

**Robot Control System**

# 오픈소스 하드웨어 개요

Open Source Hardware (OSHW)

- Arduino      - Raspberry Pi

강사 양석환



# 오픈소스 하드웨어 (OSHW)

- **OSHW (Open Source HardWare)**

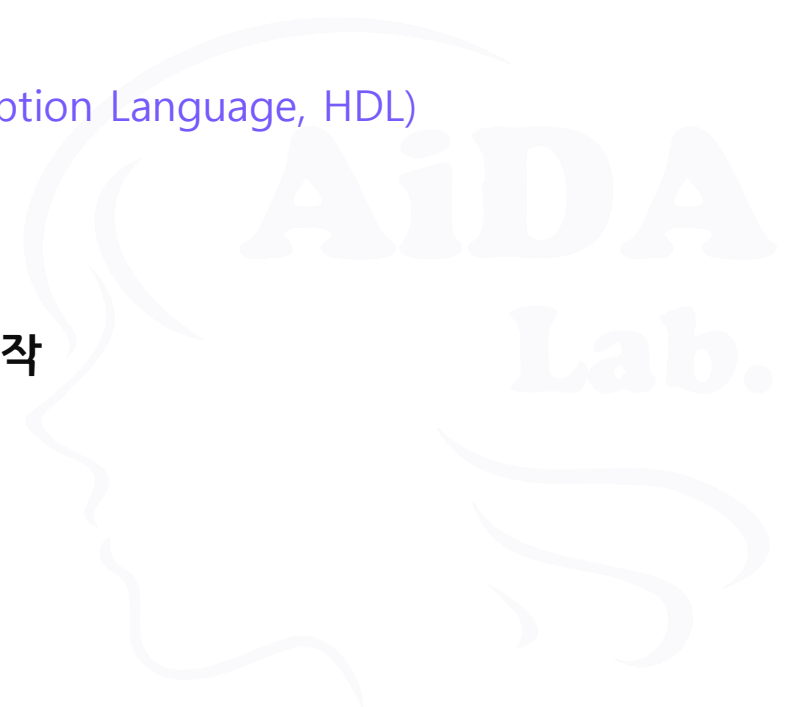
- 해당 제품과 똑같은 모양 및 기능을 가진 제품을 만드는 데 필요한 모든 것(회로도, 자재 명세서, 인쇄 회로 기판 도면 등)을 대중에게 공개한 전자제품

- **하드웨어 기술 언어**가 대중에게 공개된 프로그래머블 논리 소자

전자회로를 정밀하게 기술하는 데 사용하는 컴퓨터 언어 (Hardware Description Language, HDL)

- **대표적인 OSHW**

- 아두이노(2005~), 라즈베리파이(2012~), 젯슨나노(2019~) 및 아류작



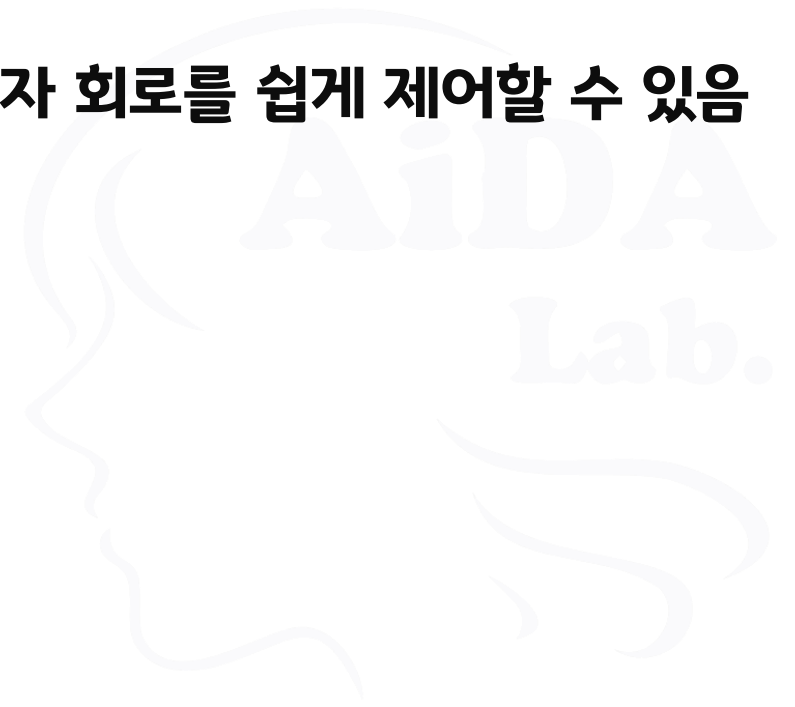
사진출처: Google 이미지 검색 결과

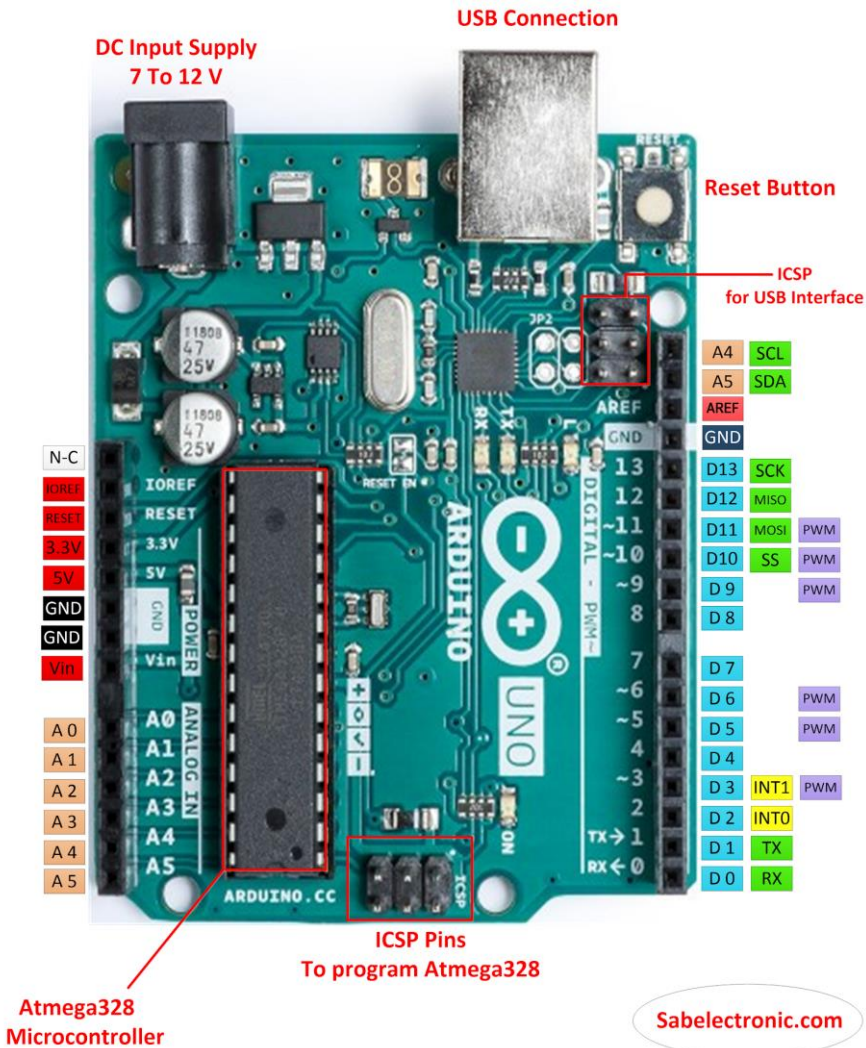
내용 출처: 테크월드(<http://www.epnc.co.kr/news/articleView.html?idxno=97412>) 에서 참조 후 편집

종류	등장시기	운영체제	가격대	접근성	특징
아두이노 	2005~	X	저가	높음	하드웨어 센서, 모듈 제어
라즈베리파이 	2012~	Linux 계열 (기본: 라즈비안)	저가	높음	범용성, 애플리케이션 개발
라떼판다 	2015~	Windows 10	고가	보통	윈도우 기반 IoT 시스템 개발 아두이노를 기반으로 고성능화
젯슨나노 	2019~	Linux 계열 (기본: Ubuntu 18.04 LTS 기반 NVIDIA 자체 커스터마이징)	중가	보통	AI 기반 애플리케이션 개발

**Arduino**

- **마이크로 컴퓨터 탑재 → 프로그램을 실행할 수 있는 디바이스**
  - Single Board Computer(SBC)는 아님
- **처리 능력이 PC보다 낮아서 GUI 프로그램 등을 실행할 수 없음**
- **LED, 스위치, 센서 등을 제어하는 기능이 내장되어 있어서 전자 회로를 쉽게 제어할 수 있음**

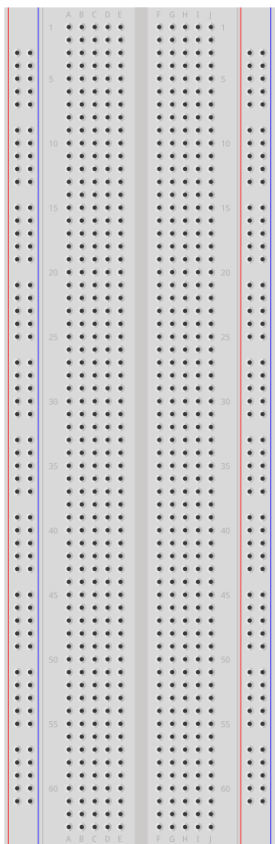




PIN	설명
N-C	사용하지 않음
IOREF	입출력 참조 핀 (아두이노에 장착해서 사용하는 실드에서 아두이노 보드에 입력된 전압을 알기 위한 용도로 사용)
RESET	프로그램의 모든 동작을 멈추고 업로드한 프로그램이 처음부터 다시 시작함
3.3V	전압 공급
5V	전압 공급
GND	접지역할 (회로의 가장 낮은 전압)
Vin	전원 입력 (외부 전원을 꽂아 전원 공급에 사용)
A0~A5	아날로그 입력 핀(센서의 아날로그 신호를 아두이노의 프로세서가 사용할 수 있도록 디지털 값으로 변환해 줌)
AREF	Analog Reference 아날로그 입력 핀의 상한선을 외부 참조 전압(0~5V사이)으로 설정할 때 사용함
D0~D13	디지털 입/출력 핀 (0 / 1)

PIN	설명
SCL	I2C 통신용 (A4와 공동사용)
SDA	I2C 통신용 (A5와 공동사용)
SCK	SPI 통신용 (D13과 공동사용)
MISO	SPI 통신용 (D12와 공동사용)
MOSI	SPI 통신용 (D11과 공동사용)
SS	SPI 통신용 (D10과 공동사용)
INT1	인터럽트용 핀 (D3과 공동사용)
INT0	인터럽트용 핀 (D2와 공동사용)
TX	시리얼 통신용 Transmission 포트 (D1과 공동사용)
RX	시리얼 통신용 Receiver 포트 (D0과 공동사용)
PWM	펄스 폭 변조 (Pulse Width Modulation) 디지털 값을 아날로그처럼 만들어 줌 (D3, D5, D6, D9, D10, D11과 공동 사용)

## • 사용 부품, 도구



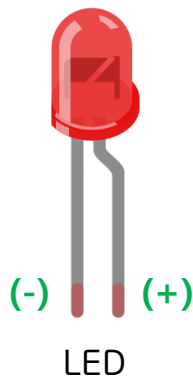
Breadboard  
(브레드보드)  
(방판)



Breadboard 내부



220Ω 저항



LED



Jumper Wire (F-M)

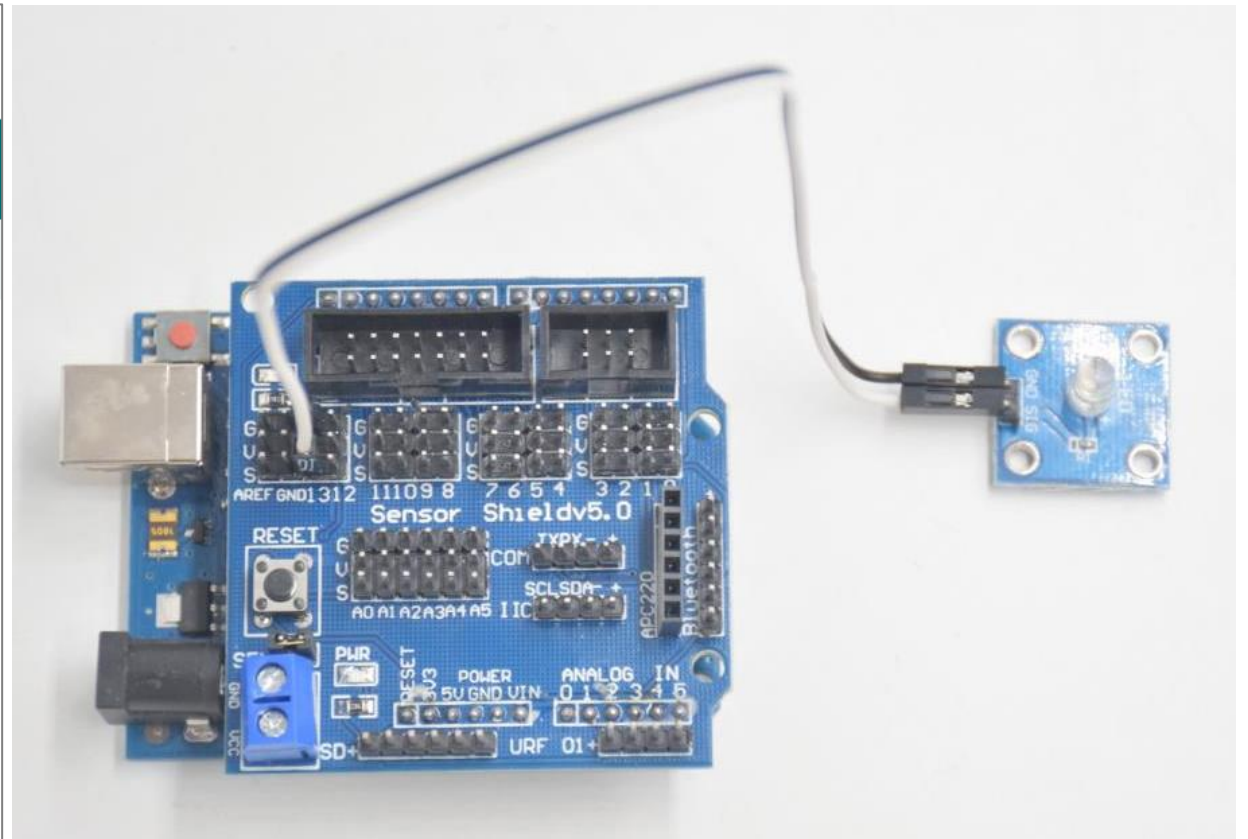


**F**emale : 암단자, F타입



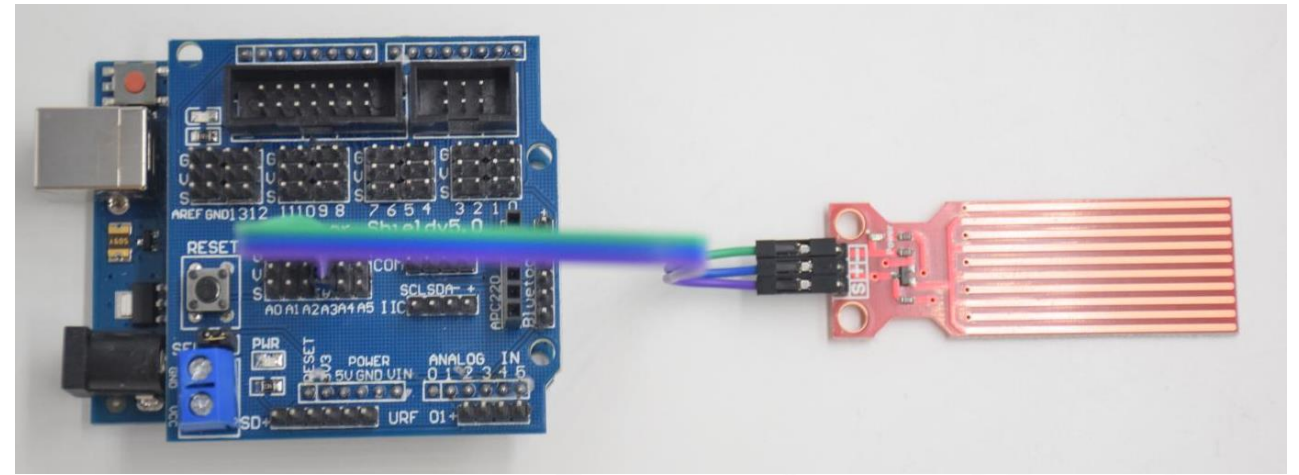
**M**ale : 수단자, M타입

```
One_LED | Arduino IDE 2.0.2
파일(F) 편집 스케치 도구 도움말
Arduino Uno
One_LED.ino
1 int LED_PIN = 13;
2
3 void setup() {
4   pinMode(LED_PIN, OUTPUT);
5 }
6
7 void loop() {
8   digitalWrite(LED_PIN, HIGH);
9   delay(1000);
10
11  digitalWrite(LED_PIN, LOW);
12  delay(1000);
13 }
14
```

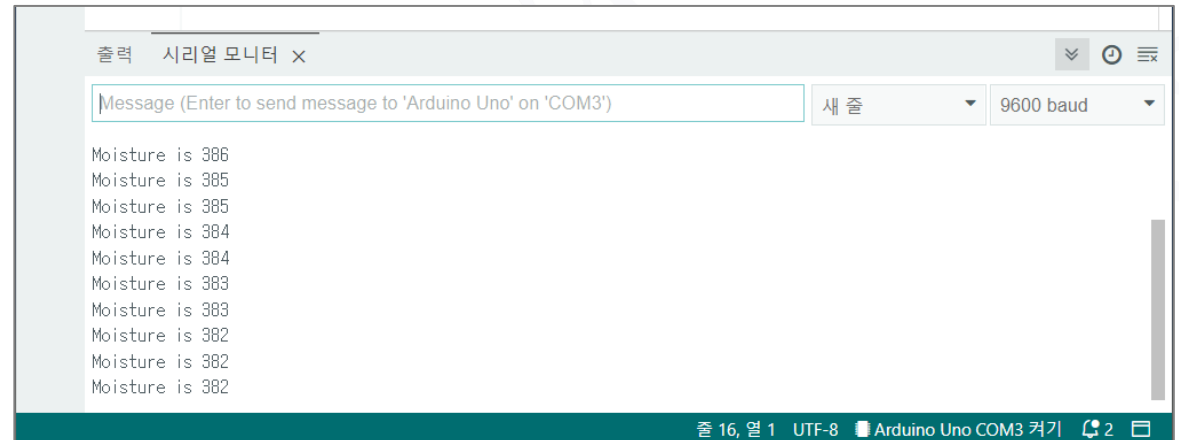


핀 연결:  
SIG → D13 / GND → GND

```
Steam_Sensor | Arduino IDE 2.0.2
파일(F) 편집 스케치 도구 도움말
보드 선택
Steam_Sensor.ino
1 int SteamPin = 3;
2
3 void setup()
4 {
5   Serial.begin(9600);
6 }
7
8 void loop()
9 {
10  int val;
11  val=analogRead(SteamPin);
12  Serial.print("Moisture is ");
13  Serial.println(val, DEC);
14  delay(1000);
15 }
16
```

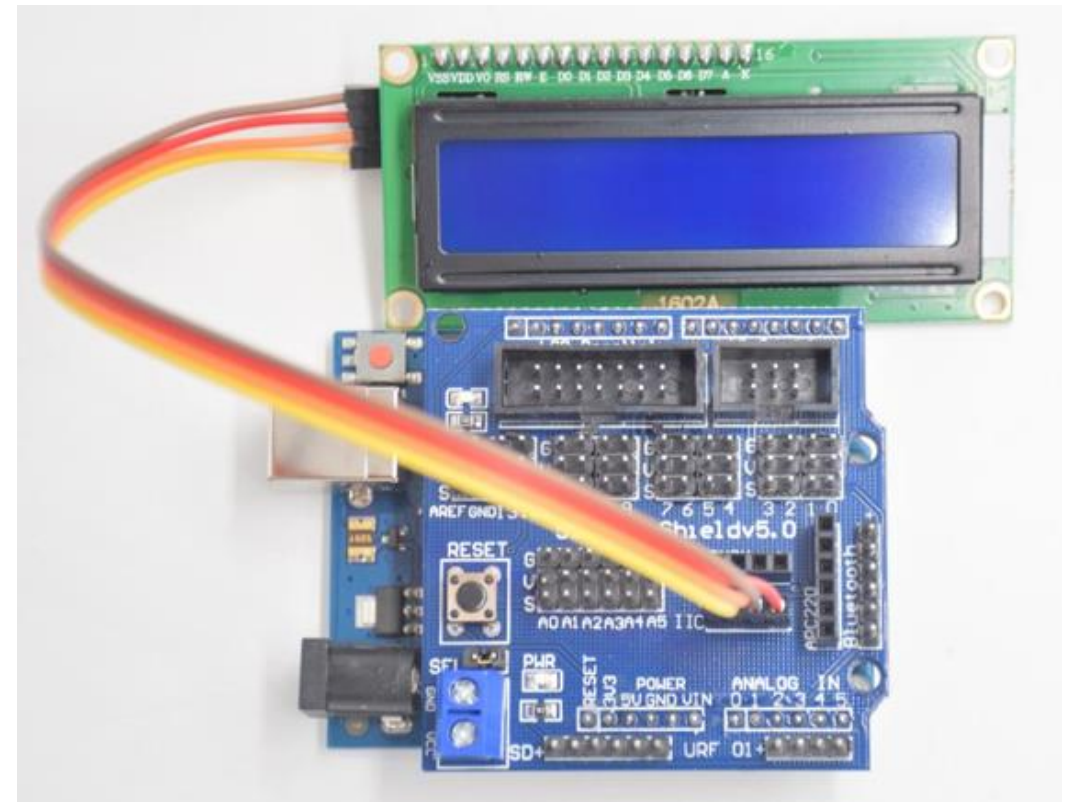


핀 연결:  
S → D3 / (+) → 연결안함 / (-) → GND



# 1602 LCD Display (1)

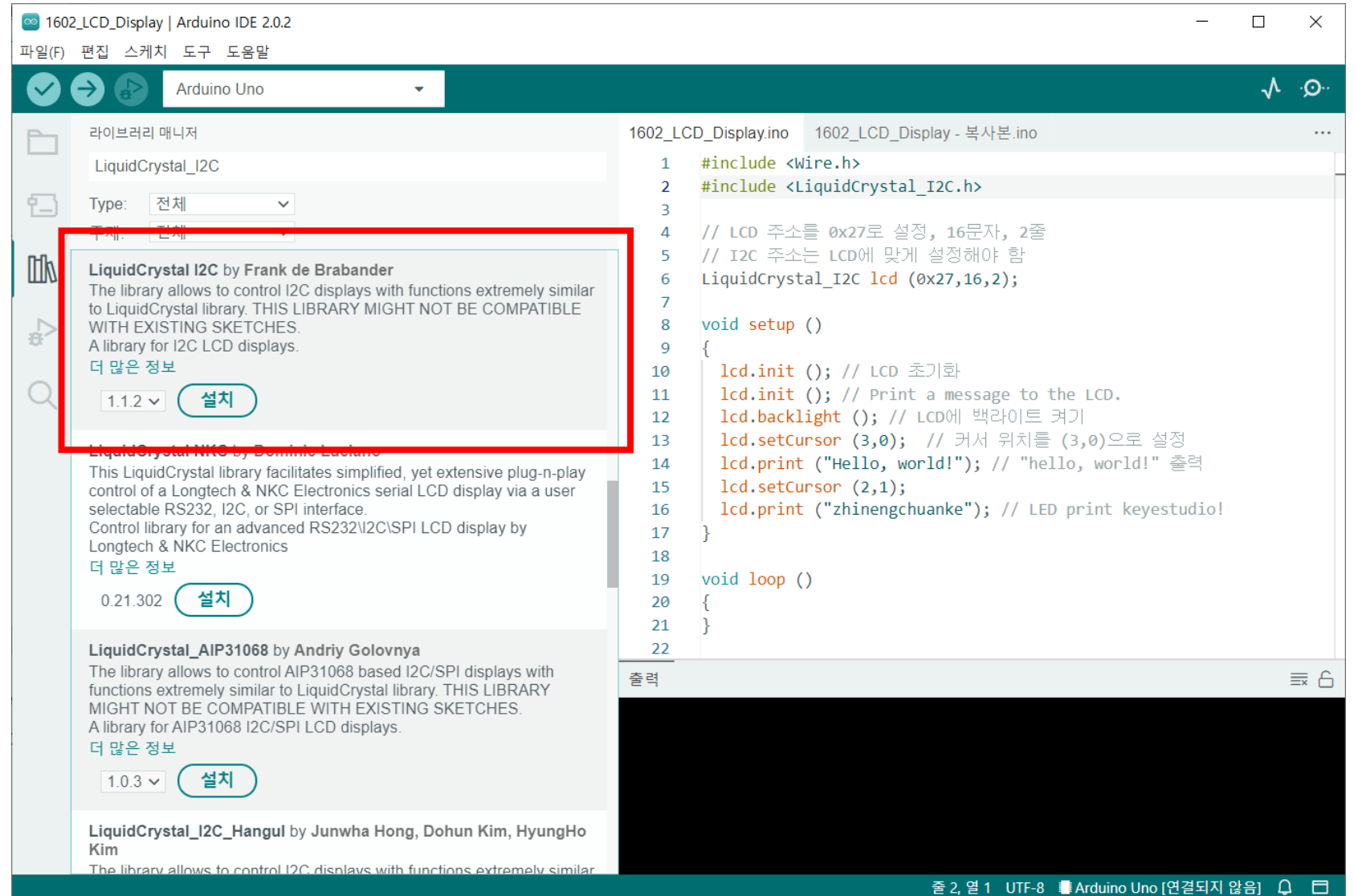
```
1602_LCD_Display | Arduino IDE 2.0.2
파일(F) 편집 스케치 도구 도움말
Arduino Uno
1602_LCD_Display.ino
1 #include <Wire.h>
2 #include <LiquidCrystal_I2C.h>
3
4 // LCD 주소를 0x27로 설정, 16문자, 2줄
5 // I2C 주소는 LCD에 맞게 설정해야 함
6 LiquidCrystal_I2C lcd (0x27,16,2);
7
8 void setup ()
9 {
10  lcd.init (); // LCD 초기화
11  lcd.init (); // Print a message to the LCD.
12  lcd.backlight (); // LCD에 백라이트 켜기
13  lcd.setCursor (3,0); // 커서 위치를 (3,0)으로 설정
14  lcd.print ("Hello, world!"); // "hello, world!" 출력
15  lcd.setCursor (2,1);
16  lcd.print ("yang seokhwan"); // LED print keystudio!
17 }
18
19 void loop ()
20 {
21 }
22
출력
스케치는 프로그램 저장 공간 3296 바이트(10%)를 사용. 최대 32256 바이트.
전역 변수는 동적 메모리 285바이트(13%)를 사용, 1763바이트의 지역변수가 남음. 최대는 2048 바이트.
출 16, 열 20 UTF-8 Arduino Uno COM3 켜기 2
```



핀 연결:  
SCL → SCL / SDA → SDA / VCC → (+) / GND → (-)

“메뉴>스케치>라이브러리 관리”에서  
필요한 라이브러리를 검색한 후 설치

사용 부품은 1602 LCD Display 중  
I2C 통신 타입이므로 LiquidCrystal\_I2C  
라이브러리를 설치함

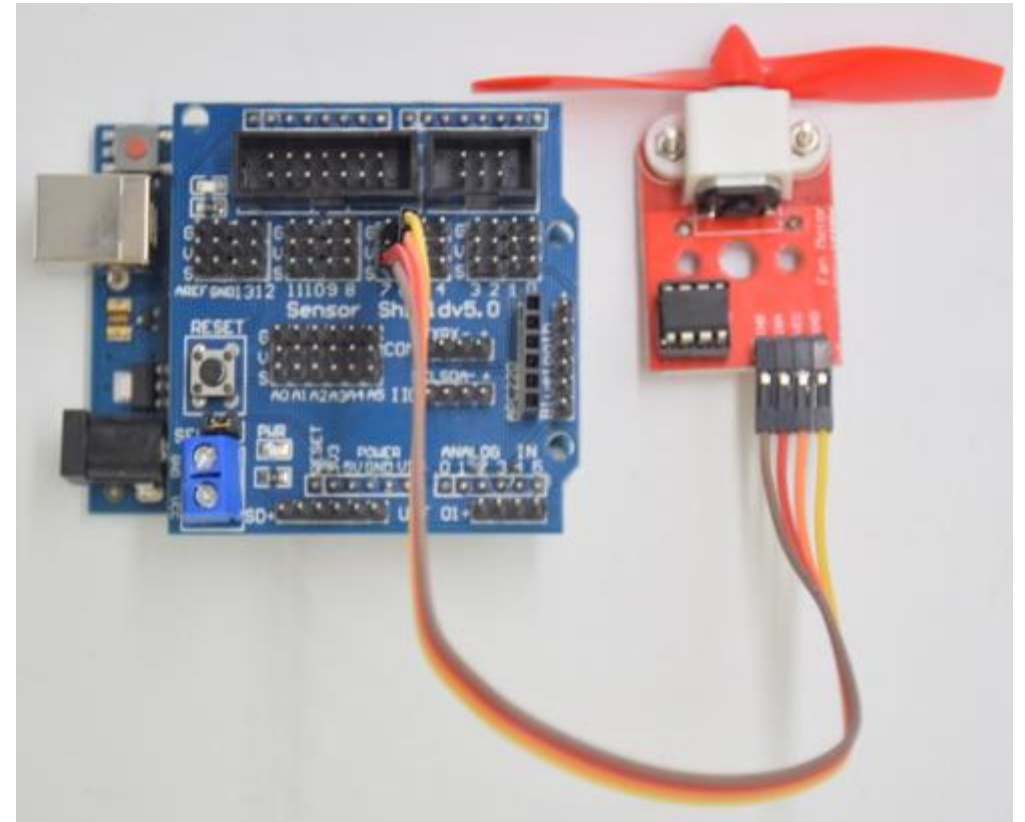


```
Servo_Motor | Arduino IDE 2.0.2
파일(F) 편집 스케치 도구 도움말
Arduino Uno
Servo_Motor.ino
1 #include <Servo.h>
2
3 Servo myservo;
4 int pos = 0; // 서보모터 시작각도
5
6 void setup ()
7 {
8   myservo.attach(9); // D9에 서보모터 접속
9   myservo.write(90);
10  delay(1000);
11 }
12 void loop ()
13 {
14   myservo.write(0);
15
16   for(pos = 0; pos < 60; pos += 10)// 0~180도
17   {
18     myservo.write(pos); // 서보모터 각도 위치
19     delay(100);
20   }
21 }
22
출력
스케치는 프로그램 저장 공간 2168 바이트(6%)를 사용. 최대 32256 바이트.
전역 변수는 동적 메모리 52바이트(2%)를 사용, 1996바이트의 지역변수가 남음. 최대는 2048 바이트.
줄 22, 열 1 UTF-8 Arduino Uno [연결되지 않음] 2
```



핀 연결:  
Yellow → D9 / Red → 5V / Brown → GND

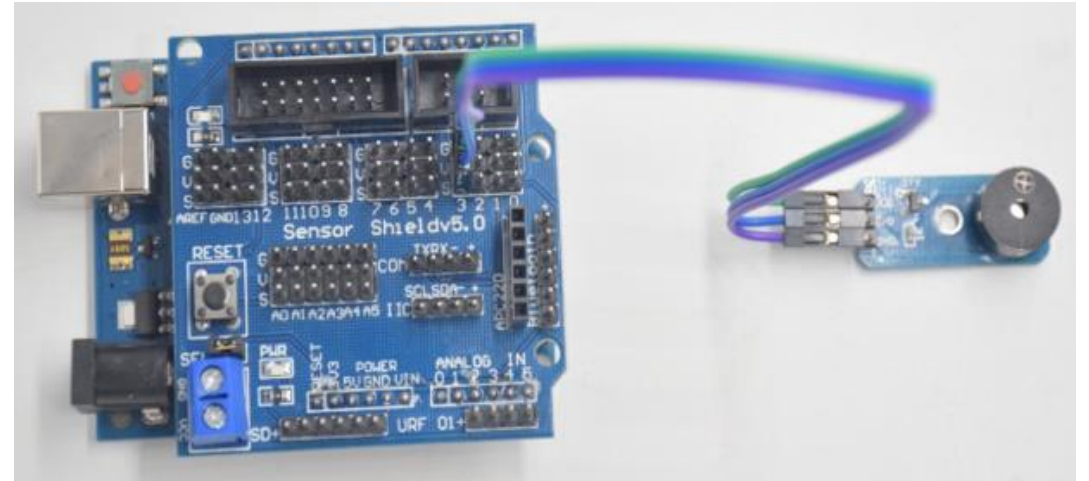
```
Fan | Arduino IDE 2.0.2
파일(F) 편집 스케치 도구 도움말
Arduino Uno
Fan.ino
1 void setup () {
2   pinMode (7, OUTPUT); //define D7 pin as output
3   pinMode (6, OUTPUT); //define D6 pin as output
4 }
5 void loop () {
6   digitalWrite (7, LOW);
7   digitalWrite (6, HIGH); // Reverse rotation of the motor
8   delay (3000); // delay 3S
9   digitalWrite (7, LOW);
10  digitalWrite (6, LOW); // The motor stops rotating
11  delay (1000); //delay 1S
12  digitalWrite (7, HIGH);
13  digitalWrite (6, LOW); // The motor rotates in the forward direction
14  delay (3000); // delay 3S
15 }
16
출력
스케치는 프로그램 저장 공간 1008 바이트(3%)를 사용. 최대 32256 바이트.
줄 2, 열 24 UTF-8 Arduino Uno [연결되지 않음] 2
```



핀 연결:

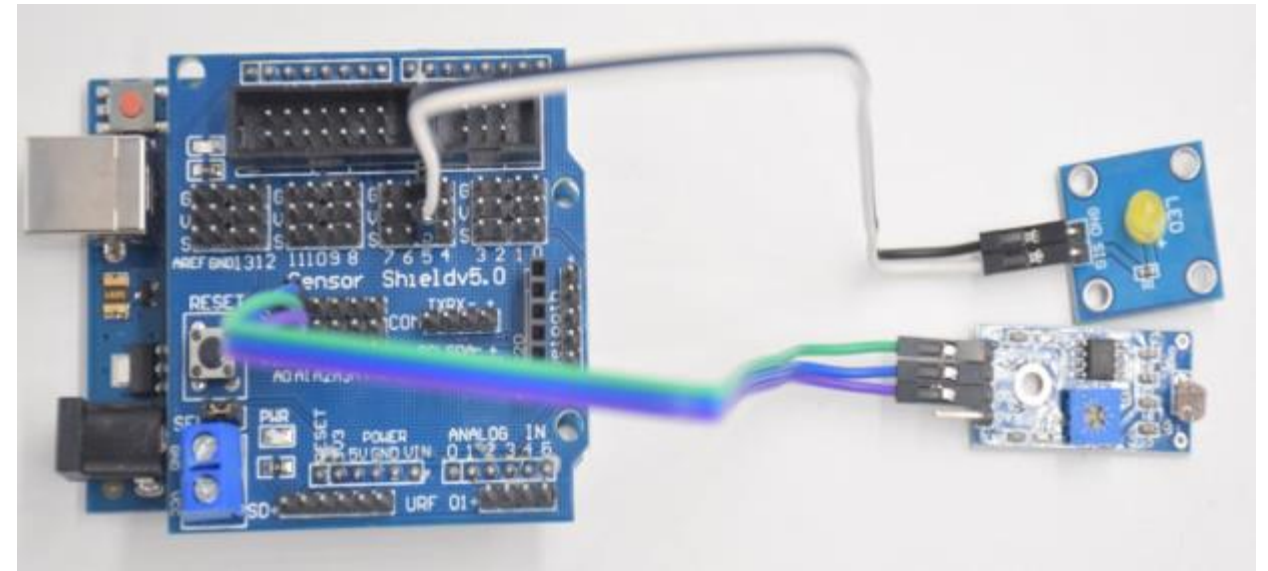
INB → D6 / INA → D7 / VCC → 5V / GND → GND

```
Passive_Buzzer | Arduino IDE 2.0.2
파일(F) 편집 스케치 도구 도움말
Arduino Uno
Passive_Buzzer.ino
7 void loop() {
8   unsigned char i, j;
9   while (1) {
10    for(i=0; i<80; i++) {
11      digitalWrite(TonePin, HIGH);
12      delay(1);
13
14      digitalWrite(TonePin, LOW);
15      delay(1);
16    }
17    for (i=0; i<100; i++) {
18      digitalWrite(TonePin, HIGH);
19      delay(2);
20
21      digitalWrite(TonePin, LOW);
22      delay(2);
23    }
24  }
25 }
26
출력
스케치는 프로그램 저장 공간 984 바이트(3%)를 사용. 최대 32256 바이트.
전역 변수는 동적 메모리 9바이트(0%)를 사용, 2039바이트의 지역변수가 남음. 최대는 2048 바이트.
줄 20, 열 1 UTF-8 Arduino Uno [연결되지 않음]
```



핀 연결:  
VCC → 5V / I/O → D3 / GND → GND

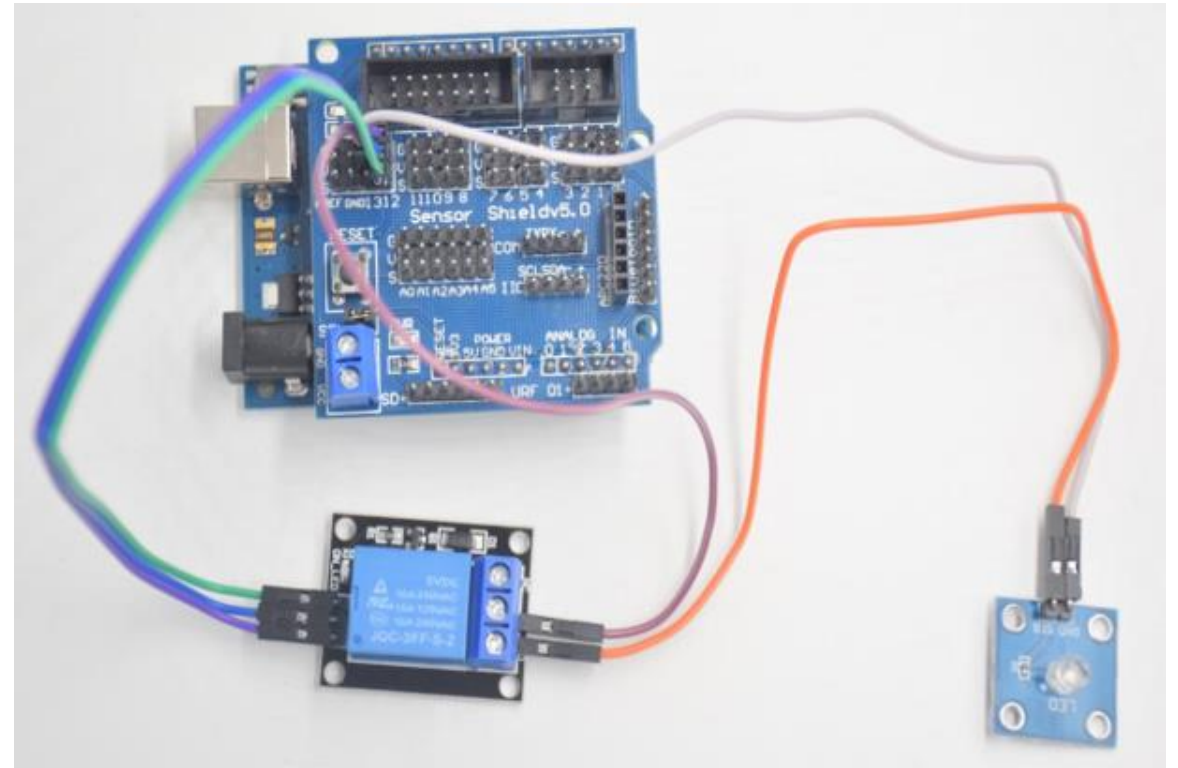
```
PhotoCell | Arduino IDE 2.0.2
파일(F) 편집 스케치 도구 도움말
Arduino Uno
PhotoCell.ino
1 int LED = 5;
2 int val = 0; // 포토다이오드의 전압 값
3
4 void setup () {
5   pinMode (LED, OUTPUT);
6   Serial.begin (9600); // 시리얼 포트의 전송속도는 9600으로 설정
7 }
8
9 void loop () {
10  val = analogRead (A1); // A1의 전압값을 읽어서 val 변수로 입력
11  Serial.println (val); // 전압 값의 변화를 보기 위한 시리얼포트 출력
12
13  // 전압 값이 900보다 작으면 LED를 끄고 900 이상이면 LED 켜기
14  if (val < 900) {
15    digitalWrite (LED, LOW);
16  } else {
17    digitalWrite (LED, HIGH);
18  }
19  delay (1000);
20 }
21
출력 시리얼 모니터 x
Message (Enter to send message to 'Arduino Uno' on 'COM3) 새 줄 9600 baud
1023
29
28
28
줄 19, 열 16 UTF-8 Arduino Uno COM3 켜기 2
```



핀 연결(Photocell):  
VCC → 5V / GND → GND /  
D0 → A1 / A0 → No Answer(연결하지 않음)

핀 연결(LED):  
SIG → D5 / GND → GND

```
Channel_Relay | Arduino IDE 2.0.2
파일(F) 편집 스케치 도구 도움말
Arduino Uno
Channel_Relay.ino
1 int Relay = 12;
2 int LedPin = 13;
3
4 void setup ()
5 {
6   pinMode (LedPin, OUTPUT);
7   digitalWrite (LedPin, HIGH);
8   pinMode (Relay, OUTPUT);
9 }
10 void loop ()
11 {
12   digitalWrite (Relay, HIGH);
13   delay (2000);
14   digitalWrite (Relay, LOW);
15   delay (2000);
16 }
17
출력
스케치는 프로그램 저장 공간 964 바이트(2%)를 사용. 최대 32256 바이트.
전역 변수는 동적 메모리 9바이트(0%)를 사용, 2039바이트의 지역변수가 남음. 최대는
```

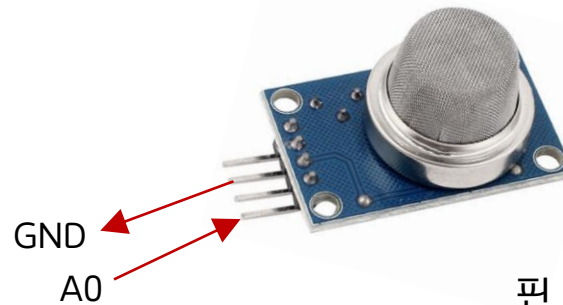
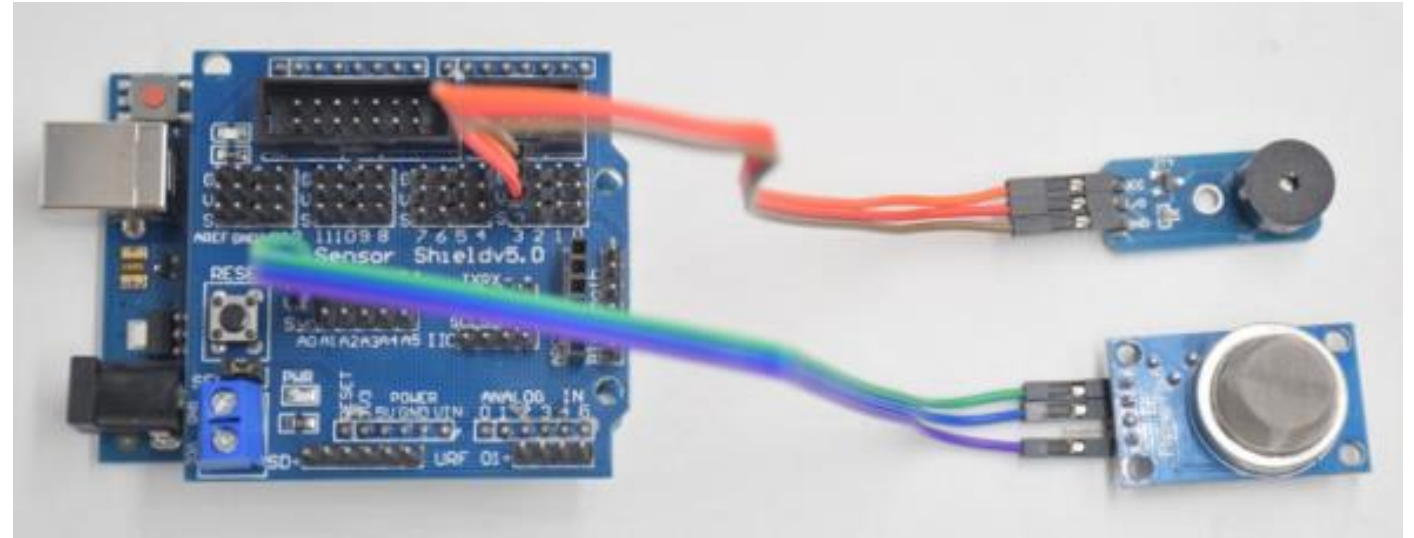


핀 연결(Relay):  
S → D12 / (+) → 5V / (-) → GND / COM → 5V

핀 연결(LED):  
SIG → NO 또는 NC / GND → COM

# MQ2 Gas Sensor (Analog)

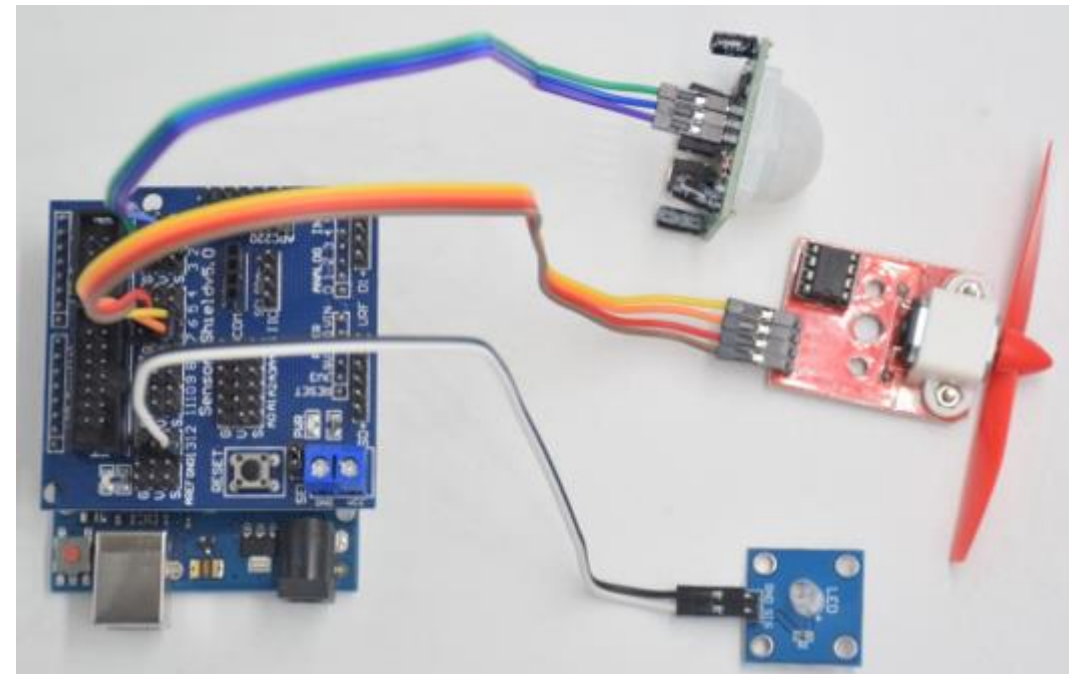
```
Analog_MQ2_Sensor | Arduino IDE 2.0.2
파일(F) 편집 스케치 도구 도움말
Arduino Uno
Analog_MQ2_Sensor.ino
1 int MQ2 = A0;
2 int val = 0;
3 int buzzer = 3;
4
5 void setup ()
6 {
7   pinMode (MQ2, INPUT);
8   Serial.begin (9600);
9   pinMode (buzzer, OUTPUT);
10 }
11 void loop ()
12 {
13   val = analogRead (MQ2);
14   Serial.println (val);
15   if (val > 400)
16   {
17     tone (buzzer, 589);
18     delay(300);
19   }
20   else
21   {
22     noTone (buzzer);
23   }
24 }
25
출력 시리얼 모니터 x
Message (Enter to send message to 'Arduino Uno' on 'COM3' 새 줄 9600 baud
0
0
0
0
0
0
줄 4, 열 1 UTF-8 Arduino Uno COM3 커기 2
```



핀 연결(MQ2):  
VCC → 5V / A0 → A0 / GND → GND

핀 연결(Passive Buzzer):  
VCC → 5V / I/O → D3 / GND → GND

```
PIR_Motion_Sensor | Arduino IDE 2.0.2
파일(F) 편집 스케치 도구 도움말
보드 선택
PIR_Motion_Sensor.ino
1 void setup () {
2   Serial.begin (9600); // 시리얼 포트 전송속도는 9600으로 설정
3   pinMode (2, INPUT); // PIR 입력은 2
4   Serial.begin (9600);
5   pinMode (13, OUTPUT); // Define LED as output in D13
6   pinMode (7, OUTPUT); // Define D7 as output
7   pinMode (6, OUTPUT); // Define D6 as output
8 }
9
10 void loop () {
11   Serial.println (digitalRead (2));
12   delay (500); // Delay 500ms
13   if (digitalRead (2) == 1) // If someone is detected walking
14   {
15     digitalWrite (13, HIGH); // LED light is on
16     digitalWrite (7, HIGH);
17     analogWrite (6, 150); // Fan rotates
18   } else // If no person is detected walking
19   {
20     digitalWrite (13, LOW); // LED light is not on
21     digitalWrite (7, LOW);
22     analogWrite (6, 0); // The fan does not rotate
23   }
24 }
25 }
26
줄 3, 열 38 UTF-8 × 선택된 보드 없음
```



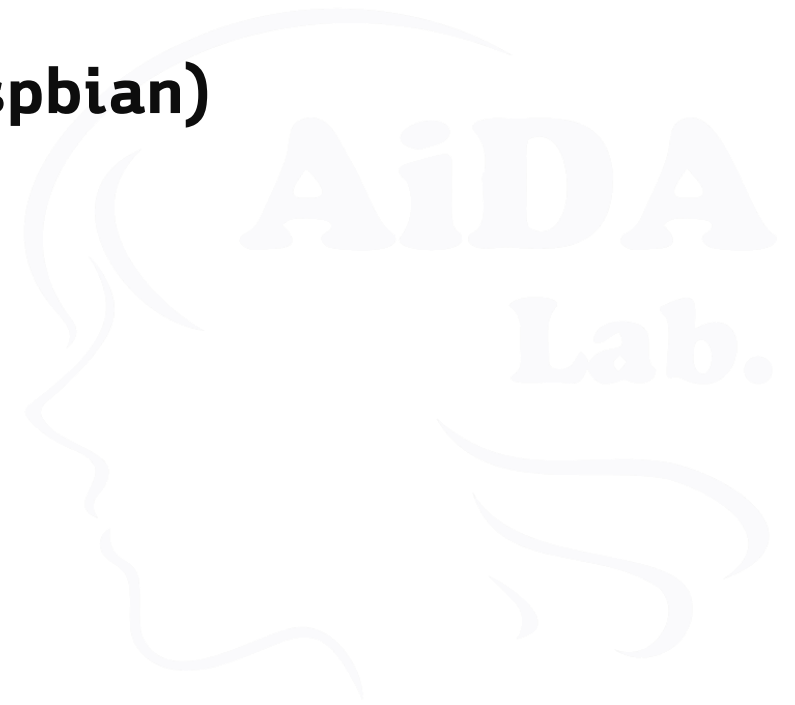
핀 연결(PIR Motion Sensor):  
VCC → 5V / OUT → AD2 / GND → GND

핀 연결(Fan):  
INB → D6 / INA → D7 / VCC → 5V / GND → GND

핀 연결(LED):  
SIG → D13 / GND → GND

# 라즈베리파이 개요

- 대표적인 Single Board Computer(SBC)
  - OS를 설치한 SD/Micro SD 카드로 부팅해서 PC로 사용 가능
- GPIO를 이용하여 전자회로 등을 직접 제어할 수 있음
- 주로 리눅스 계열의 OS 사용. 권장하는 OS는 라즈비안(Raspbian)
  - 리눅스 기반 OS를 사용하므로 파이썬 기본 설치
  - 파이썬을 이용하여 GUI, 대규모 프로그램을 개발 가능
  - 전자회로 제어를 기반으로 한 다양한 응용 시스템 개발 가능





Raspberry Pi Zero W



Raspberry Pi Zero



Model A



Model A+



Pi2 Model B



Model B



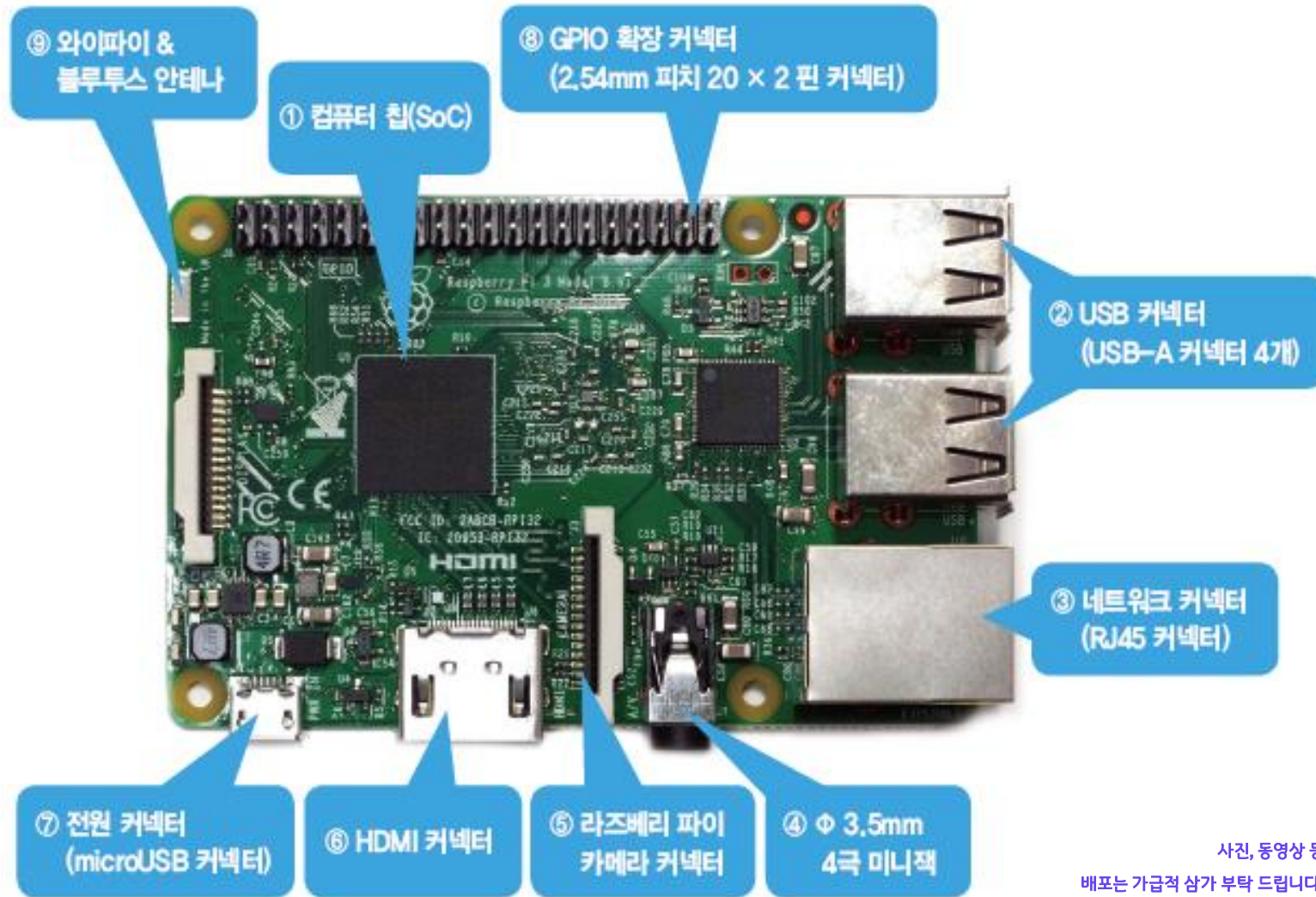
Model B+



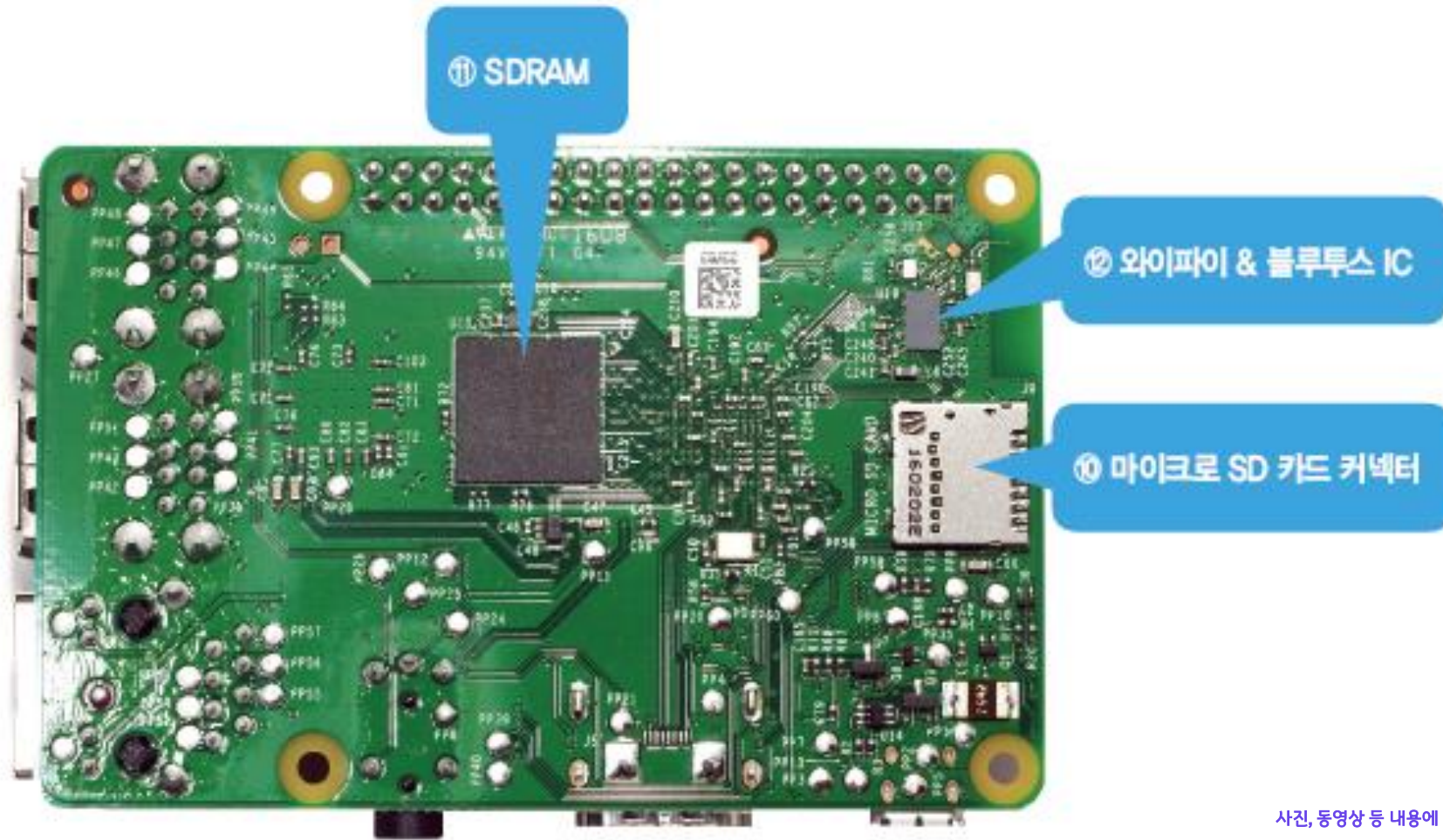
Pi3 Model B



사진, 동영상 등 내용에 저작권이 있을 수 있습니다.  
배포는 가급적 삼가 부탁드립니다. 학습 용도로만 활용 부탁드립니다.



사진, 동영상 등 내용에 저작권이 있을 수 있습니다.  
배포는 가급적 삼가 부탁드립니다. 학습 용도로만 활용 부탁드립니다.



사진, 동영상 등 내용에 저작권이 있을 수 있습니다.  
배포는 가급적 삼가 부탁드립니다. 학습 용도로만 활용 부탁드립니다.

- **컴퓨터 칩 SoC(System On Chip)**
  - 라즈베리 파이의 심장부
  - 이 칩에 ARM 코어 프로세서, GPU, 인터페이스 등 다양한 기능이 내장되어 있음
- **USB 커넥터**
  - USB-A 타입 커넥터를 4개 탑재하고 있음 (USB 2.0 지원)
- **네트워크 커넥터**
  - LAN 케이블을 사용할 수 있는 RJ45 커넥터 (10/100Mbps 지원)



## • $\Phi$ 3.5mm 4극 미니잭

- 4극 타입인  $\Phi$  3.5mm 미니잭에 콤포지트 비디오 신호와 아날로그 스테레오 오디오 신호를 출력하는 커넥터
- 콤포지트 비디오 신호를 입력할 수 있는 구형 브라운관 TV 등에 연결하려면 핀 플러그로 변환하는 케이블을 사용하면 됨



- **라즈베리 파이 카메라 커넥터**

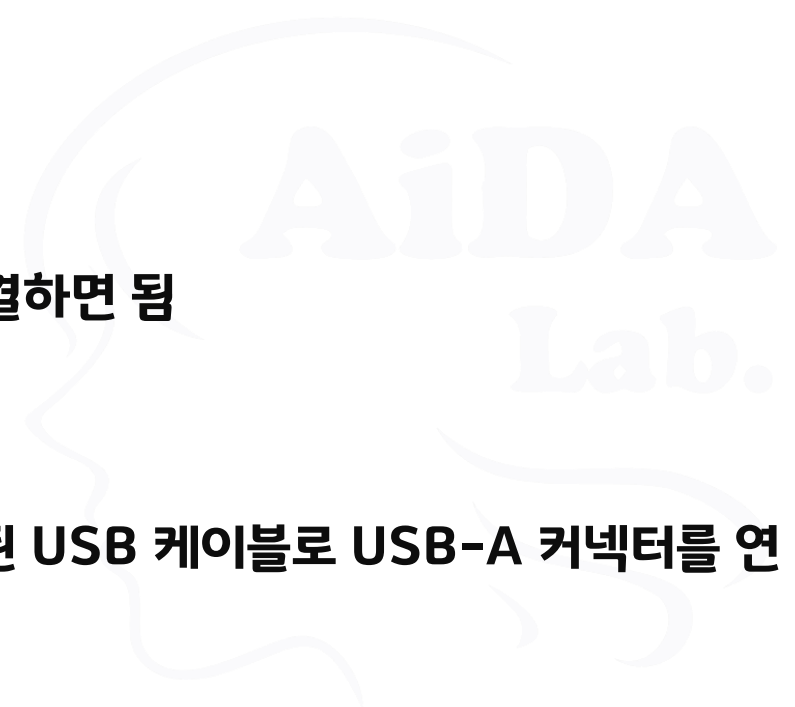
- 라즈베리 파이의 순정 부품인 카메라 모듈을 연결하는 전용 커넥터
- 일반 카메라 모듈 뿐만아니라 적외선 카메라 모듈인 Pi NoIR Camera도 연결 가능

- **HDMI 커넥터**

- TV나 모니터를 연결하는 커넥터
- 음성 신호도 출력할 수 있으므로 TV나 스피커가 내장된 모니터에 연결하면 됨

- **전원 커넥터**

- USB-microB 커넥터 (USB-microB와 USB-A 커넥터 쌍으로 된 USB 케이블로 USB-A 커넥터를 연결할 수 있는 AC 어댑터와 연결)



- **GPIO 확장 커넥터**
  - 프로세서(SoC) 포트에 직접 연결하는 핀 타입 커넥터임
  - LED나 스위치 등 전자 부품을 연결할 수 있음
- **와이파이 & 블루투스 안테나**
- **와이파이 & 블루투스 IC**
- **마이크로 SD 카드 커넥터**
  - OS를 탑재한 마이크로 SD 카드를 삽입함
  - 라즈베리 파이에서는 마이크로 SD 카드가 PC의 하드디스크가 됨
- **1GB SDRAM**





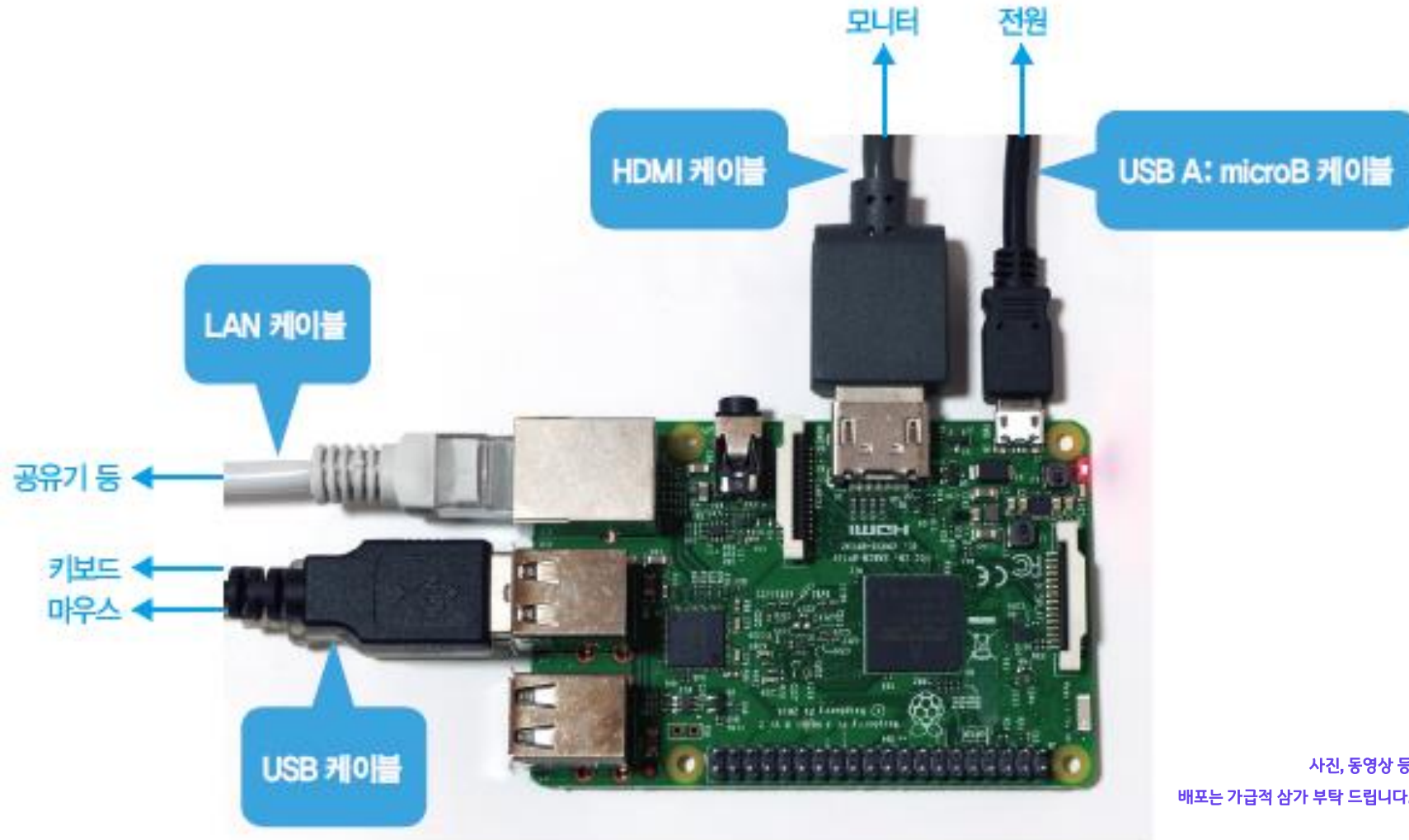
Alternate Function					Alternate Function
	3.3V PWR	1		2	5V PWR
I2C1 SDA	GPIO 2	3		4	5V PWR
I2C1 SCL	GPIO 3	5		6	GND
	GPIO 4	7		8	UART0 TX
	GND	9		10	UART0 RX
	GPIO 17	11		12	GPIO 18
	GPIO 27	13		14	GND
	GPIO 22	15		16	GPIO 23
	3.3V PWR	17		18	GPIO 24
SPI0 MOSI	GPIO 10	19		20	GND
SPI0 MISO	GPIO 9	21		22	GPIO 25
SPI0 SCLK	GPIO 11	23		24	GPIO 8
	GND	25		26	GPIO 7
	Reserved	27		28	Reserved
	GPIO 5	29		30	GND
	GPIO 6	31		32	GPIO 12
	GPIO 13	33		34	GND
SPI1 MISO	GPIO 19	35		36	GPIO 16
	GPIO 26	37		38	GPIO 20
	GND	39		40	GPIO 21
					SPI0 CS0
					SPI0 CS1
					SPI1 CS0
					SPI1 MOSI
					SPI1 SCLK

## GPIO

General Purpose Input/Output  
(범용 목적 입출력)

## RPi의 핀번호 참조방식

- GPIO.BOARD: RPi의 보드에 적혀있는 pin 번호 참조
- BCM: GPIO에 할당된 번호 참조  
예) pin 22 = GPIO25



DA  
Lab.

사진, 동영상 등 내용에 저작권이 있을 수 있습니다.  
배포는 가급적 삼가 부탁드립니다. 학습 용도로만 활용 부탁드립니다.

- 기판이 노출되어 있어서 외부 충격에 약하며 물에 닿으면 금방 고장남
- 컴퓨터 칩(SoC)에 직접 연결된 GPIO 확장 커넥터 핀도 돌출되어 있어서 정전기 등 과도한 전압이 걸리면 SoC가 파손될 수 있음
- 보드를 집어 들다가 손가락에 마이크로 SD 카드가 걸려서 빠져버리거나 하면 마이크로 SD 카드에 접근할 수 없어서 운영체제가 망가지거나 작성한 파일이 저장되지 않는 문제 등이 생길 수 있음
- 이런 사고를 막기 위해 라즈베리 파이용 케이스를 씌우기도 함

- 일반적인 개발 순서

- 시스템 초기화 → 개발 환경 구축 → 개발 → 테스트 → 완성

- Raspberry Pi 시스템 개발 작업 순서

1. 먼저 회로를 구성한다.
2. 파이썬 코드를 작성한다.
3. 파이썬 코드를 실행한다.



- 일단은 초기 상태에서 바로 GPIO를 제어해 보자

- 작업 순서

- 먼저 회로를 구성한다.
- 파이썬 코드를 작성한다.
- 파이썬 코드를 실행한다.

- GPIO 제어 내용

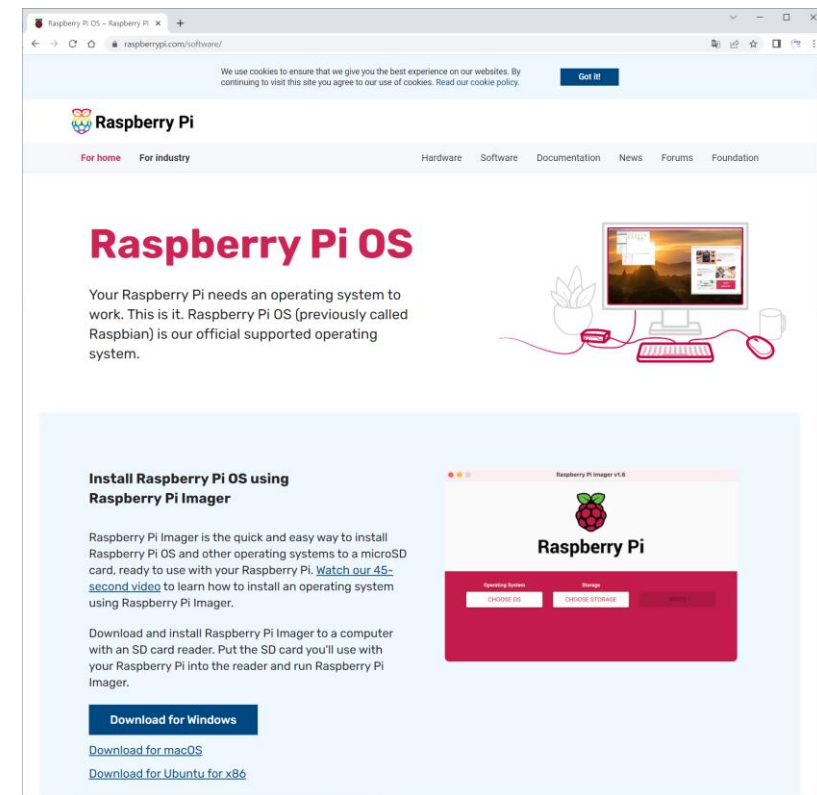
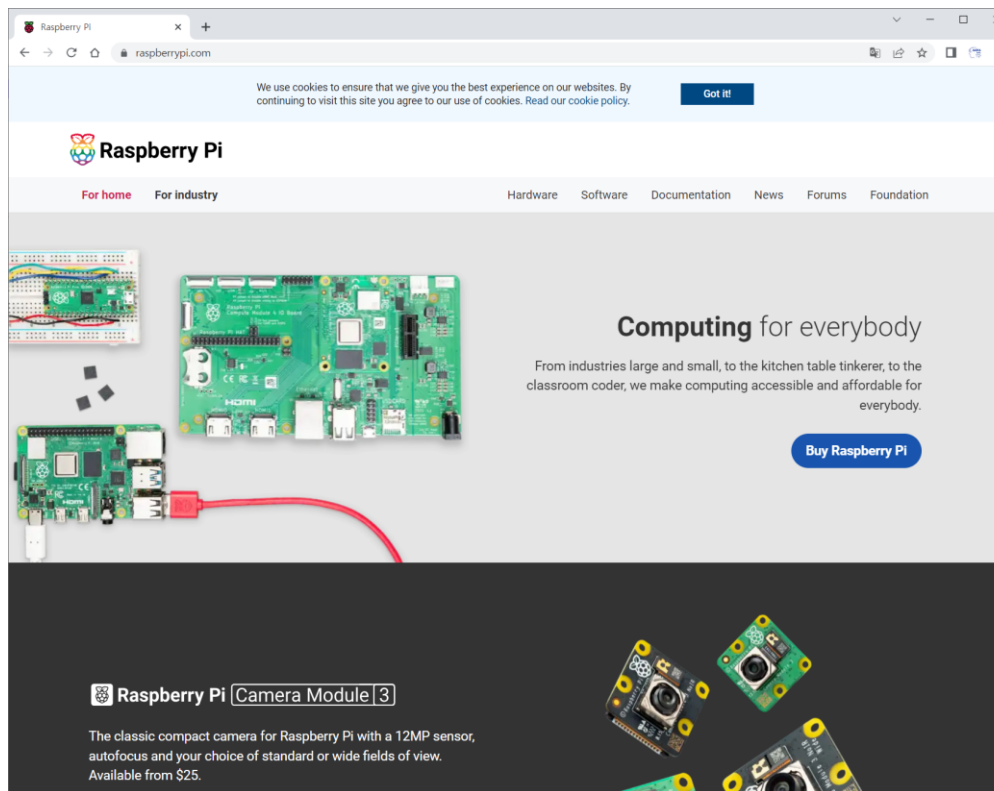
- LED에 불을 켜 보자



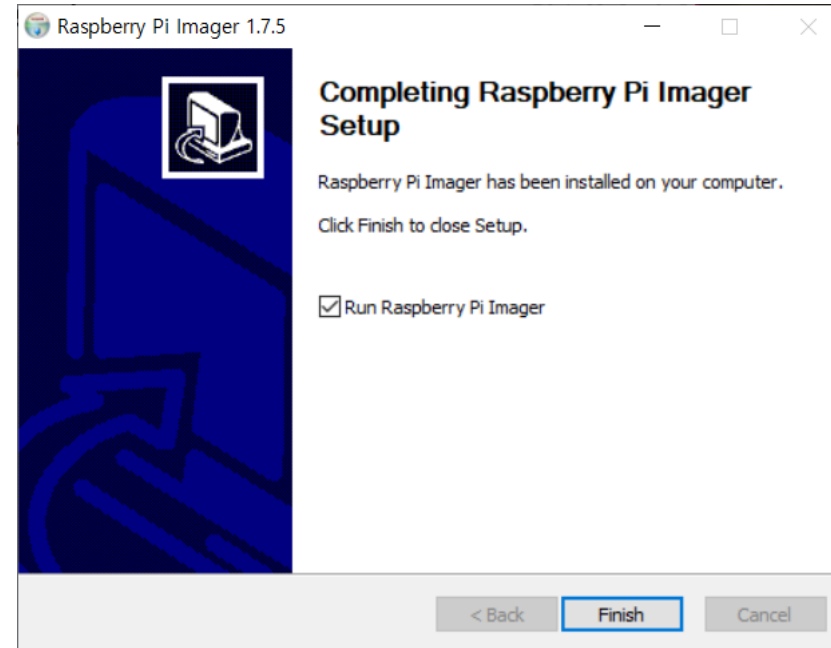
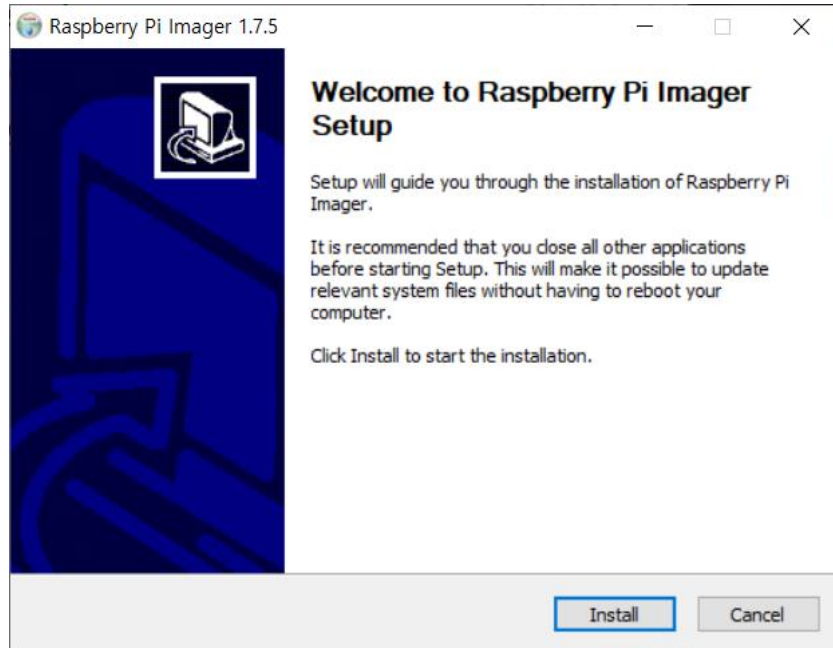
# 라즈베리파이 환경 설정

- <https://www.raspberrypi.com/software/>

- 먼저 Raspberry Pi Imager를 설치해야 하므로 본인이 사용하는 PC의 OS용을 다운로드 하여 설치

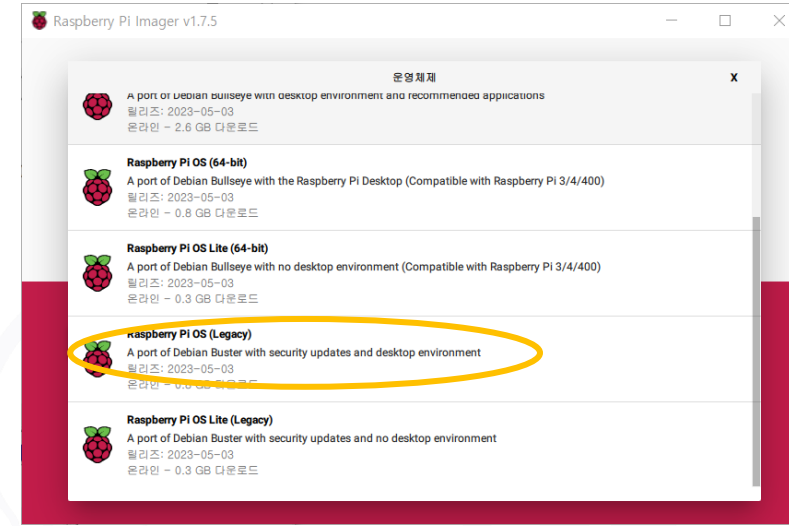
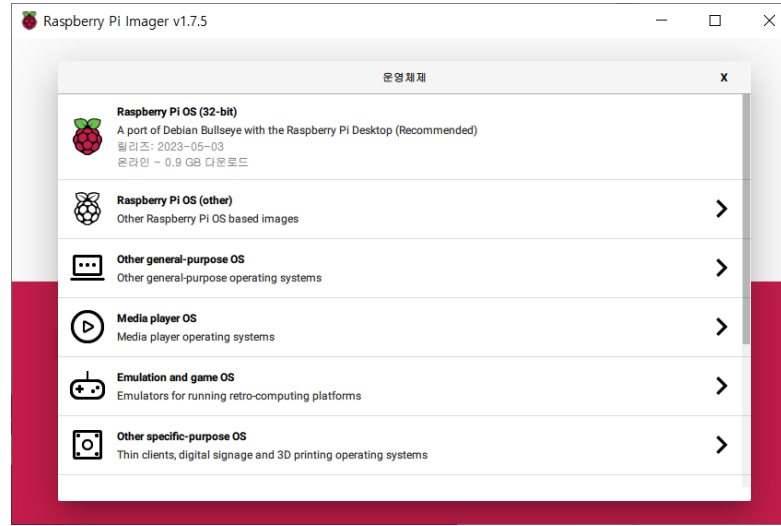


- Raspberry Pi Imager 설치

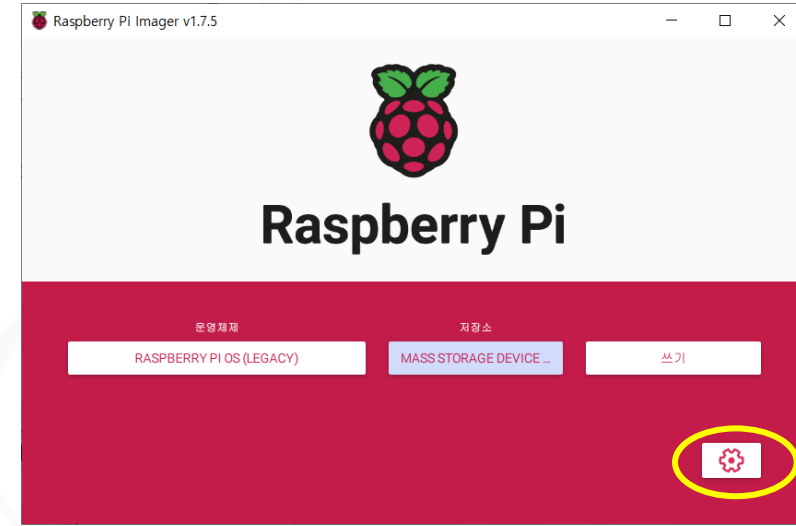
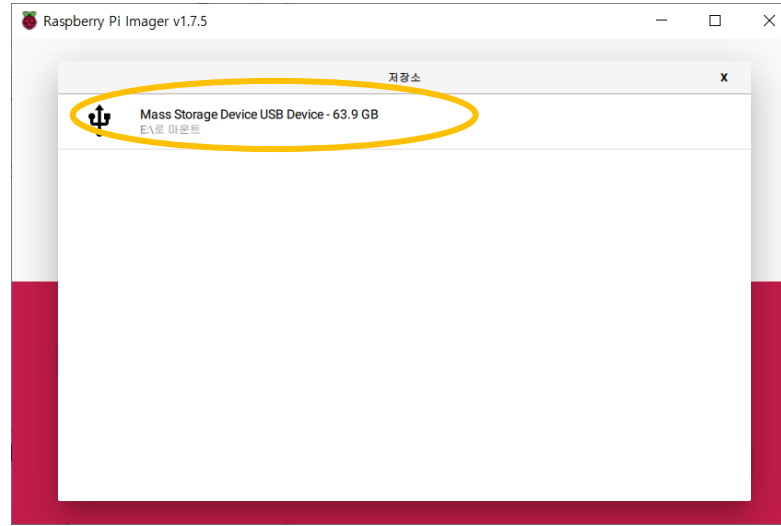


DA  
Lab.

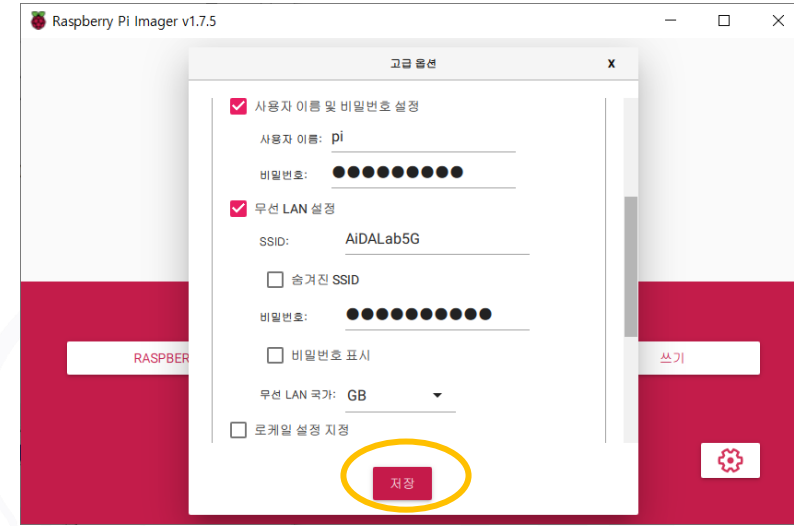
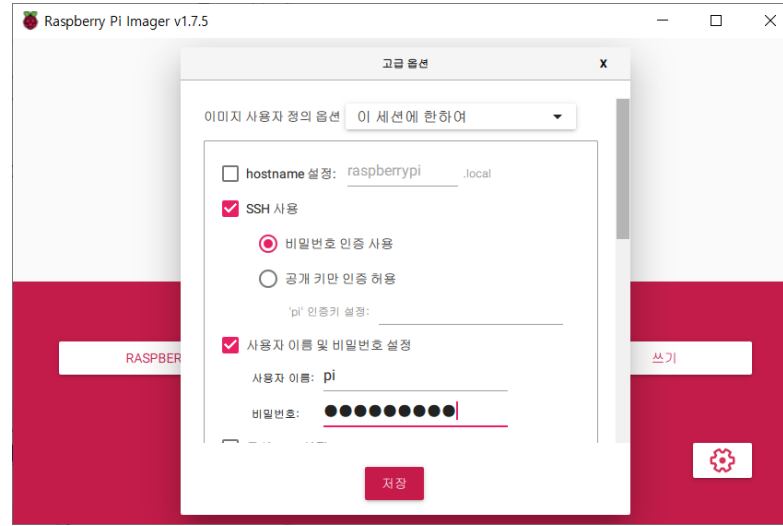
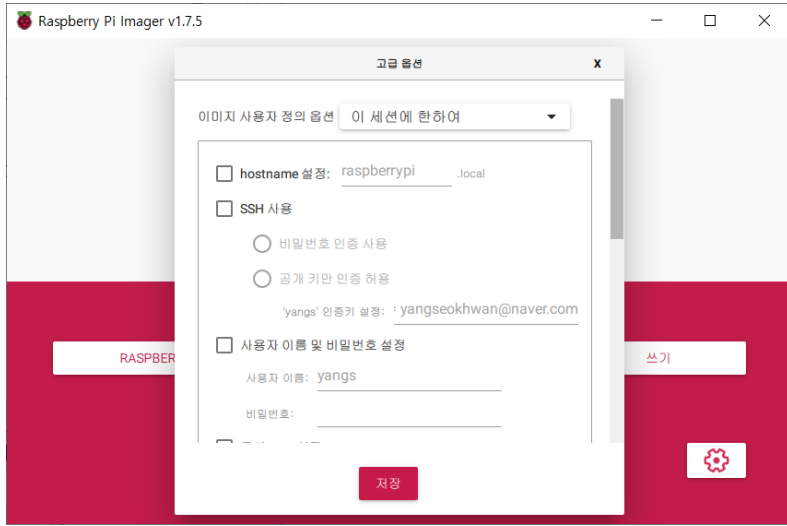
- SD Card/Micro SD Card를 PC에 연결한 후 Raspberry Pi Imager 실행



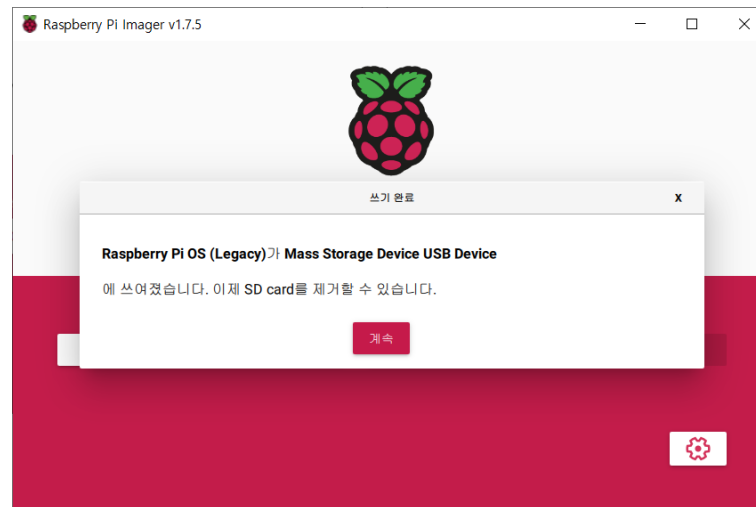
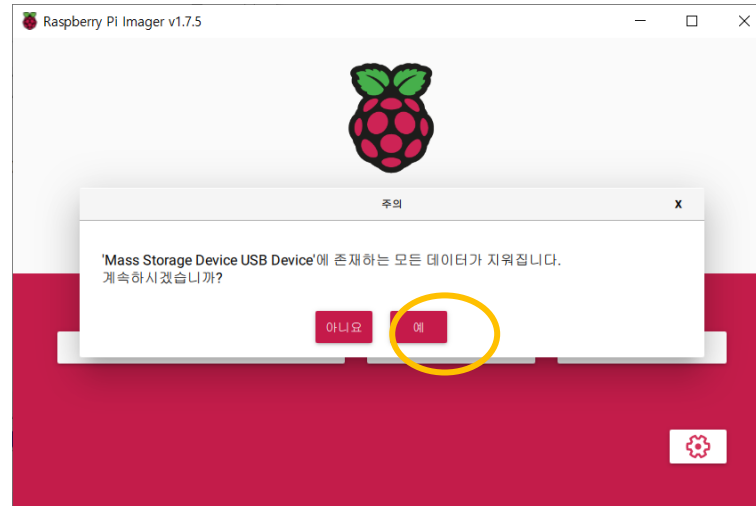
- OS와 SD Card를 선택한 후 톱니바퀴(설정) 버튼을 누름



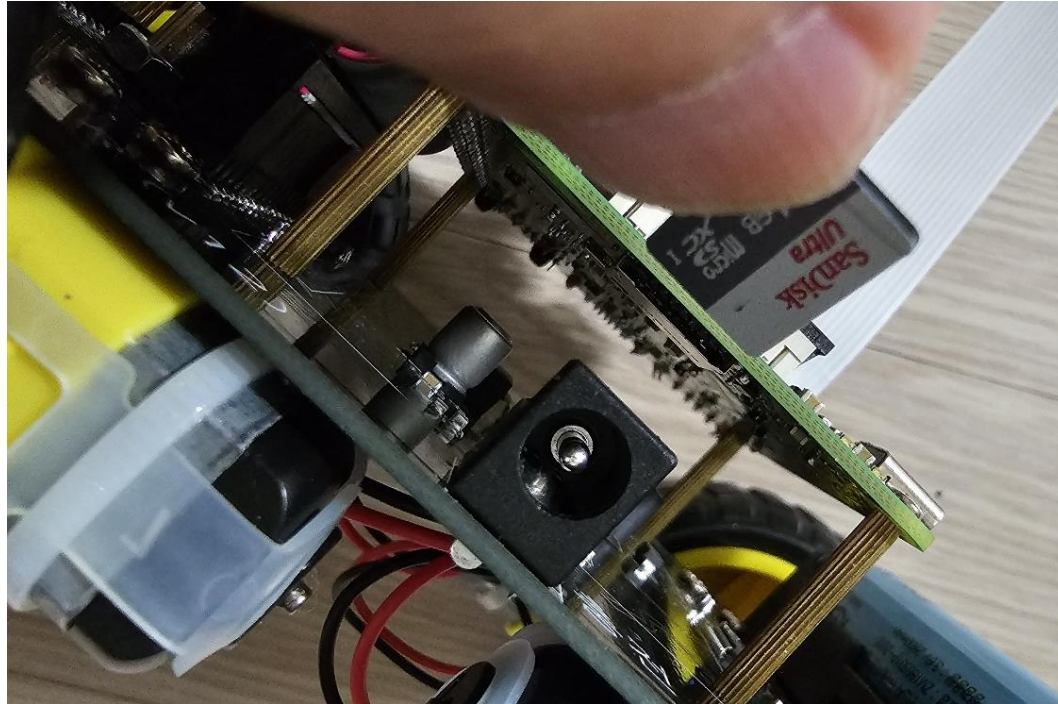
- SSH 사용 설정을 하고 내용 입력, 저장



## • 쓰기를 선택하여 MicroSD에 저장

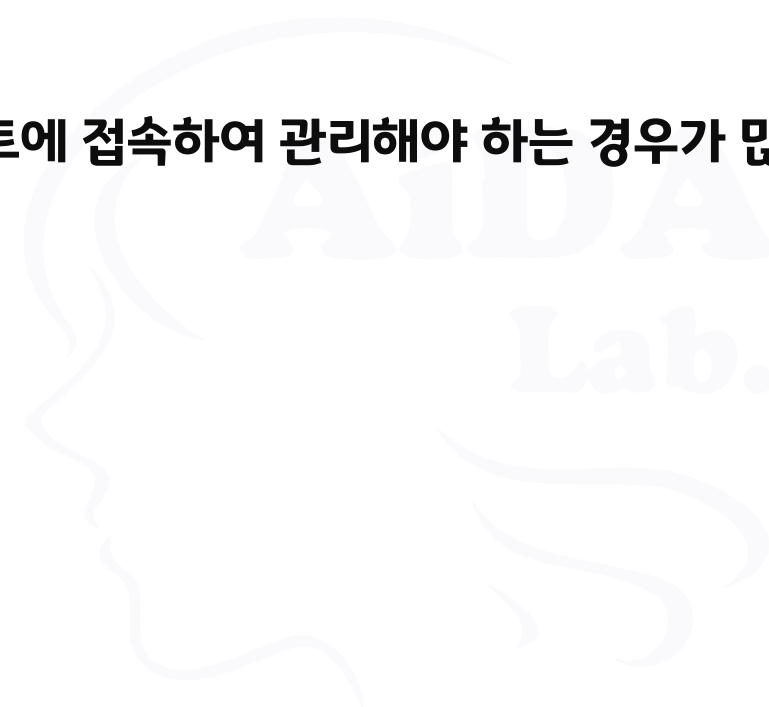


- MicroSD카드를 라즈베리파이에 꽂은 후 전원 스위치를 켜고 부팅



## • SSH (Secure Shell)

- 보안이 지원되지 않는 네트워크를 통해 네트워크 서비스를 안전하게 운영하기 위한 암호화 네트워크 프로토콜
- 원격으로 다른 시스템에 로그인 할 수 있는 대표적인 프로그램
- 다중 접속을 허용하는 리눅스에서는 하나의 서버에서 여러 클라이언트에 접속하여 관리해야 하는 경우가 많은데 이런 경우에 유용하게 사용됨
- 많이 사용되는 패키지: OpenSSH

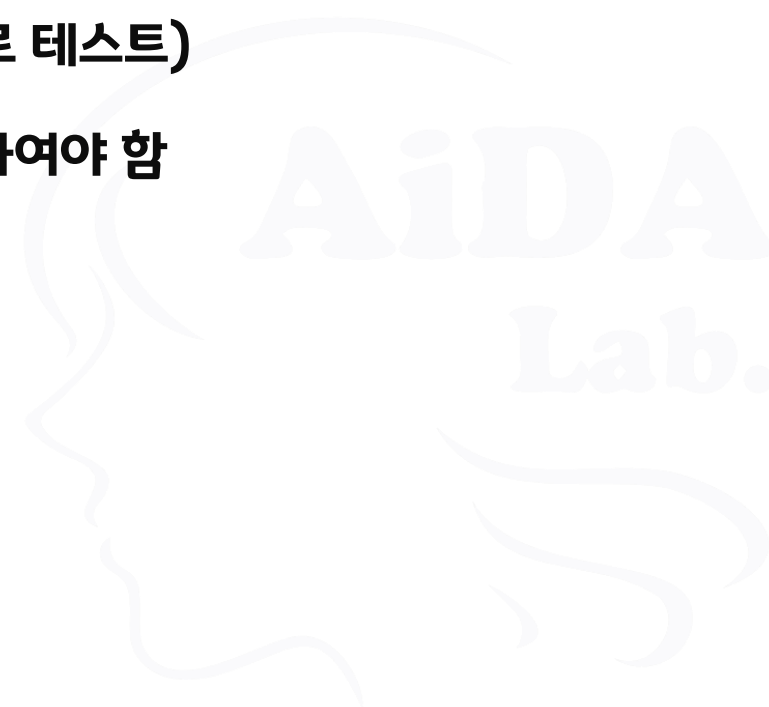


## • SSH의 특징

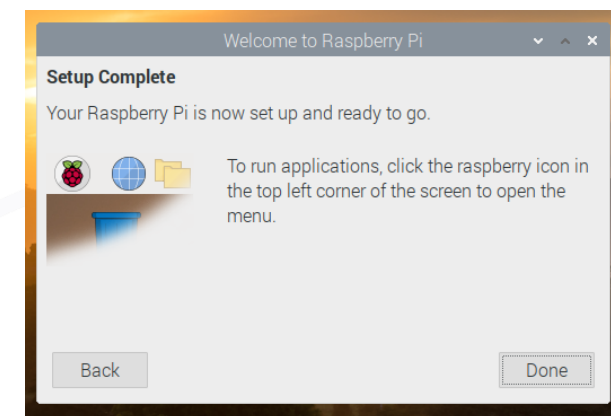
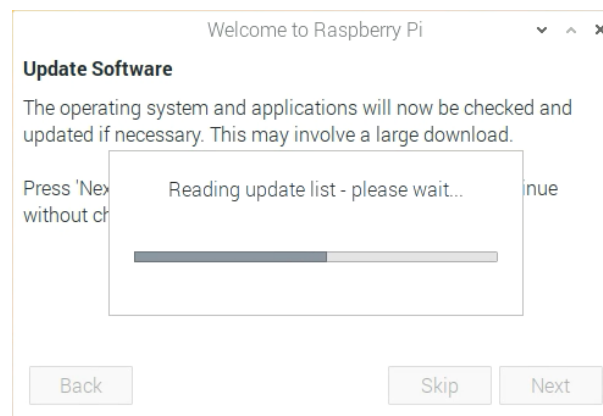
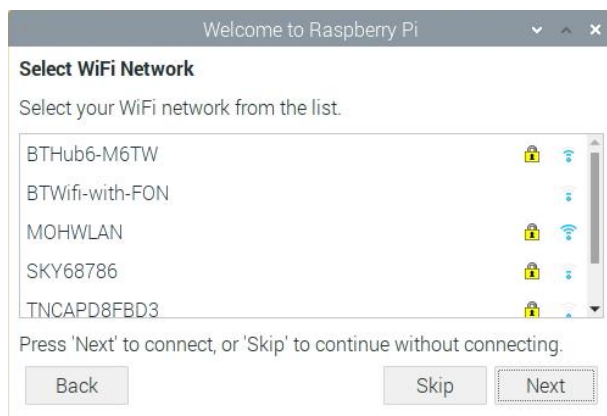
- 암호화된 패킷을 전송함
- 클라이언트와 서버라는 관계로 연결됨
- sftp를 지원함
  - ~\$ `sftp://[계정명]@[ip주소][파일경로]`
- 비밀번호 없이 로그인 가능함
- scp(원격복사 기능)를 사용할 수 있음
  - ~\$ `scp [복사될 파일명] [원격지 ip주소]:[파일]`



- SSH를 이용한 보안 원격 연결 설정 이전에 확인해야 할 조건들
  - 원격 서버의 전원이 켜져 있고 안정적인 네트워크 연결이 있어야 함
  - 서버의 IP 주소가 필요함
  - 해당 IP를 통해서 원격 서버에 액세스할 수 있어야 함(ping 명령으로 테스트)
  - SSH 서버 및 클라이언트 도구는 각 서버 및 클라이언트 OS에 설치하여야 함
  - 원격 서버의 사용자 이름과 비밀번호가 필요함
  - 방화벽이 연결을 차단해서는 안됨(미리 열어줄 필요가 있음)



## • Wi-Fi 연결 후 종료(또는 시스템 업데이트)



• 비밀번호 →  
설정 후

Wi-Fi 연결



시스템 업데이트



종료

## • 리눅스(Ubuntu 기준) 환경 설정

### • OpenSSH 서버 설치

```
~$ sudo apt update           # 패키지 데이터베이스 업데이트
~$ sudo apt upgrade         # 설치된 패키지 업그레이드
~$ sudo apt install openssh-server # OpenSSH 서버 설치
```

### • SSH 서버 구성

```
~$ sudo nano /etc/ssh/ssh_config # 환경설정 파일 열기
```

### • SSH 서비스 상태 확인

```
~$ sudo service ssh status # SSH 서비스 활성화 상태 확인
~$ sudo service ssh start  # SSH 서비스를 시작할 때
~$ sudo service ssh restart # SSH 서비스를 재시작할 때
~$ sudo service ssh stop   # SSH 서비스를 중지할 때
```

```
# IdentityFile ~/.ssh/id_rsa
# IdentityFile ~/.ssh/id_dsa
# IdentityFile ~/.ssh/id_ecdsa
# IdentityFile ~/.ssh/id_ed25519
Port 22
MaxAuthTries 4
# Protocol 2
# Ciphers aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,3des-cbc
# MACs hmac-md5,hmac-sha1,umac-64@openssh.com
# EscapeChar ~
# Tunnel no
# TunnelDevice any:any
# PermitLocalCommand no
# VisualHostKey no
# ProxyCommand ssh -q -w %h:%p gateway.example.com
# RekeyLimit 1G 1h
SendEnv LANG LC_*
HashKnownHosts yes
GSSAPIAuthentication yes
```

ssh\_config 파일을 위와 같이 수정함

SSH 서비스가 실행 중인지 확인함

## • 클라이언트에서 리눅스 SSH에 접속

### • 터미널 프로그램을 사용하여 SSH 서버에 접속함

- Windows의 경우 PowerShell 을 사용하거나 적당한 터미널 프로그램을 설치하여 사용할 수 있음

### • 접속 명령

> ssh [원격 서버] [포트번호]

또는

> ssh [사용자 이름]@[원격 서버] [포트번호]

접속명령  
입력

제대로 접속되었음

```
seokhwan@ao756: ~
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

새로운 크로스 플랫폼 PowerShell 사용 https://aka.ms/pscore6

PS C:\Users\yang> ssh seokhwan@192.168.75.81
seokhwan@192.168.75.81's password:
Welcome to Ubuntu 20.04.4 LTS (GNU/Linux 5.4.0-107-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed 14 Dec 2022 01:50:55 AM UTC

System load:  0.35           Temperature:           52.0 C
Usage of /:   92.8% of 195.86GB Processes:             138
Memory usage: 12%          Users logged in:      1
Swap usage:   0%           IPv4 address for enp4s0f0: 192.168.75.81

=> / is using 92.8% of 195.86GB

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

53 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

New release '22.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

*** System restart required ***
Last login: Wed Dec 14 01:50:19 2022 from 192.168.75.243
seokhwan@ao756:~$
```

- 원격 접속 서버(Ubuntu 기준) 의 GUI환경을 사용하고 싶다면..
  - 다양한 프로그램을 사용하여 원격 접속 서버의 GUI환경을 사용할 수 있음
    - RealVNC (VNC Viewer)
    - TightVNC
    - Xrdp
    - 어떤 프로그램을 사용해도 무방함. 자기에게 편한 것을 선택하여 사용

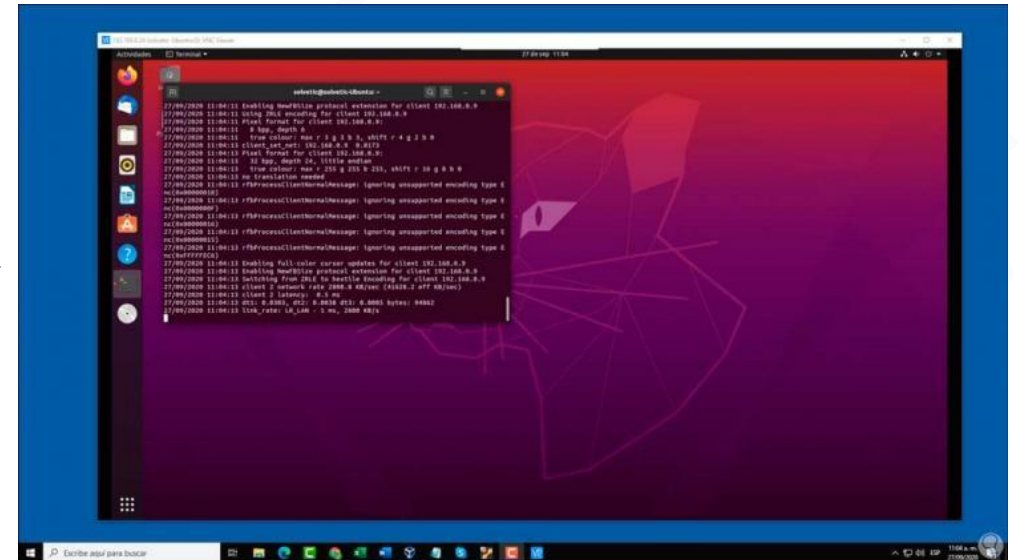
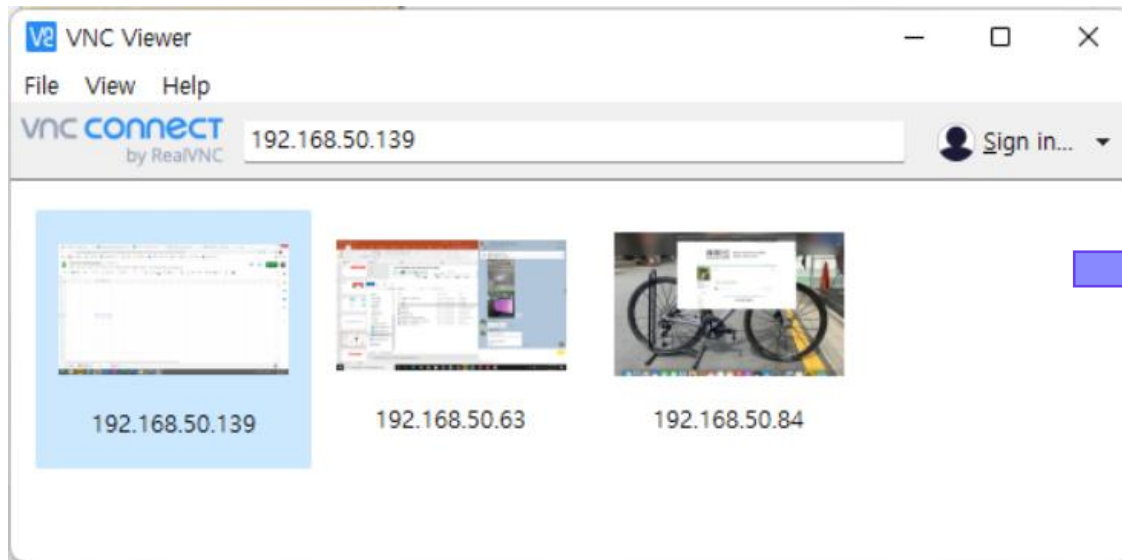


- VNC Viewer를 이용한 예시 (원격서버는 Ubuntu 기준)

- VNC Viewer 다운로드 후 설치

- <https://www.realvnc.com/en/connect/download/viewer/>

- VNC Viewer 실행 → IP 입력 → 접속



- **라즈베리파이를 사용하여 개발하는 방법**

- **첫 번째 방법**

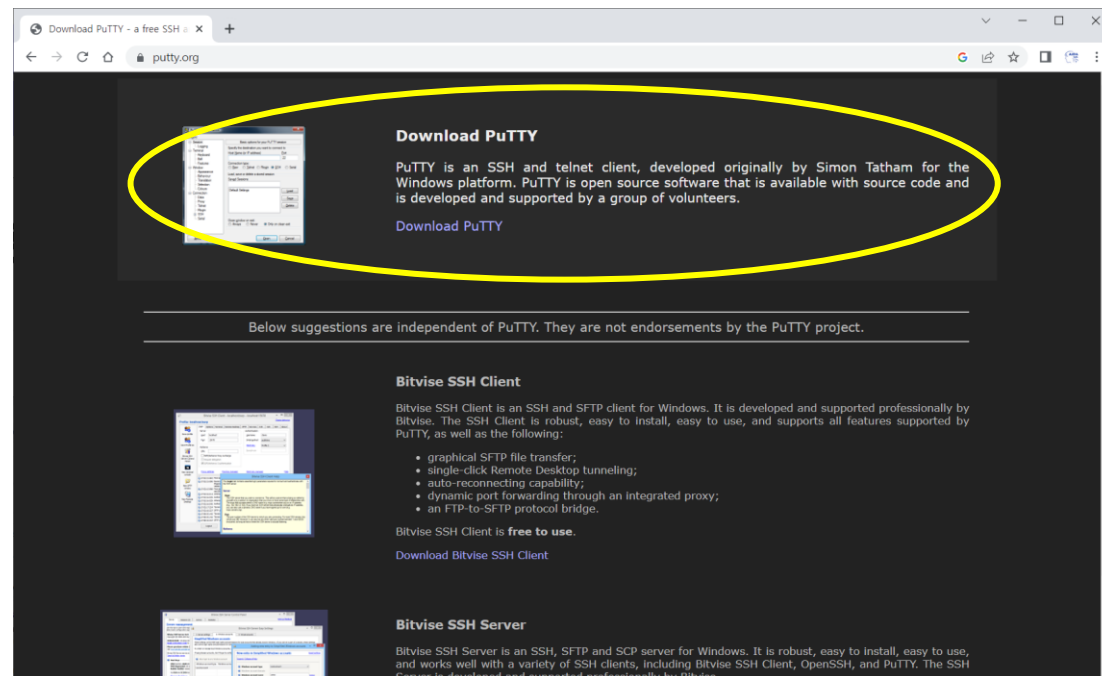
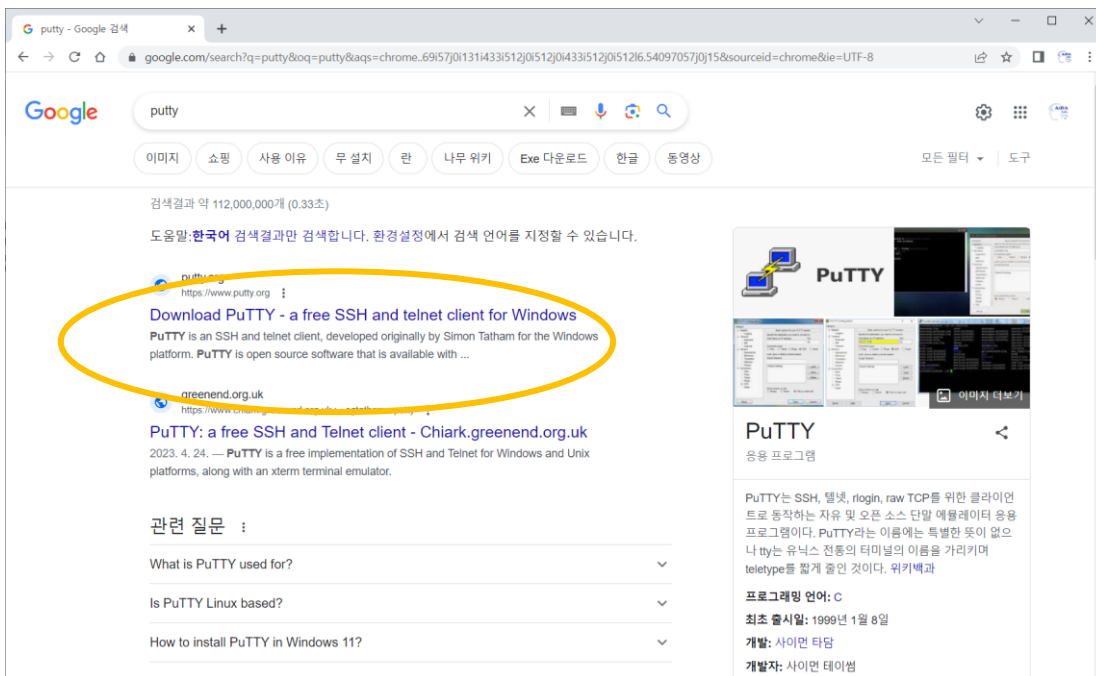
- 그냥 컴퓨터처럼 모니터, 키보드, 마우스를 연결하여 사용함
- 자율주행 자동차의 경우 줄줄이 달고 다니기 힘들

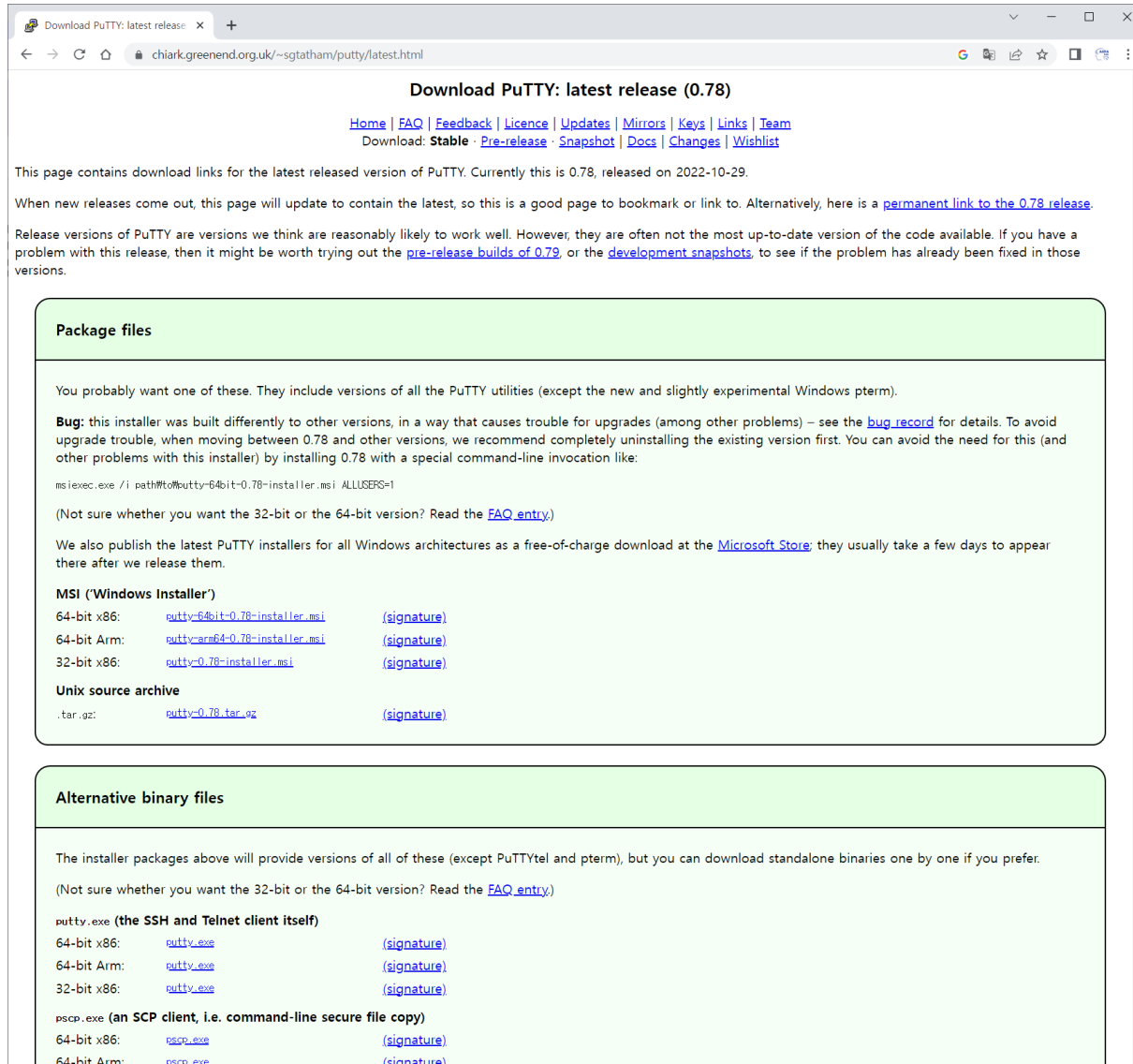
- **두 번째 방법**

- 원격 GUI 환경으로 접속하여 사용함
- 네트워크 환경이 필요함
- 작업 시 속도가 조금 느림(실제 사용 시에는 문제 없음)



- PuTTY 프로그램을 이용하여 SSH 프로토콜로 원격 접속
  - PuTTY 검색하여 다운로드





Download PuTTY: latest release (0.78)

[Home](#) | [FAQ](#) | [Feedback](#) | [Licence](#) | [Updates](#) | [Mirrors](#) | [Keys](#) | [Links](#) | [Team](#)  
Download: [Stable](#) · [Pre-release](#) · [Snapshot](#) | [Docs](#) | [Changes](#) | [Wishlist](#)

This page contains download links for the latest released version of PuTTY. Currently this is 0.78, released on 2022-10-29.

When new releases come out, this page will update to contain the latest, so this is a good page to bookmark or link to. Alternatively, here is a [permanent link to the 0.78 release](#).

Release versions of PuTTY are versions we think are reasonably likely to work well. However, they are often not the most up-to-date version of the code available. If you have a problem with this release, then it might be worth trying out the [pre-release builds of 0.79](#), or the [development snapshots](#), to see if the problem has already been fixed in those versions.

### Package files

You probably want one of these. They include versions of all the PuTTY utilities (except the new and slightly experimental Windows pterm).

**Bug:** this installer was built differently to other versions, in a way that causes trouble for upgrades (among other problems) – see the [bug record](#) for details. To avoid upgrade trouble, when moving between 0.78 and other versions, we recommend completely uninstalling the existing version first. You can avoid the need for this (and other problems with this installer) by installing 0.78 with a special command-line invocation like:

```
msiexec.exe /i path\to\putty-64bit-0.78-installer.msi ALLUSERS=1
```

(Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

We also publish the latest PuTTY installers for all Windows architectures as a free-of-charge download at the [Microsoft Store](#); they usually take a few days to appear there after we release them.

**MSI ('Windows Installer')**

64-bit x86:	<a href="#">putty-64bit-0.78-installer.msi</a>	<a href="#">(signature)</a>
64-bit Arm:	<a href="#">putty-arm64-0.78-installer.msi</a>	<a href="#">(signature)</a>
32-bit x86:	<a href="#">putty-0.78-installer.msi</a>	<a href="#">(signature)</a>

**Unix source archive**

.tar.gz:	<a href="#">putty-0.78.tar.gz</a>	<a href="#">(signature)</a>
----------	-----------------------------------	-----------------------------

### Alternative binary files

The installer packages above will provide versions of all of these (except PuTTYtel and pterm), but you can download standalone binaries one by one if you prefer.

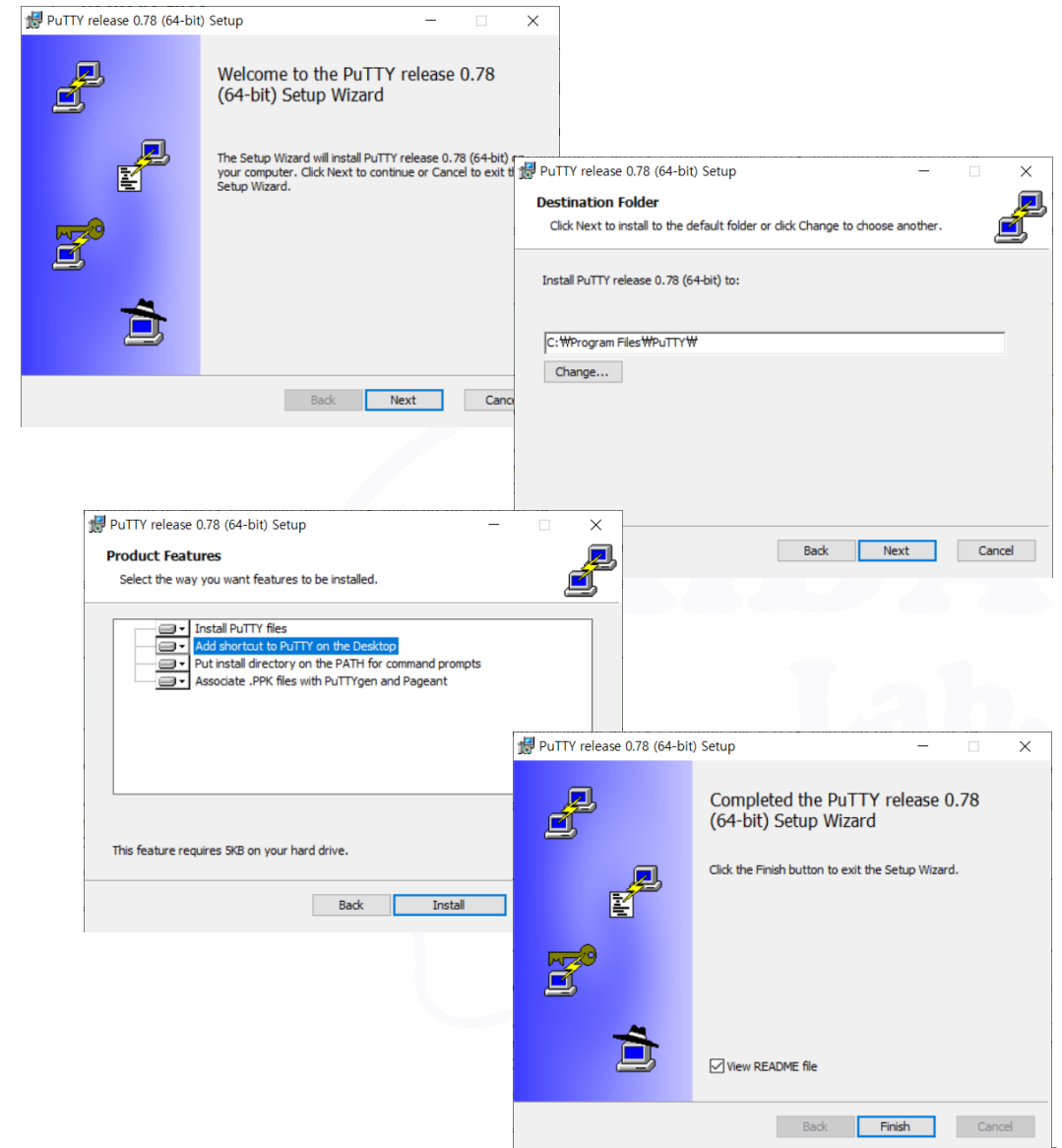
(Not sure whether you want the 32-bit or the 64-bit version? Read the [FAQ entry](#).)

**putty.exe (the SSH and Telnet client itself)**

64-bit x86:	<a href="#">putty.exe</a>	<a href="#">(signature)</a>
64-bit Arm:	<a href="#">putty.exe</a>	<a href="#">(signature)</a>
32-bit x86:	<a href="#">putty.exe</a>	<a href="#">(signature)</a>

**pscp.exe (an SCP client, i.e. command-line secure file copy)**

64-bit x86:	<a href="#">pscp.exe</a>	<a href="#">(signature)</a>
64-bit Arm:	<a href="#">pscp.exe</a>	<a href="#">(signature)</a>



PuTTY release 0.78 (64-bit) Setup

Welcome to the PuTTY release 0.78 (64-bit) Setup Wizard

The Setup Wizard will install PuTTY release 0.78 (64-bit) on your computer. Click Next to continue or Cancel to exit the Setup Wizard.

**Destination Folder**

Click Next to install to the default folder or click Change to choose another.

Install PuTTY release 0.78 (64-bit) to:

C:\Program Files\PuTTY\

**Product Features**

Select the way you want features to be installed.

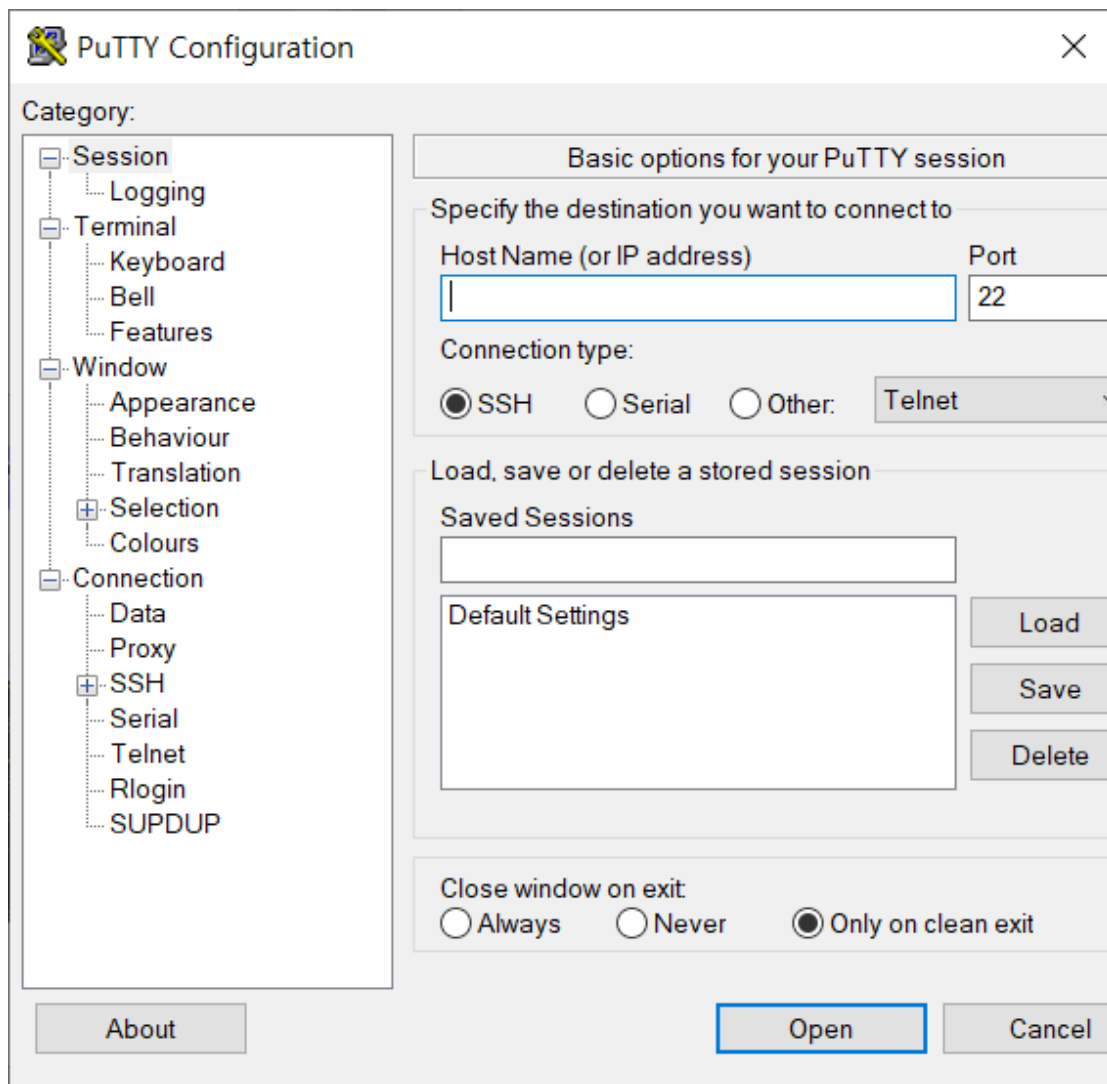
- Install PuTTY files
- Add shortcut to PuTTY on the Desktop
- Put install directory on the PATH for command prompts
- Associate .PPK files with PuTTYgen and Pageant

This feature requires 8KB on your hard drive.

Completed the PuTTY release 0.78 (64-bit) Setup Wizard

Click the Finish button to exit the Setup Wizard.

View README file



- 할당받은 IP 사용
- ID/PW: pi / raspberry



**THANK  
YOU**

