노 개발환경을 종료
 - [아두이노 설치폴더\libraries] 폴더로 이동
 - 라이브러리 폴더를 만들어 파일을 넣으면 아두이노 개발환경이 시작할 때 인식
 - 아두이노 개발환경을 실행하고 [스케치 - 라이브러리 가져오기] 메뉴를 확인
 - 예제는 [파일 - 예제] 메뉴를 확인

Come with 

- libraries\ArduinoTest (폴더)
- libraries\ArduinoTest\ArduinoTest.cpp
- libraries\ArduinoTest\ArduinoTest.h
- libraries\ArduinoTest\examples (폴더)

DHT11 라이브러리 추가



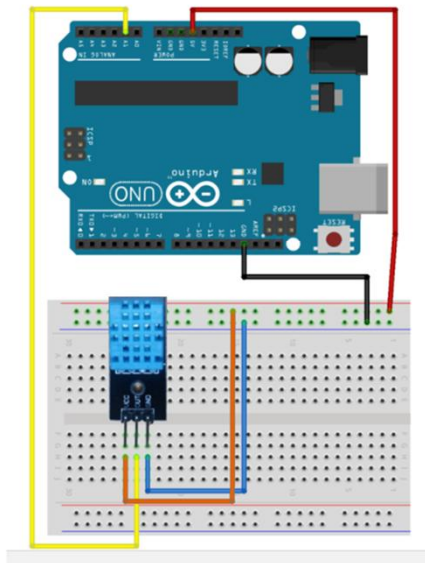
37

[온도감지센서 - DHT11]

- 내부에 써미스터(thermistor)와 정전식 습도 센서가 내장되어있는 온습도 센서
- 측정값을 디지털 센서 신호로 출력

참고) 라이브러리 추가시 폴더명은 DHT가 되어야 하고 DHT 폴더에는 Adafruit_Sensor.h 파일이 반드시 존재

온도 습도 측정



```
#include "DHT.h"
#define DHTTYPE DHT11 // DHT 11

// DHT Sensor
const int DHTPin = 5;
DHT dht(DHTPin, DHTTYPE);

void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  delay(10);

  dht.begin();
}

void loop() {
  // put your main code here, to run repeatedly:
  float h = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float t = dht.readTemperature();

  if (isnan(h) || isnan(t)) {
    Serial.println("Failed to read from DHT sensor!");
  }
  else{
    Serial.print("Humidity: ");
    Serial.print(h);
    Serial.print(" %Wt Temperature: ");
    Serial.print(t);
    Serial.print(" %Wn");
  }
}
```



아두이노(Arduino) WiFi



39

[ESP8266 모듈]

- WiFi 네트워킹을 지원하는 통신 모듈이며, 하드웨어 제어 기능도 상당부분 수행할 수 있음

[WeMos D1 보드]

- ESP-12(E) 모듈을 이용해서 아두이노 보드와 호환되도록 만든 보드
- 아두이노 개발 환경을 이용해서 ESP8266 펌웨어를 만들 수 있음

D1 R1 WiFi 확인

```
#include <ESP8266WiFi.h>

#ifndef STASSID
#define STASSID "your-ssid"
#define STAPSK "your-password"
#endif

const char* ssid = STASSID;
const char* password = STAPSK;

const char* host = "dijmmx.net";
const uint16_t port = 17;

void setup() {
  Serial.begin(115200);

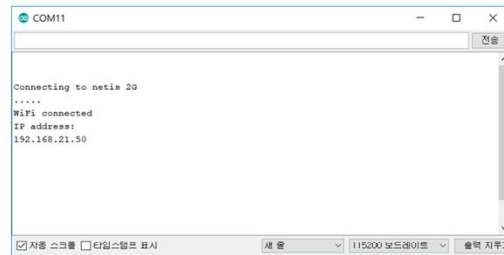
  // We start by connecting to a WiFi network
  Serial.println();
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void loop() {
}
```



아두이노와 파이어베이스 연결

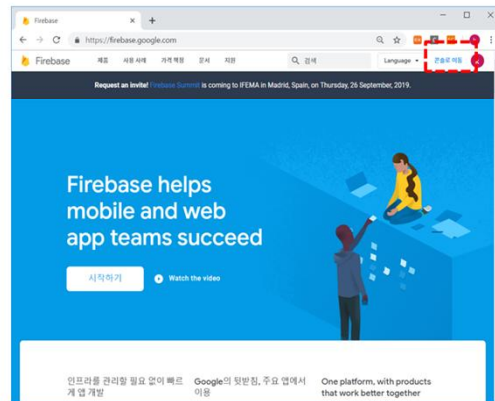


41

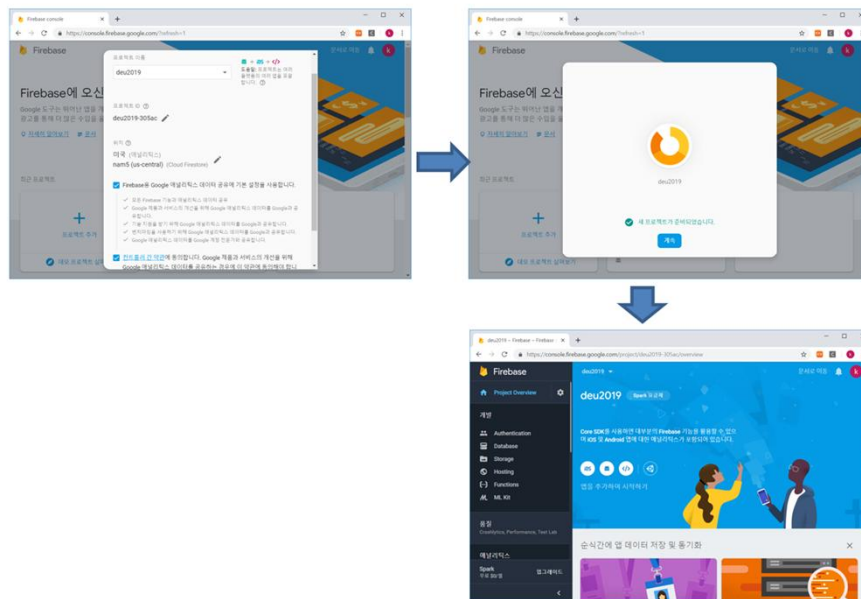
파이어베이스(Firebase)는 2011년 파이어베이스(Firebase, Inc)사가 개발하고 2014년 구글에 인수된 모바일 및 웹 애플리케이션 개발 플랫폼이다.

Firestore 환경설정하기

- <https://firebase.google.com/>
- 구글 계정으로 로그인
- Console로 이동



프로젝트 생성



아두이노와 파이어베이스 연결

- https://github.com/firebase/firebase-arduino/tree/master/examples/FirebaseDemo_ESP8266
 - Download [FirebaseArduino library](#)
 - Start Arduino
 - Click Sketch > Include Library > Add .ZIP Library...
 - Choose firebase-arduino-master.zip downloaded

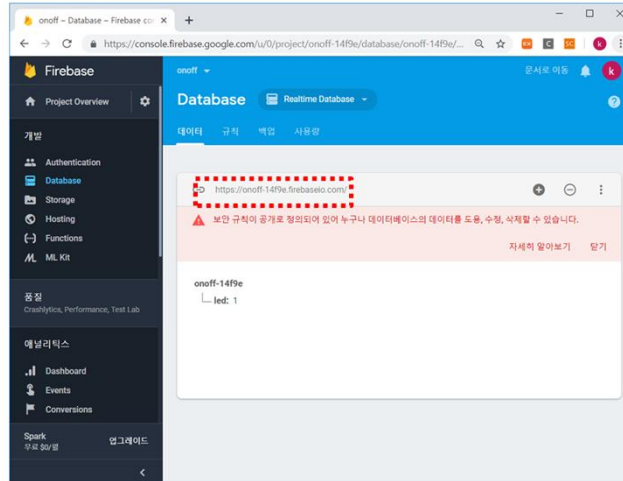


44

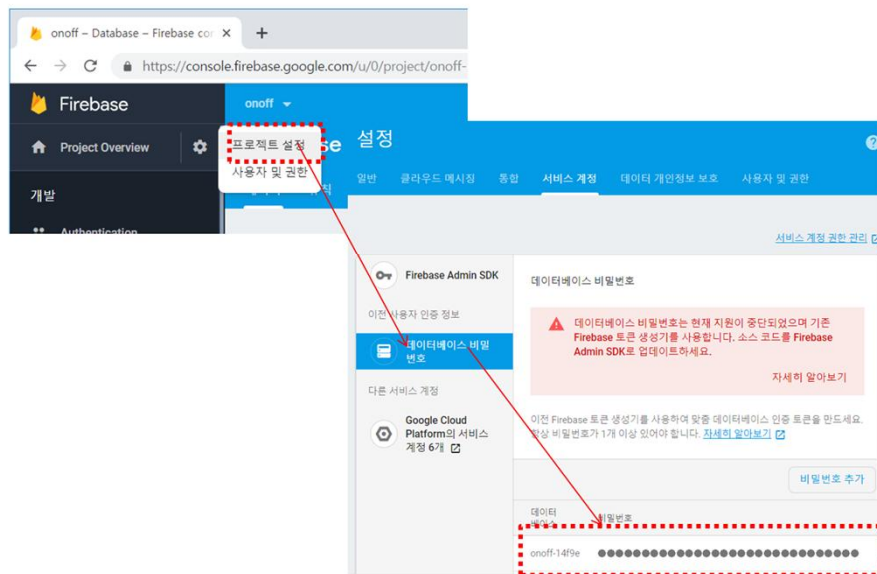
FirebaseArduino now depends on [ArduinoJson library](#) instead of containing its own version of it. Please either use Library Manager or download specific version of the library from github. We recommend that ArduinoJson is at least version [5.13.1](#)

파이어베이스 – Realtime Database

- <https://console.firebase.google.com/>



파이어베이스-데이터베이스 비번



ESP8266 WiFi 설정 및 firebase 설정

```
#include <ESP8266WiFi.h>
#include <FirebaseArduino.h>

// Set these to run example.
#define FIREBASE_HOST ""
#define FIREBASE_AUTH ""
#define WIFI_SSID ""
#define WIFI_PASSWORD "11110000"

void setup() {
  Serial.begin(115200);

  // connect to wifi.
  WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
  Serial.print("connecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(500);
  }
  Serial.println();
  Serial.print("connected: ");
  Serial.println(WiFi.localIP());

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
}
```

```
float n = 0;

void loop() {
  // set value
  Firebase.setFloat("temp", n);
  // handle error
  if (Firebase.failed()) {
    Serial.print("setting /number failed:");
    Serial.println(Firebase.error());
    return;
  }
  n = n + 0.1 ;
  delay(1000);
}
```

onoff-14f9e

led: 1

temp: 18.1000328

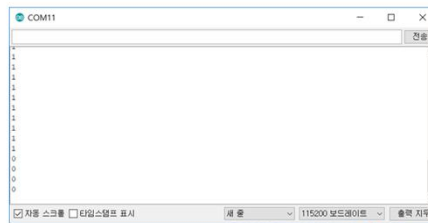


해결문제

```
int ledck ;
void loop() {
  ledck = Firebase.getInt("led") ;

  if ( ledck == 1 ) {
    digitalWrite(ledPin, HIGH);
  }
  else {
    digitalWrite(ledPin, LOW);
  }

  Serial.println(ledck);
  delay(1000);
}
```



해결문제 : 파이어베이스의 led값에 따라 LED의 불을 끄고 켜보세요.