

IoT Platform 5th Week

- RPi and Sensors -

Jaeseok Yun

Soonchunhyang University

PIR 센서

HC-SR501

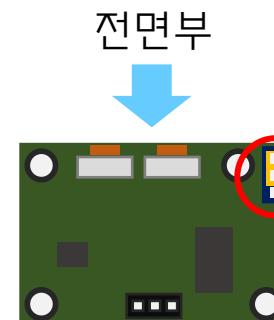
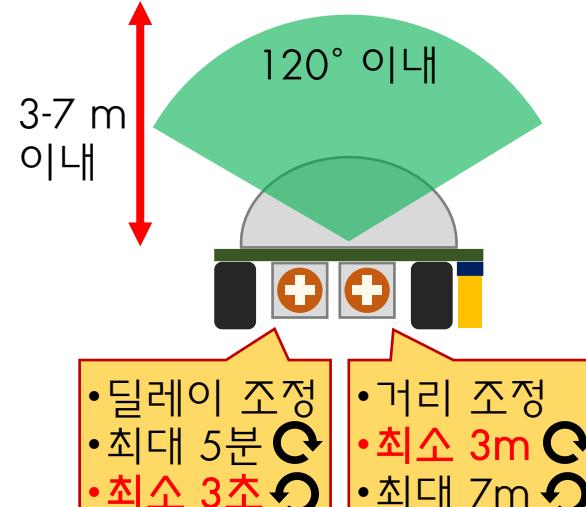
PIR 센서 특징과 조절

■ HC-SR501

- PIR¹ (passive infrared sensor) 센서인 LHI778를 내장한 센서 모듈
- 디지털 인터페이스
- 딜레이 시간: 조정 가능 (3 sec → 5 min)
- 감지 범위: 120° 이내, 3-7 m 이내 (조정 가능)



HC-SR501²



주의: 딜레이 종료 후 3초간 무동작

• 단일 트리거
• 동작 감지 후 바로 딜레이 시작

• 반복 트리거
• 동작 감지가 딜레이 초기화
• 마지막 감지 후 바로 딜레이 시작

¹ 인체에서 나오는 적외선의 변화에 전하의 배열이 변화하는 성질을 가진 소재를 이용해 인체를 감지하는 센서, pyroelectric infrared sensor 라고 부르기도 함

² <http://www.eleparts.co.kr/goods/view?no=3227278>

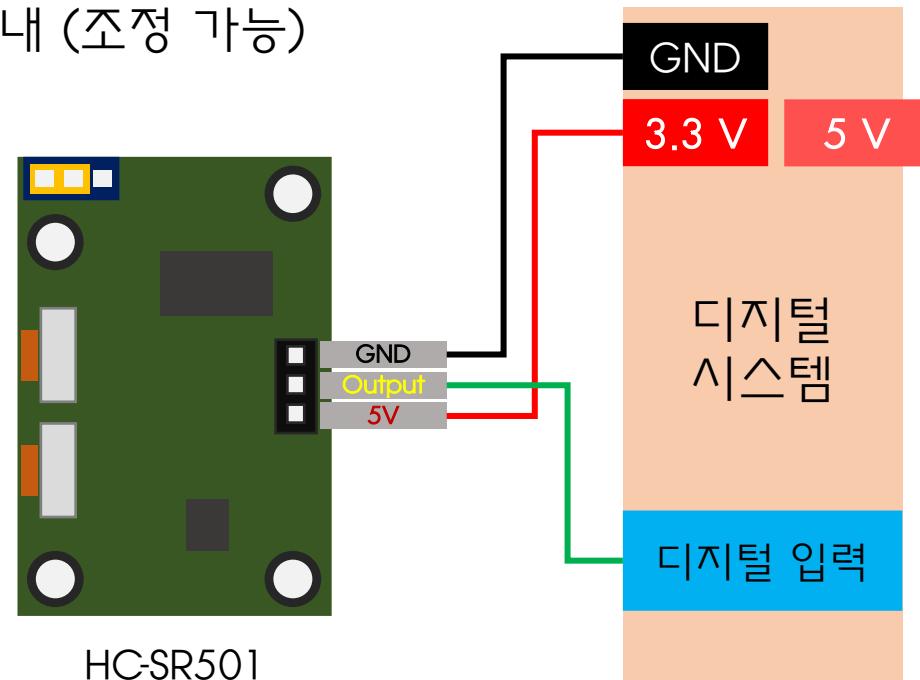
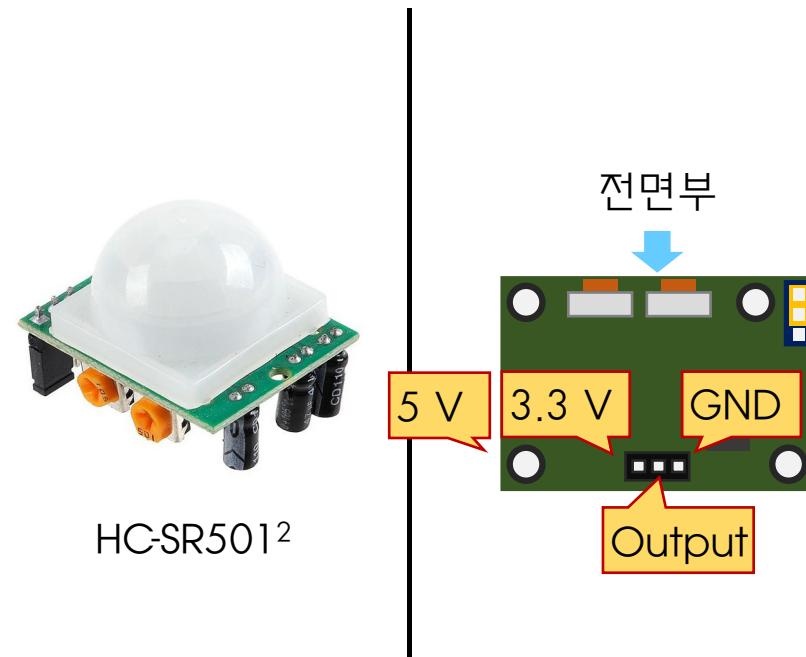
³ <http://henrysbench.capnfatz.com/henrys-bench/arduino-sensors-and-input/arduino-hc-sr501-motion-sensor-tutorial/>

PIR 센서 회로 구성

Product Type	HC-SR501 Body Sensor Module
Operating Voltage Range	5-20VDC
Quiescent Current	<50uA
Level output	High 3.3 V /Low 0V

■ HC-SR501

- PIR¹ (passive infrared sensor) 센서인 LHI778를 내장한 센서 모듈
- 디지털 인터페이스
- 딜레이 시간: 조정 가능 (3 sec → 5 min)
- 감지 범위: 120° 이내, 3-7 m 이내 (조정 가능)



¹ 인체에서 나오는 적외선의 변화에 전하의 배열이 변화하는 성질을 가진 소재를 이용해 인체를 감지하는 센서, pyroelectric infrared sensor 라고 부르기도 함

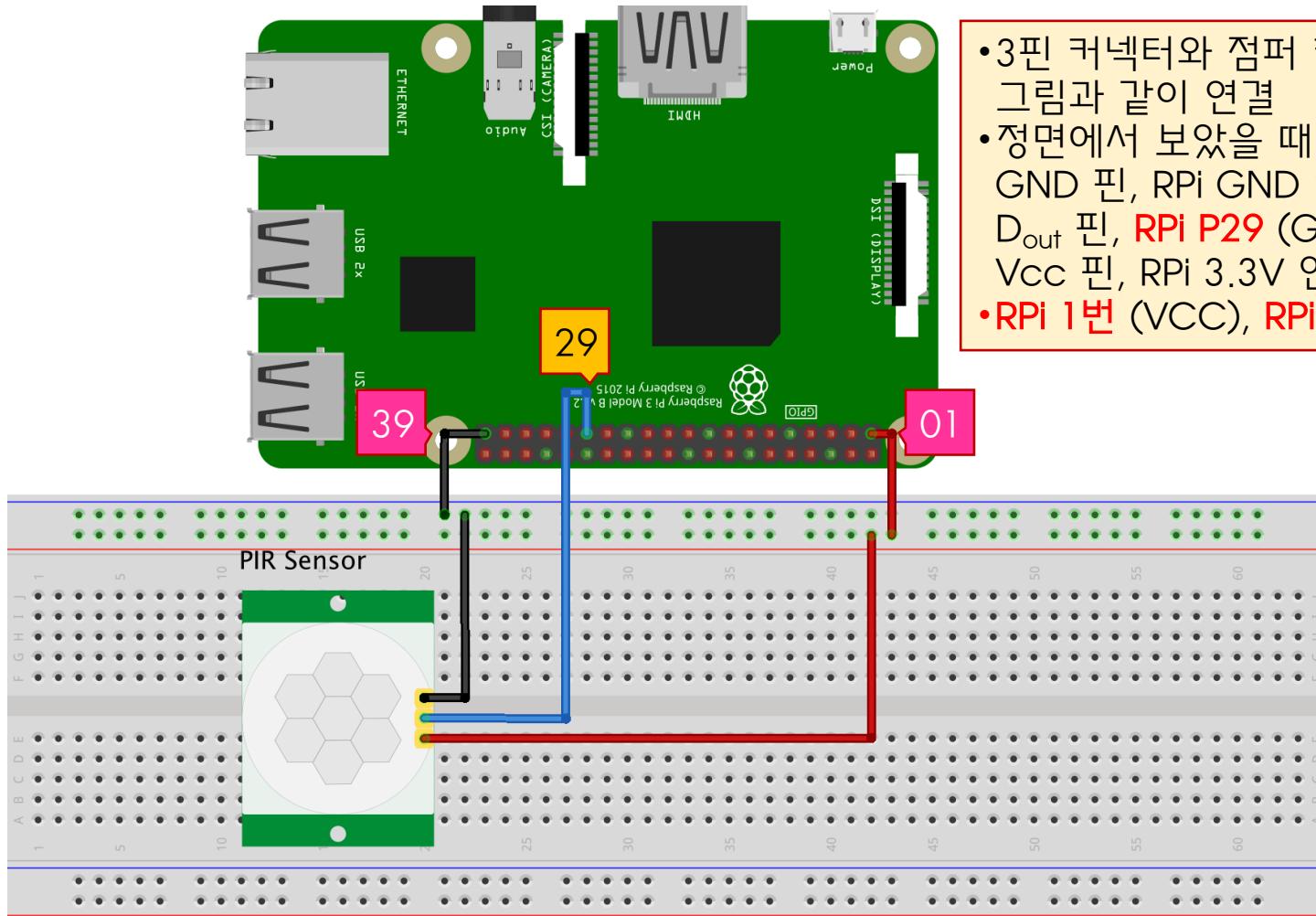
² <http://www.eleparts.co.kr/goods/view?no=3227278>

³ <http://henrysbench.capnfatz.com/henrys-bench/arduino-sensors-and-input/arduino-hc-sr501-motion-sensor-tutorial/>

PIR 센서 회로 구성

Practice

■ PIR 센서와 RPi 연결



- 3핀 커넥터와 점퍼 케이블을 이용하여 그림과 같이 연결
- 정면에서 보았을 때 (데이터시트 확인!), GND 핀, RPi GND 연결
D_{out} 핀, **RPi P29** (GPIO05) 연결
Vcc 핀, RPi 3.3V 연결
- **RPi 1번** (VCC), **RPi 39번** (GND) 구성

PIR 센서 wPi 프로그래밍

Practice

```
#include <iostream>
#include <wiringPi.h>
using namespace std;

#define PIN_PIR 5

void sensePIR() {
    static int count = 0;
    cout << "Somebody sensed " << count++ << endl;
}
```

```
int main(int argc, char* argv[]) {

    wiringPiSetupGpio();
    pinMode(PIN_PIR, INPUT);

    wiringPiISR(PIN_PIR, INT_EDGE_RISING, &sensePIR);

    for (;;) {
    }

    return 0;
}
```

• PIR 센서 입력으로 GPIO05 (P29)를 사용

• ISR이 호출 될 때마다 감지 횟수 (=count) 출력

• GPIO05 디지털 입력으로 설정

• ISR `sensePIR()` 등록

• '버튼 입력' 등을 **무한 루프를 벗어나기** 위한 조건으로 이용하는 것이 필요
 • 없을 때에는 **CTRL + C**로 실행 중단 가능

PIR 센서 프로그램 실행

Practice

```
pi@raspberrypi: ~/ex $ g++ XX_Sensor_PIR.cpp -o XX_Sensor_PIR -lwiringPi  
pi@raspberrypi:~/ex $ ./XX_Sensor_PIR  
Somebody sensed 0  
Somebody sensed 1  
Somebody sensed 2  
Somebody sensed 3  
Somebody sensed 4  
Somebody sensed 5  
Somebody sensed 6  
Somebody sensed 7  
Somebody sensed 8  
Somebody sensed 9  
Somebody sensed 10
```

•-lwiringPi 옵션으로 wPi 라이브러리 위치를 지정해서 컴파일

•감지 회수 출력

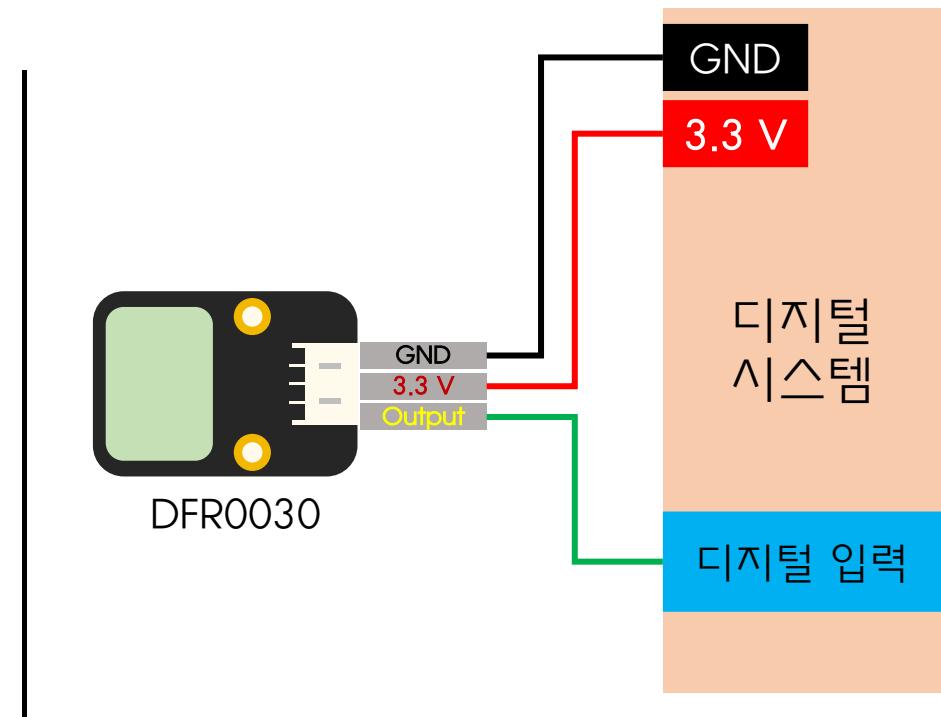
터치 센서

DFR0030

터치 센서

■ DFR0030

- DFRobot 터치 센서 모듈¹
- 디지털 인터페이스
- 사람의 터치를 감지하면 디지털 신호를 발생



DFRobot 디지털 터치 센서²

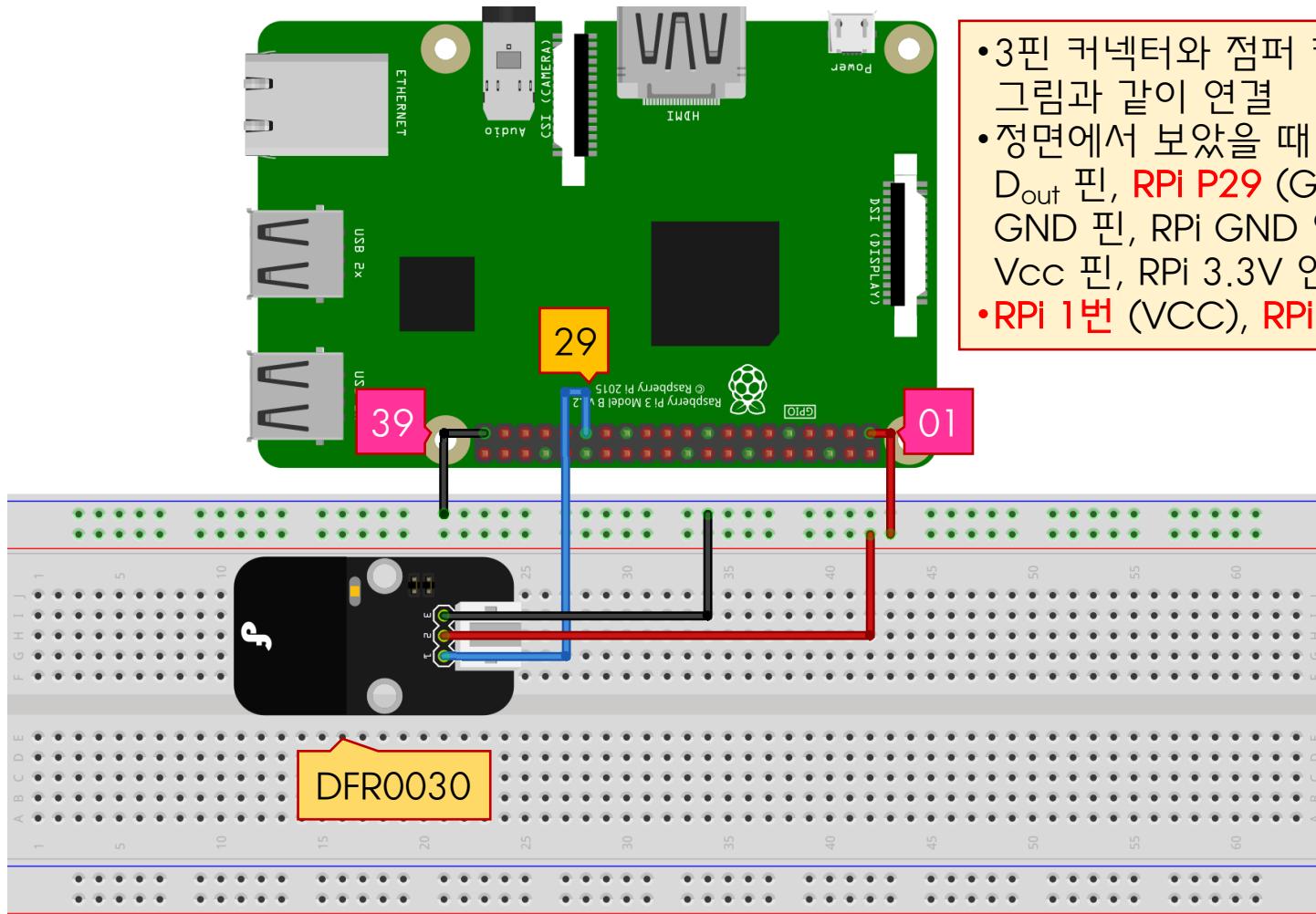
¹ 센서 모듈은 센서 동작에 필요한 전기 회로 (예, 저항)와 인터페이스를 제공하여 센서를 이용하기 편리하게 만들어 놓은 장치, 센서를 이용해 누구나 제작, 판매 가능

² https://www.dfrobot.com/wiki/index.php/DFRobot_Capacitive_Touch_Sensor_SKU:DFR0030

터치 센서 회로 구성

Practice

■ 터치 센서와 RPi 연결



- 3핀 커넥터와 점퍼 케이블을 이용하여 그림과 같이 연결
- 정면에서 보았을 때 (데이터시트 확인!), D_{out} 핀, **RPi P29** (GPIO05) 연결
GND 핀, RPi GND 연결
Vcc 핀, RPi 3.3V 연결
- **RPi 1번** (VCC), **RPi 39번** (GND) 구성

터치 센서 wPi 프로그래밍

Practice

```
#include <iostream>
#include <wiringPi.h>
using namespace std;

#define PIN_TOUCH 5
#define DEBOUNCE_TIME 200

void senseTouch() {
    static unsigned long lastISRTIME = 0;
    static unsigned long count = 0;
    unsigned long currentISRTIME = millis();
    if (currentISRTIME - lastISRTIME > DEBOUNCE_TIME) {
        cout << "Touch sensed " << count++ << endl;
    }
    lastISRTIME = currentISRTIME;
}

int main(int argc, char* argv[]){
    wiringPiSetupGpio();
    pinMode(PIN_TOUCH, INPUT);

    wiringPiISR(PIN_TOUCH, INT_EDGE_RISING, &senseTouch);

    for(;;) { }

    cout << "Finished the touch sensor program" << endl;
    return 0;
}
```

- 터치 센서 입력으로 GPIO05 를 사용
- Capacitive 터치 센서 특징으로 인한 전기적 채터링 제거를 위해 디바운스 적용

- ISR이 호출 될 때마다 터치 횟수 (=count) 출력

- GPIO05 디지털 입력으로 설정

- ISR **senseTouch()** 등록

터치 센서 프로그램 실행

Practice

```
pi@raspberrypi: ~/ex  
pi@raspberrypi:~/ex $ g++ ./XX_Sensor_touch.cpp -o XX_Sensor_touch -lwiringPi  
pi@raspberrypi:~/ex $ ./XX_Sensor_touch  
Touch sensed 0  
Touch sensed 1  
Touch sensed 2  
Touch sensed 3 •터치 회수 출력  
Touch sensed 4  
Touch sensed 5  
Touch sensed 6  
Touch sensed 7  
Touch sensed 8  
Touch sensed 9
```

•-lwiringPi 옵션으로 wPi 라이브러리 위치를 지정해서 컴파일

주변 소리 센서

Waveshare sound sensor

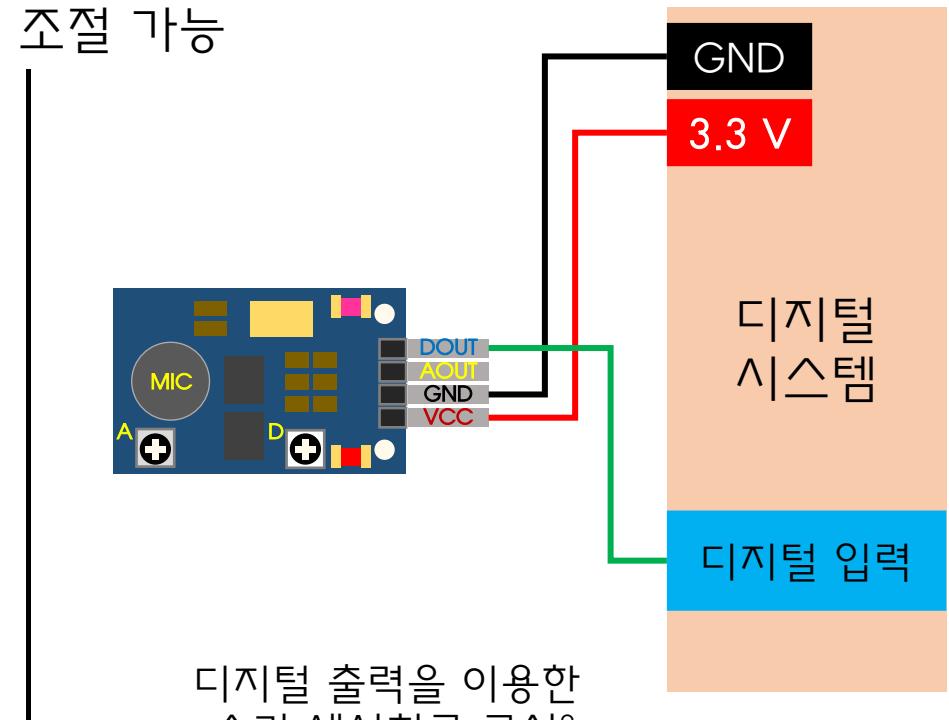
주변 소리 센서 회로 구성

■ 소리 센서 (ambient sound sensor)

- WaveShare 소리 센서 모듈¹
- 아날로그/디지털 인터페이스
- 마이크로 소리를 감지하면 신호를 발생
- 가변 저항으로 반응 소리 감도 조절 가능



WaveShare 소리 센서²



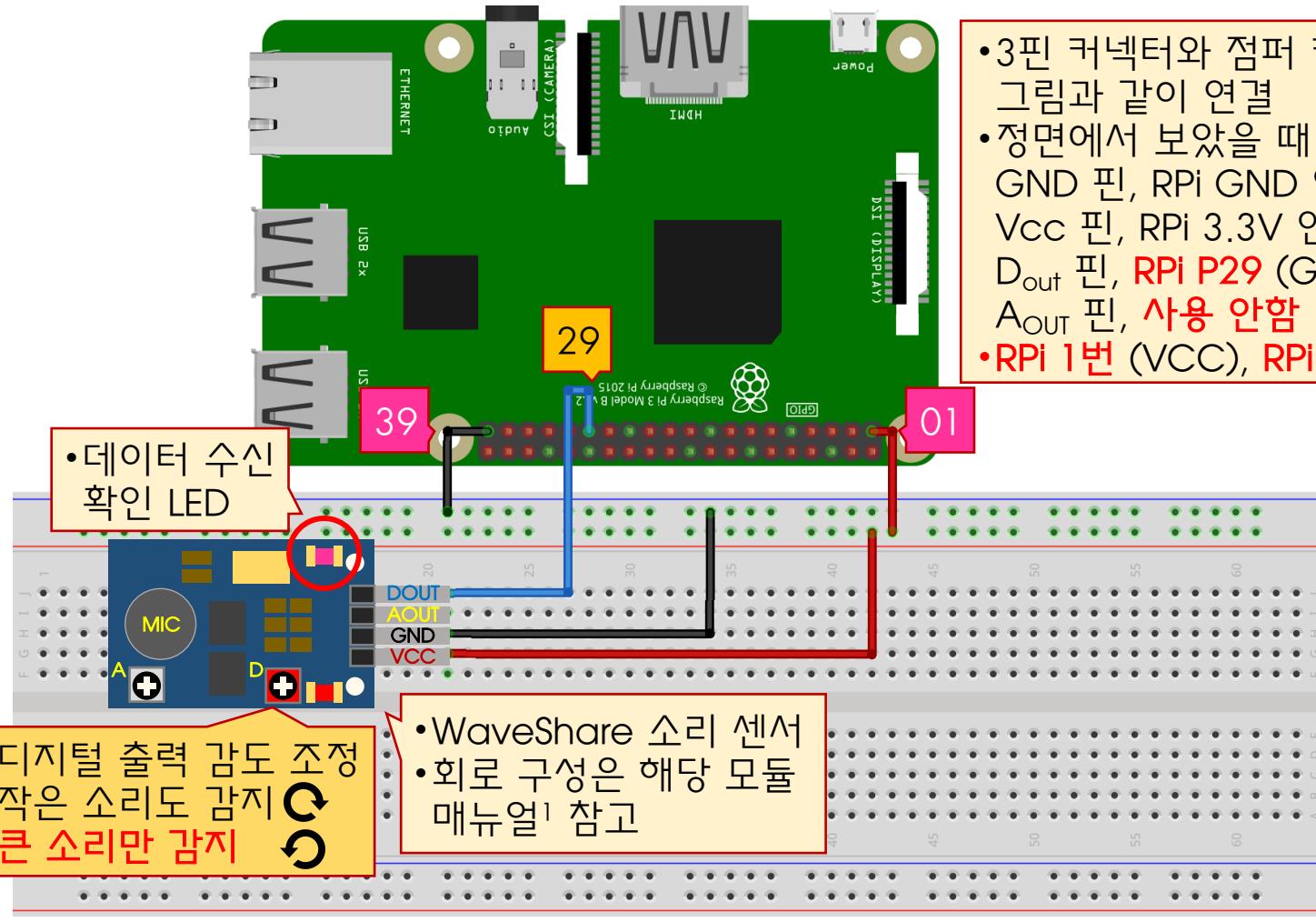
¹ 센서 모듈은 센서 동작에 필요한 전기 회로 (예, 저항)와 인터페이스를 제공하여 센서를 이용하기 편리하게 만들어 놓은 장치, 센서를 이용해 누구나 제작, 판매 가능

² <https://www.waveshare.com/sound-sensor.htm>

주변 소리 센서 회로 구성

Practice

■ 소리 센서와 RPi 연결



주변 소리 센서 프로그램 실행

Practice

pi@raspberrypi: ~/ex

```
pi@raspberrypi:~/ex $ g++ ./XX_Sensor_touch.cpp -o XX_Sensor_touch -lwiringPi  
pi@raspberrypi:~/ex $ ./XX_Sensor_touch
```

```
Touch sensed 0  
Touch sensed 1  
Touch sensed 2  
Touch sensed 3  
Touch sensed 4  
Touch sensed 5  
Touch sensed 6  
Touch sensed 7  
Touch sensed 8  
Touch sensed 9
```

- lwiringPi 옵션으로 wPi 라이브러리 위치를 지정해서 컴파일

- 터치 센서에서 사용한 프로그램과 동일한 프로그램을 적용
- 소리가 날 때마다 메시지 출력 확인

초음파 센서

HC-SR04P

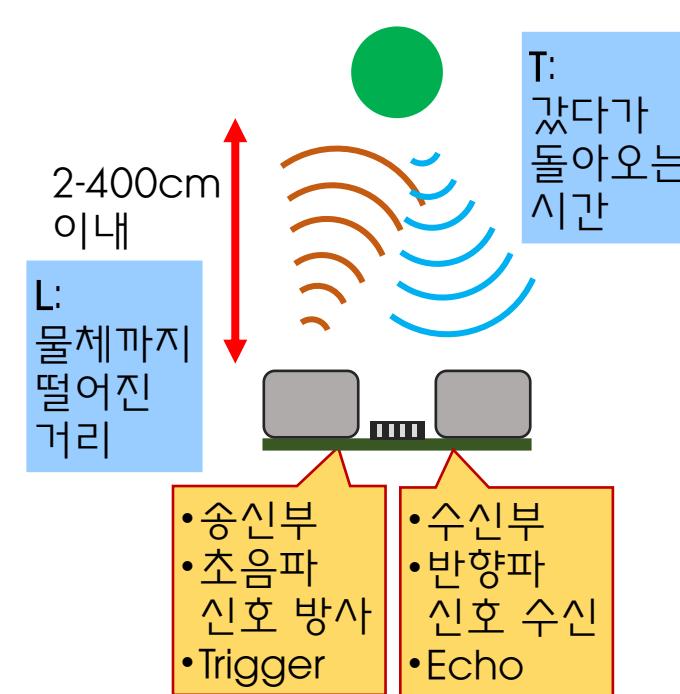
초음파 센서 특징과 원리³

■ HC-SR04P

- 거리/근접 여부를 측정할 수 있는 초음파¹ 센서 모듈
- HC-SR04 (4.8-5V 동작)의 새 버전으로서 3.3-5V에서 동작
- 동작 거리/정확도 (operating range & accuracy): 2-400cm ($\pm 2\text{mm}$)
- 유효 각도 (effective angle): 15°



HC-SR04P²



$$\begin{aligned} \text{음속} &= 340 \left(\frac{\text{m}}{\text{s}} \right) = 340 \frac{10^2 \text{cm}}{10^6 \mu\text{s}} \\ &= \frac{34}{10^3} \left(\frac{\text{cm}}{\mu\text{s}} \right) \end{aligned}$$

$$\begin{aligned} T(\mu\text{s}) &= \frac{\text{거리}/\text{속도}}{2L (\text{cm})} \\ &= \frac{34/10^3 (\text{cm}/\mu\text{s})}{2L (\text{cm})} \end{aligned}$$

$$\begin{aligned} L(\text{cm}) &= T(\mu\text{s}) * \frac{34}{10^3} * \frac{1}{2} \left(\frac{\text{cm}}{\mu\text{s}} \right) \\ &= T * \frac{17}{1000} (\text{cm}) = T * 0.017 (\text{cm}) \\ &= \frac{T}{58.82} (\text{cm}) \end{aligned}$$

¹ Ultrasound: 인간이 들을 수 있는 영역을 벗어난 주파수를 ($> 20\text{ kHz}$) 갖는 소리파 (sound wave)로서 물체에 반사되어 돌아오는 성질을 이용해 거리 계산 활용

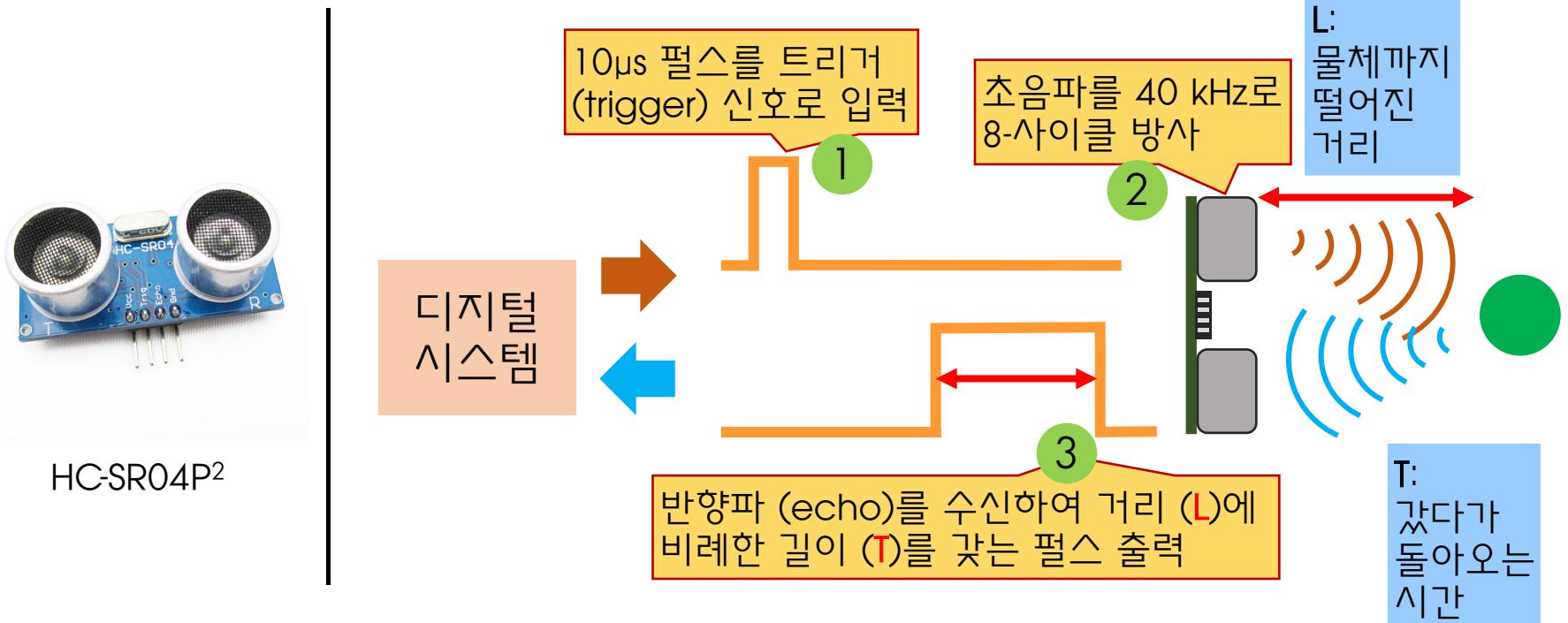
² <http://www.eleparts.co.kr/goods/view?no=3135372>

³ <http://handsontec.com/index.php/product/hc-sr04-ultrasonic-ranging-module/>

초음파 센서 특징과 원리³

■ HC-SR04P

- 거리/근접 여부를 측정할 수 있는 초음파¹ 센서 모듈
- HC-SR04 (4.8-5V 동작)의 새 버전으로서 3.3-5V에서 동작
- 동작 거리/정확도 (operating range & accuracy): 2-400cm ($\pm 2\text{mm}$)
- 유효 각도 (effective angle): 15°



¹ Ultrasound: 인간이 들을 수 있는 영역을 벗어난 주파수를 ($> 20\text{ kHz}$) 갖는 소리파 (sound wave)로서 물체에 반사되어 돌아오는 성질을 이용해 거리 계산 활용

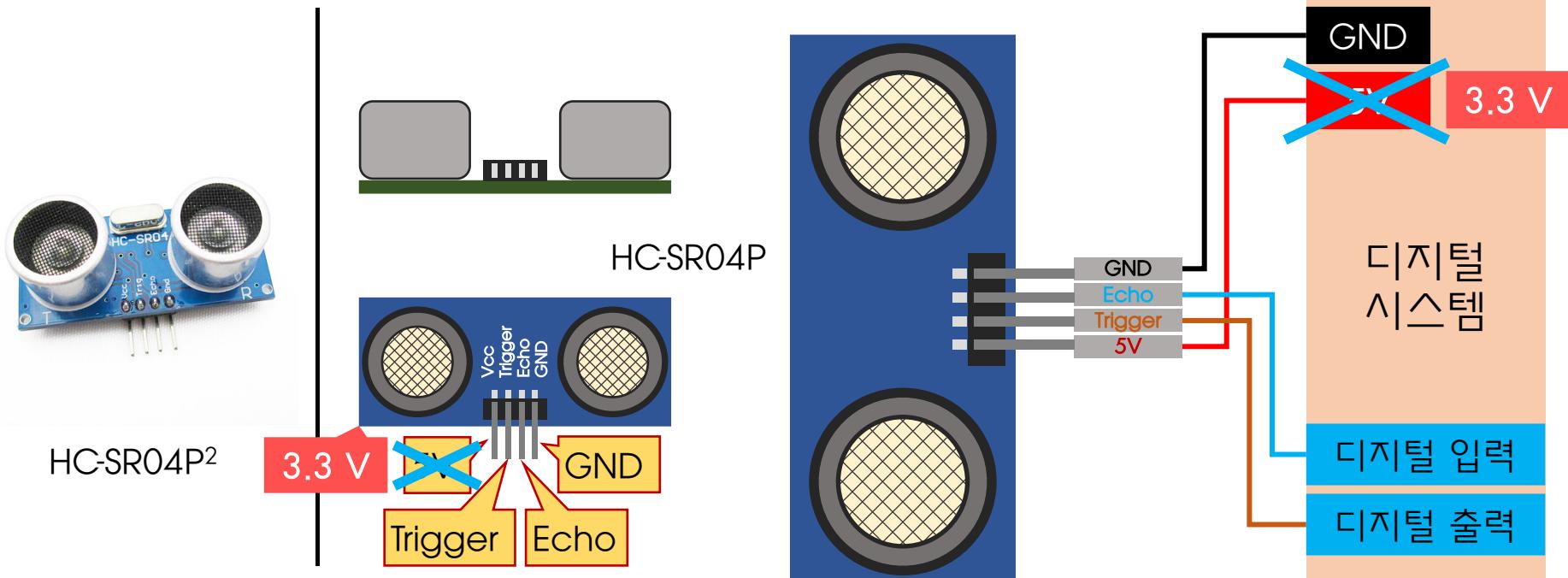
² <http://www.eleparts.co.kr/goods/view?no=3135372>

³ <http://handsontec.com/index.php/product/hc-sr04-ultrasonic-ranging-module/>

초음파 센서 회로 구성³

■ HC-SR04P

- 거리/근접 여부를 측정할 수 있는 초음파¹ 센서 모듈
- HC-SR04 (4.8-5V 동작)의 새 버전으로서 **3.3-5V에서 동작**
- 동작 거리/정확도 (operating range & accuracy): 2-400cm ($\pm 2\text{mm}$)
- 유효 각도 (effective angle): 15°



¹ Ultrasound: 인간이 들을 수 있는 영역을 벗어난 주파수를 ($> 20\text{ kHz}$) 갖는 소리파 (sound wave)로서 물체에 반사되어 돌아오는 성질을 이용해 거리 계산 활용

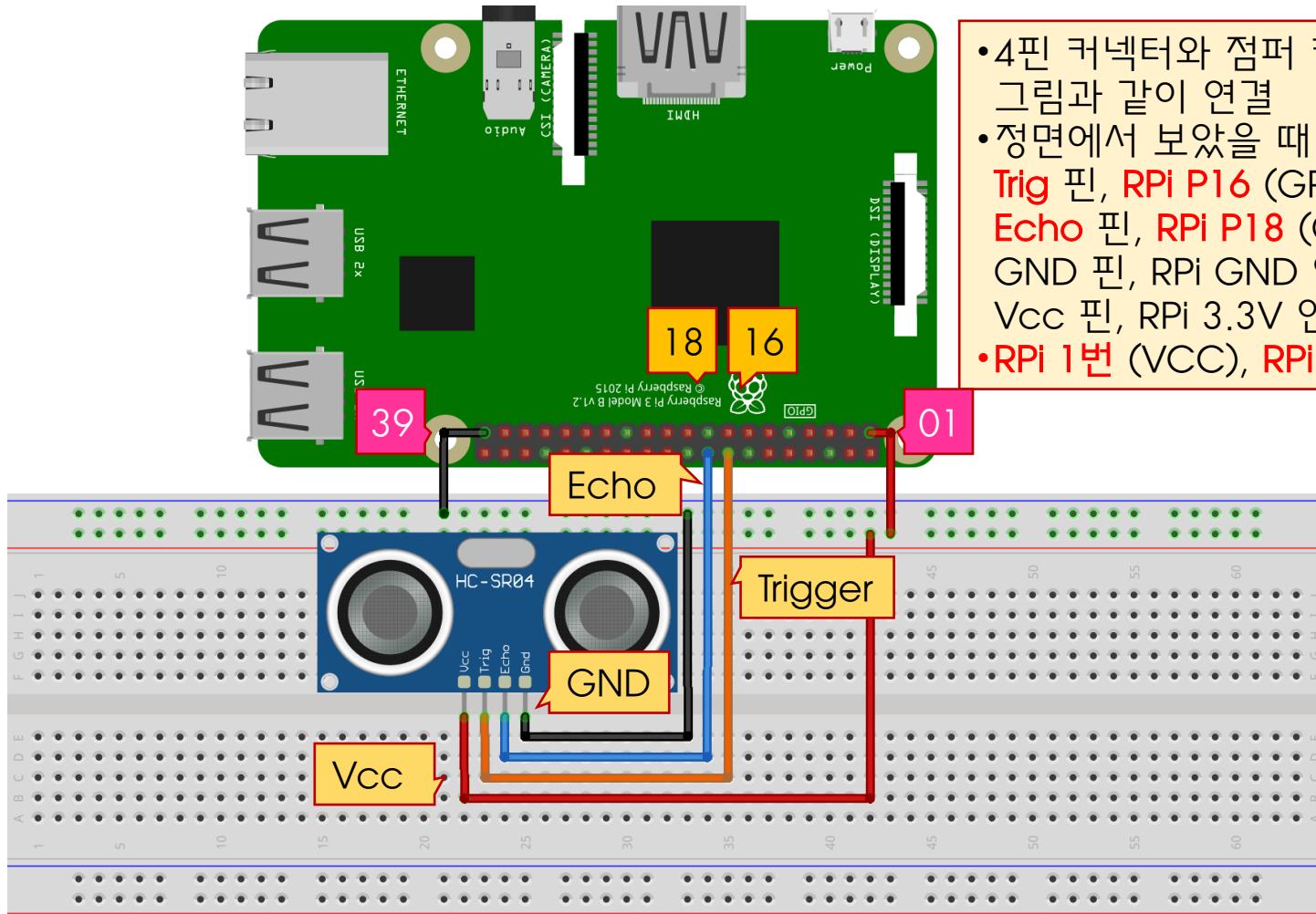
² <http://www.eleparts.co.kr/goods/view?no=3135372>

³ <http://handsontec.com/index.php/product/hc-sr04-ultrasonic-ranging-module/>

초음파 센서 회로 구성

Practice

■ 초음파 센서와 RPi 연결



- 4핀 커넥터와 점퍼 케이블을 이용하여 그림과 같이 연결
- 정면에서 보았을 때 (**데이터시트 확인!**), **Trig** 핀, **RPI P16** (GPIO23) 연결
- **Echo** 핀, **RPI P18** (GPIO24) 연결
- GND 핀, RPI GND 연결
- Vcc 핀, RPI 3.3V 연결
- **RPI 1번** (VCC), **RPI 39번** (GND) 구성

초음파 센서 wPi 프로그래밍

```
#include <iostream>
#include <wiringPi.h>
using namespace std;
#define PIN_TRIGGER 23
#define PIN_ECHO 24
#define RANGE_MAX 200
#define RANGE_MIN 0
```

- 프로그램에서 wiringSetup() 함수가 호출 된 후 측정 시작
- unsigned int millis(void): msec 단위 시간 반환
- unsigned int micros(void): μsec 단위 시간 반환
- void delay(unsigned int howLong): howLong msec 만큼 대기
- void delayMicroseconds(unsigned int howLong): howLong μsec 만큼 대기 → wPi 라이브러리 정의 함수¹

```
int main(int argc, char* argv[]){
    wiringPiSetupGpio(); unsigned int T, L;
    pinMode(PIN_TRIGGER, OUTPUT); pinMode(PIN_ECHO, INPUT);
```

```
    while (1) {
        digitalWrite(PIN_TRIGGER, LOW); delayMicroseconds(2);
        digitalWrite(PIN_TRIGGER, HIGH); delayMicroseconds(20);
        digitalWrite(PIN_TRIGGER, LOW);
```

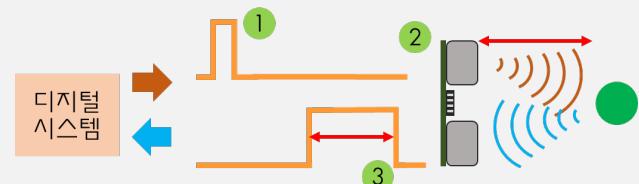
```
        while (digitalRead(PIN_ECHO) == LOW);
```

```
        unsigned int startTime = micros();
        while (digitalRead(PIN_ECHO) == HIGH);
        T = micros() - startTime;
        L = T / 58.2;
```

```
        if (L >= RANGE_MAX || L <= RANGE_MIN) cout << "-1" << endl;
        else cout << "Distance is " << L << " cm" << endl;
        delay(100);
    }
```

- 초음파 센서 트리거 출력 (GPIO23)
- 초음파 센서 에코 입력 (GPIO24)
- 최대/최소 측정 거리 설정 (0-200 cm)

- 측정 거리 (L)와 측정 시간 (T) 변수 설정
- 트리거는 출력, 에코는 입력 모드 설정



- 방사한 초음파가 물체에 반사되어 Echo 핀으로 들어올 때 까지 대기

- Echo 핀으로 HIGH 값이 유지된 시간 측정
- 측정 거리 (L)와 시간 (T) 관계 수식

- 결과가 최대 (200 cm) 또는 최소 측정 거리 (0 cm) 밖일 때 에러를 의미하는 '-1' 출력
- 100 ms 동안 대기

초음파 센서 프로그램 실행

Practice

```
pi@raspberrypi: ~/ex  
pi@raspberrypi:~/ex $ g++ ./XX_Sensor_ultrasound.cpp -o XX_Sensor_ultrasound -lwiringPi  
pi@raspberrypi:~/ex $ ./XX_Sensor_ultrasound  
Distance is 8 cm  
Distance is 7 cm  
Distance is 8 cm  
Distance is 8 cm  
Distance is 8 cm  
Distance is 8 cm  
Distance is 9 cm  
Distance is 10 cm  
Distance is 12 cm  
Distance is 13 cm  
Distance is 113 cm  
-1  
Distance is 126 cm  
Distance is 137 cm  
Distance is 103 cm  
Distance is 16 cm  
Distance is 14 cm  
Distance is 14 cm  
Distance is 14 cm  
Distance is 13 cm  
Distance is 12 cm
```

•측정 거리 출력

•-lwiringPi 옵션으로 wPi 라이브러리 위치를 지정해서 컴파일

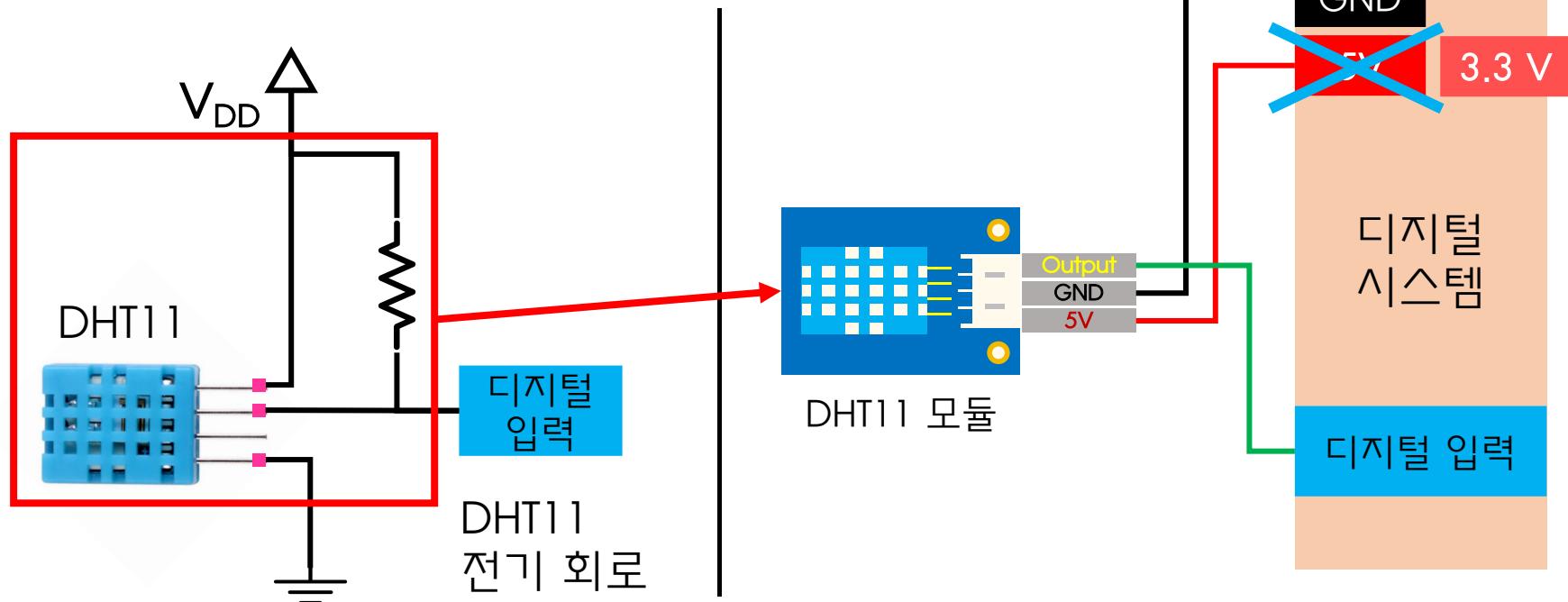
온습도 센서

DHT11

온습도 센서

■ DHT11¹ (DHT11 모듈²)

- 서미스터 (thermistor)³, 커패시터형 (capacitive) 습도 센서로 구성
- 디지털 인터페이스
- 1-2 초 마다 새로운 데이터 측정
- 고급형 온습도 센서 DHT22



¹ <https://www.eleparts.co.kr/goods/view?no=3353799>

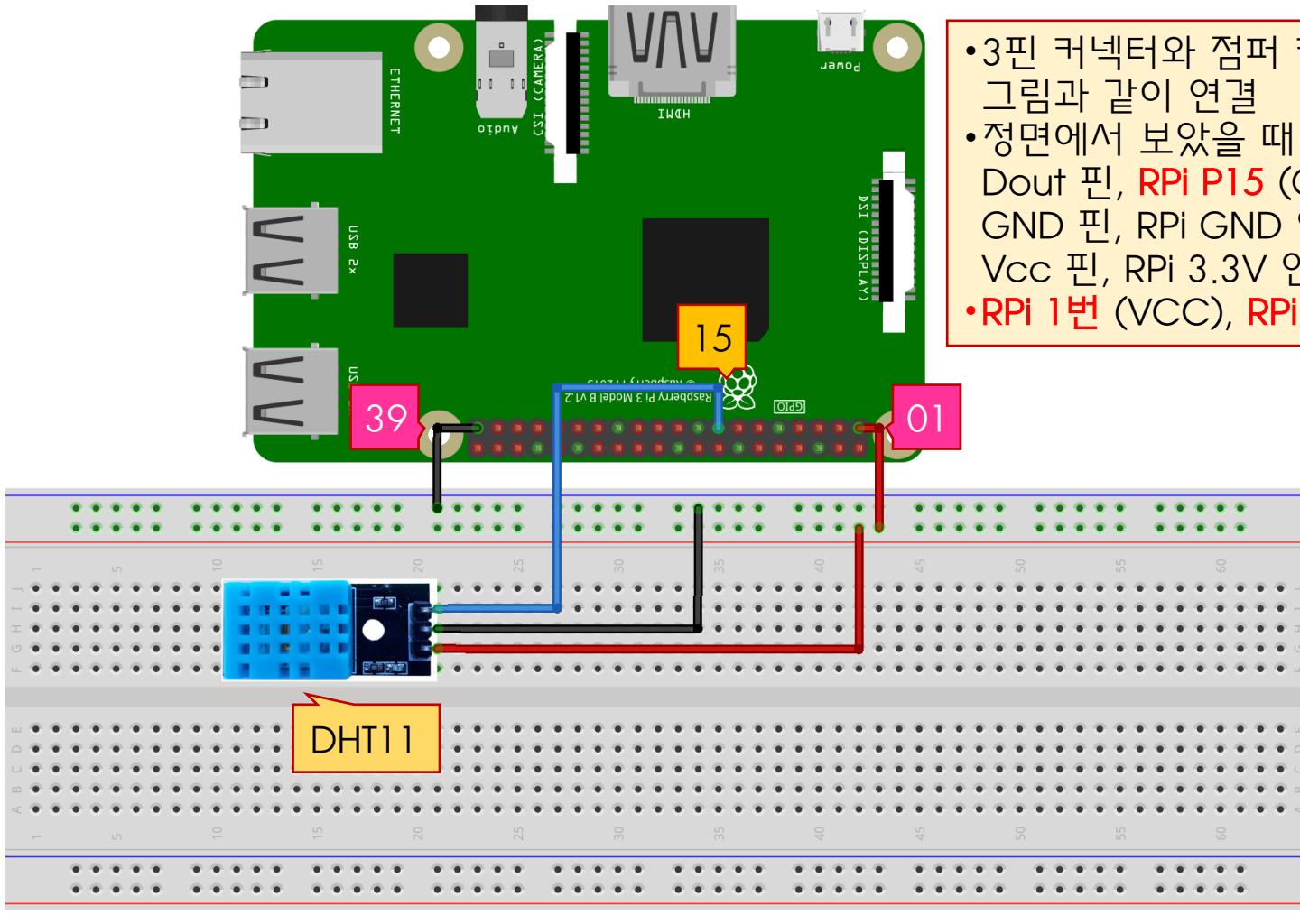
² <https://www.eleparts.co.kr/goods/view?no=3118109>

³ 온도에 따라 저항이 변하는 센싱 소자를 활용한 센서, 열가변저항이라고 부르기도 함

온습도 센서 회로 구성

Practice

■ wPi를 이용한 온도 센서 읽기



- 3핀 커넥터와 점퍼 케이블을 이용하여 그림과 같이 연결
- 정면에서 보았을 때 (데이터시트 확인!), Dout 핀, RPi P15 (GPIO22) 연결
GND 핀, RPi GND 연결
Vcc 핀, RPi 3.3V 연결
- RPi 1번 (VCC), RPi 39번 (GND) 구성

온습도 센서 wPi 프로그래밍

Practice

```
#include <iostream>
#include <unistd.h>
#include <wiringPi.h>
#include <iomanip>
using namespace std;
```

- DHT11 데이터를 읽기 위한 절차를 C++로 구현
- 타 언어 (예, Java)로 프로그램을 제작 하기위해서는 동일한 절차를 해당 언어와 wPi 대체 라이브러리를 이용하여 구현 필요

- DHT11을 사용하기 위해 **false**를 **true**로 바꿈

```
#define USING_DHT11 true // The DHT11 uses only 8 bits
#define DHT_GPIO 22 // Using GPIO 22 for this example
#define LH_THRESHOLD 26 // Low=~14, High=~38 - pick avg.
```

```
int main(){
    int humid = 0, temp = 0;
    cout << "Starting the one-wire sensor program" << endl;
    wiringPiSetupGpio();
    piHiPri(99);
TRYAGAIN:    // If checksum fails (come back here)
    unsigned char data[5] = {0,0,0,0,0};
    pinMode(DHT_GPIO, OUTPUT);      // gpio starts as output
    digitalWrite(DHT_GPIO, LOW);    // pull the line low
    usleep(18000);                // wait for 18ms
    digitalWrite(DHT_GPIO, HIGH);   // set the line high
    pinMode(DHT_GPIO, INPUT);     // now gpio is an input
```

온습도 센서 프로그램 실행

Practice

```
pi@raspberrypi: ~/exploringrp1/chp06/dht
```

```
pi@raspberrypi:~/exploringrp1/chp06/dht $ ls  
build dht dht.cpp  
pi@raspberrypi:~/exploringrp1/chp06/dht $ nano dht.cpp  
pi@raspberrypi:~/exploringrp1/chp06/dht $  
pi@raspberrypi:~/exploringrp1/chp06/dht $ g++ dht.cpp -o dht  
pi@raspberrypi:~/exploringrp1/chp06/dht $ ls  
build dht dht.cpp  
pi@raspberrypi:~/exploringrp1/chp06/dht $ ./dht  
Starting the one-wire sensor program  
The checksum is good  
The temperature is 26°C  
The humidity is 21%  
pi@raspberrypi:~/exploringrp1/chp06/dht $
```

- 소스 코드에서 DHT11 사용을 위해 수정
- **#define USING_DHT11 true**

- -lwiringPi 옵션으로 wPi 라이브러리 위치를 지정해서 컴파일

- 온도와 습도 출력 값 확인
- 온습도 값을 활용 하는 프로그램 실습

VNC

Virtual network computing

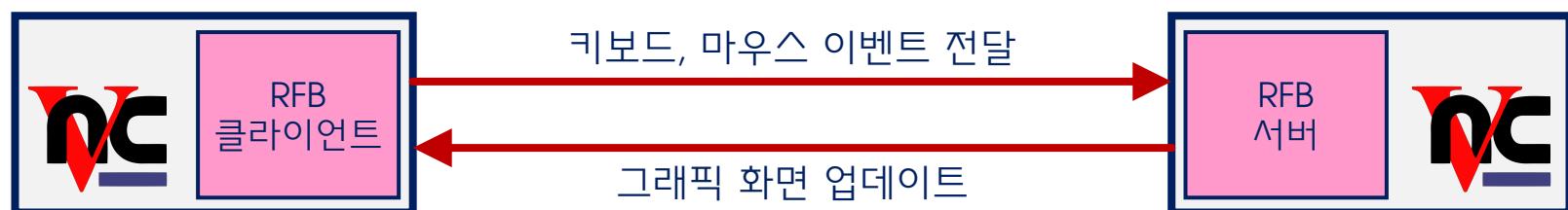
VNC 가상 네트워크 컴퓨팅

■ VNC (virtual network computing)

- 원격 프레임 버퍼 (RFB: remote frame buffer) 프로토콜을 사용하는 '**그래픽 데스크톱 공유 시스템** (graphical desktop-sharing system)'으로서 원격으로 다른 컴퓨터를 접속하기 위해 사용
 - ✓ 로컬 컴퓨터에서 일어난 '키보드와 마우스 이벤트'를 원격 접속한 컴퓨터에 전달,
 - ✓ 원격 접속한 컴퓨터에서 일어난 '그래픽 화면 업데이트' 내용을 다시 로컬 컴퓨터에게 되돌려 전송

■ VNC의 장점은,

- 원격 프레임 버퍼 (컴퓨터 RAM에 존재하는 비트맵 이미지 데이터)를 사용하기 때문에 **특정 OS에 종속적이지 않음**
- **물리적 디스플레이 없이** 원격으로 그래픽 데스크톱을 공유하여 사용 가능

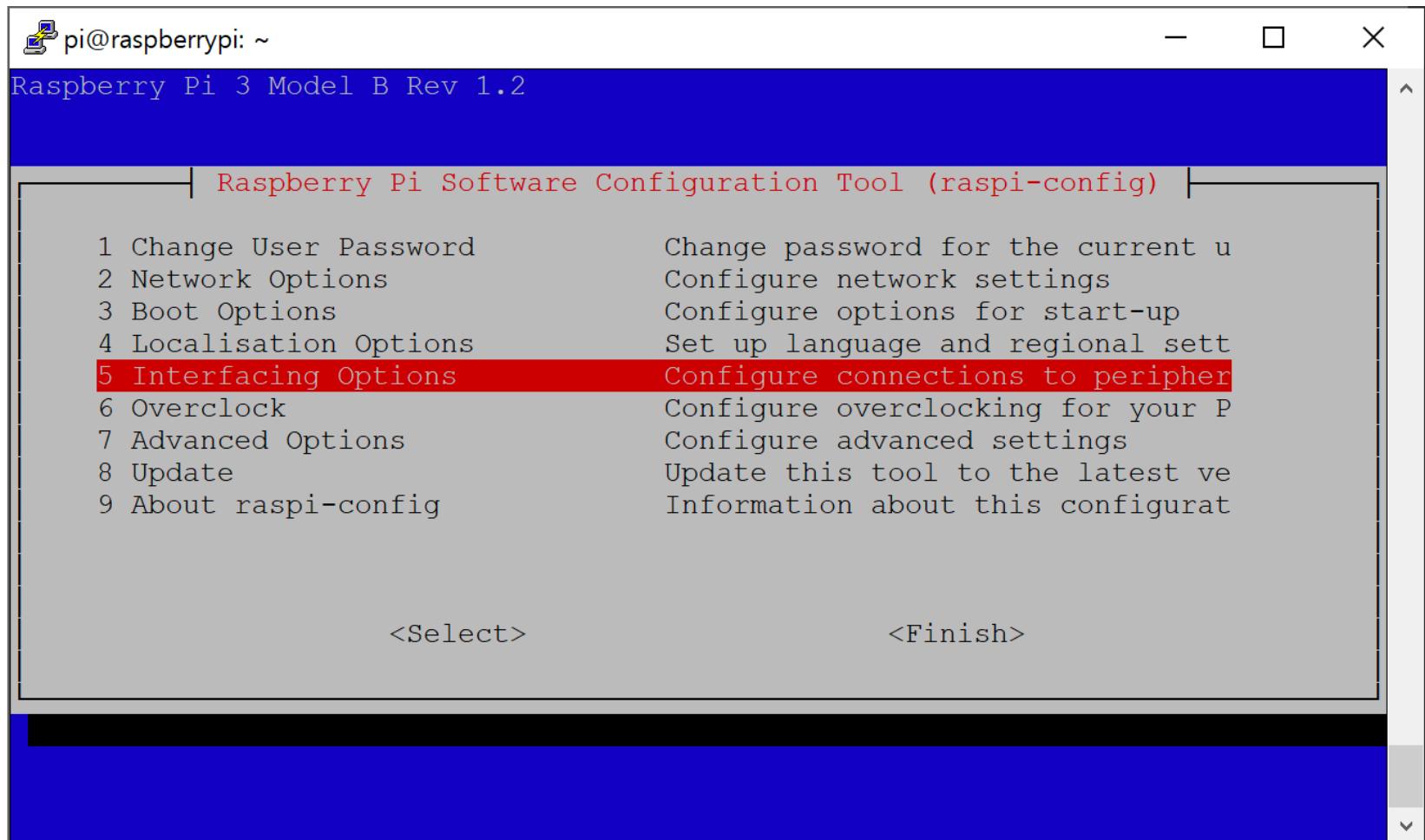


RPi VNC 설정

raspi-config

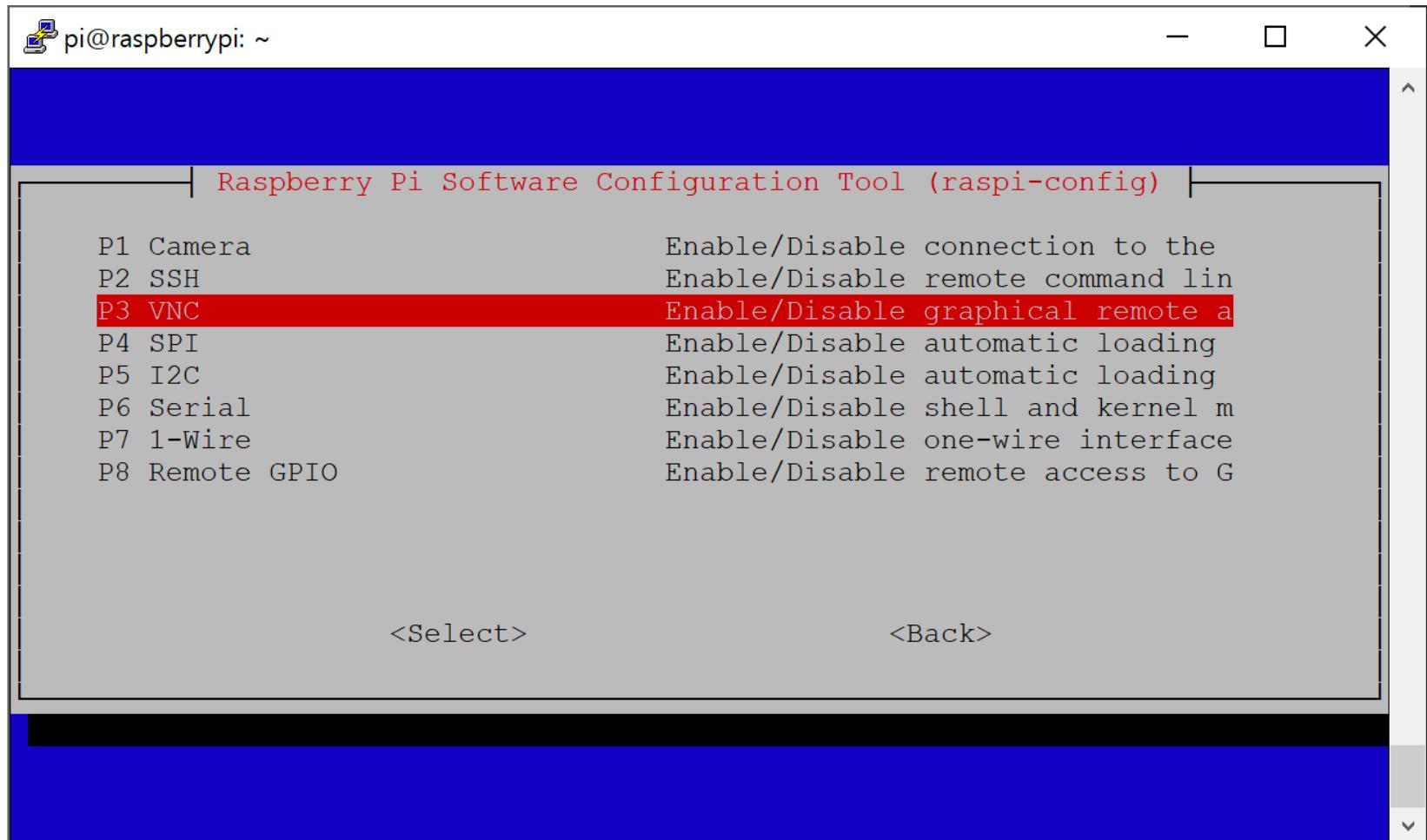
RPi VNC 활성화

■ sudo raspi-config



RPi VNC 활성화

■ sudo raspi-config



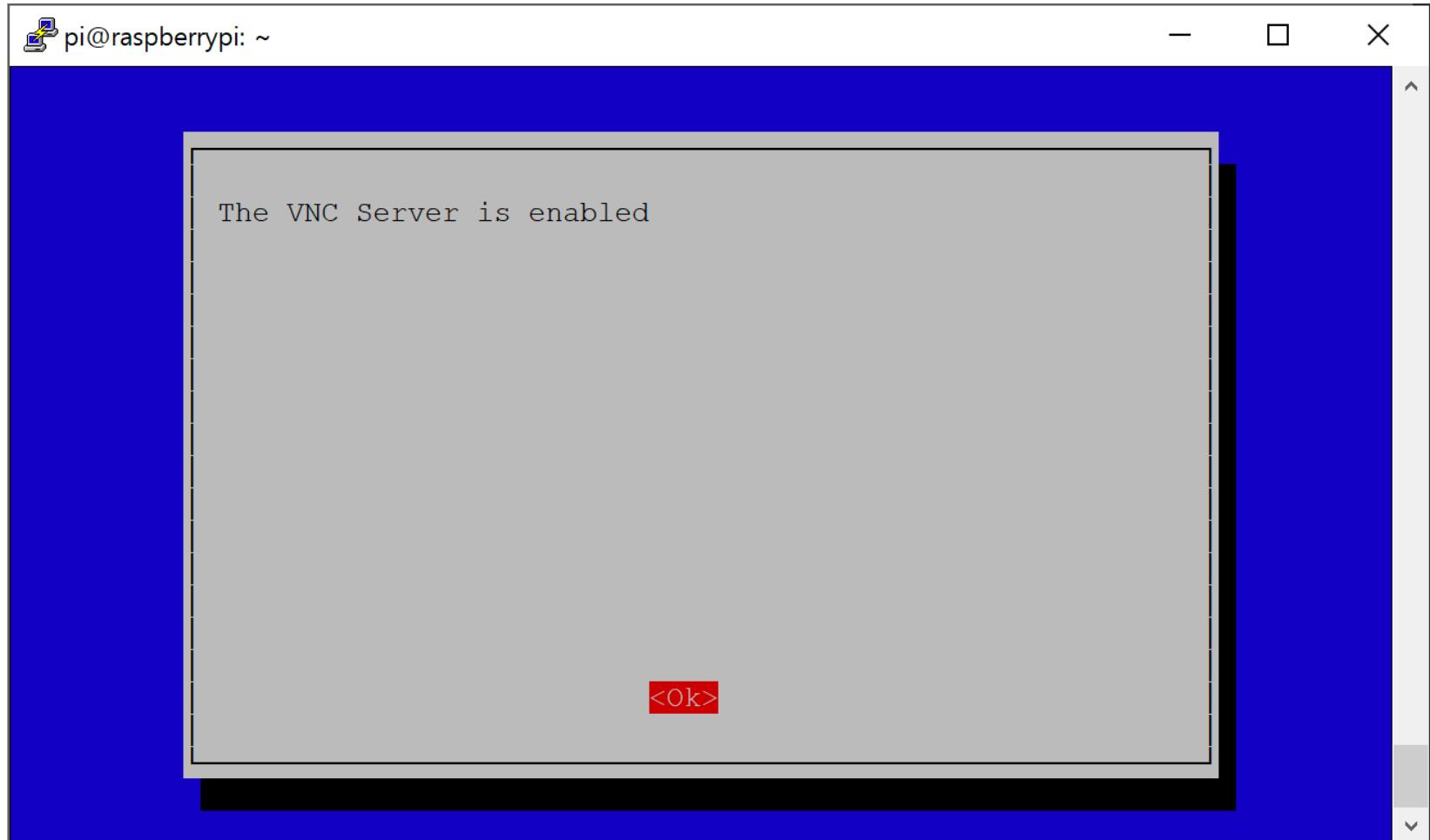
RPi VNC 활성화

■ sudo raspi-config



RPi VNC 활성화

■ sudo raspi-config



RPi VNC 활성화

■ sudo raspi-config

```
pi@raspberrypi: ~
```

```
pi@raspberrypi:~ $ ps aux | grep vnc
pi      7159  0.0  0.0  4368   560 pts/0    S+   21:10  0:00 grep --color=auto vnc
pi@raspberrypi:~ $ sudo raspi-config
```

```
Created symlink /etc/systemd/system/multi-user.target.wants/vncserver-x11-serviced.service → /usr/lib/systemd/system/vncserver-x11-serviced.service.
```

```
pi@raspberrypi:~ $ ps aux | grep vnc
root    7190  0.2  0.2  5688  2748 ?        Ss    21:10  0:00 /usr/bin/vncserver-x11-serviced -fg
root    7191  2.2  1.2 22492 11660 ?        S     21:10  0:00 /usr/bin/vncserver-x11-core -service
root    7201  0.8  0.4   8528  4396 ?        S     21:10  0:00 /usr/bin/vncagent service 14
pi      7203  8.6  1.3 28108 12524 ?        S     21:10  0:00 /usr/bin/vncserverui service 15
pi      7211  3.5  1.4 27424 14192 ?        S     21:10  0:00 /usr/bin/vncserverui -statusicon 7
pi      7219  0.0  0.0   4368   540 pts/0    S+   21:10  0:00 grep --color=auto vnc
pi@raspberrypi:~ $
```

- VNC 관련 프로세스 검사
- **ps aux | grep vnc**

- VNC 관련 실행 프로세스 없음

- VNC 서비스 활성화 이후 프로세스 리스트

VNC 클라이언트

RealVNC

VNC 클라이언트

■ VNC 클라이언트 란?

- 원격 컴퓨터 접속을 위해 로컬 컴퓨터에 설치하는 애플리케이션으로서 '**VNC 뷰어 (viewer)**'라고도 부름
- 수많은 유료/무료 프로그램들이 공개되어 있으며 사용하는 OS에 따라 필요한 프로그램을 다운로드하여 설치¹
 - ✓ RealVNC
 - ✓ TigerVNC
 - ✓ TightVNC
 - ✓ UltraVNC
 - ✓ LinkVNC
 - ✓ etc.
- 본 강의에서는 대표적인 무료 VNC 뷰어인 'RealVNC'를 사용

VNC 클라이언트 RealVNC¹

■ RealVNC 다운로드

The screenshot shows the RealVNC Connect download page. At the top, there's a navigation bar with links for Products, Company, Contact us, EN, and Sign in. Below the navigation is a dark header bar with the text "vnc connect" and links for Discover, Pricing, Download, Support, and Partners, along with "Try" and "Buy" buttons.

The main content area features the text "VNC® Connect consists of VNC® Viewer and VNC® Server". Below this, a message says "Download VNC® Viewer to the device you want to control from, below. Make sure you've [installed VNC® Server](#) on the computer you want to control." A row of icons represents different platforms: Windows (highlighted with a red box), macOS, Linux, Raspberry Pi, iOS, Android, Chrome, Solaris, and HP-UX. Below this row, there's a separate icon for AIX. In the bottom right corner of the main content area, there's a blue button labeled "Download VNC Viewer" and a yellow box with the Korean text "클릭!" (Click!).

• 접속한 컴퓨터의
운영체제에 맞는
VNC 뷰어 선택

Download VNC Viewer

클릭!

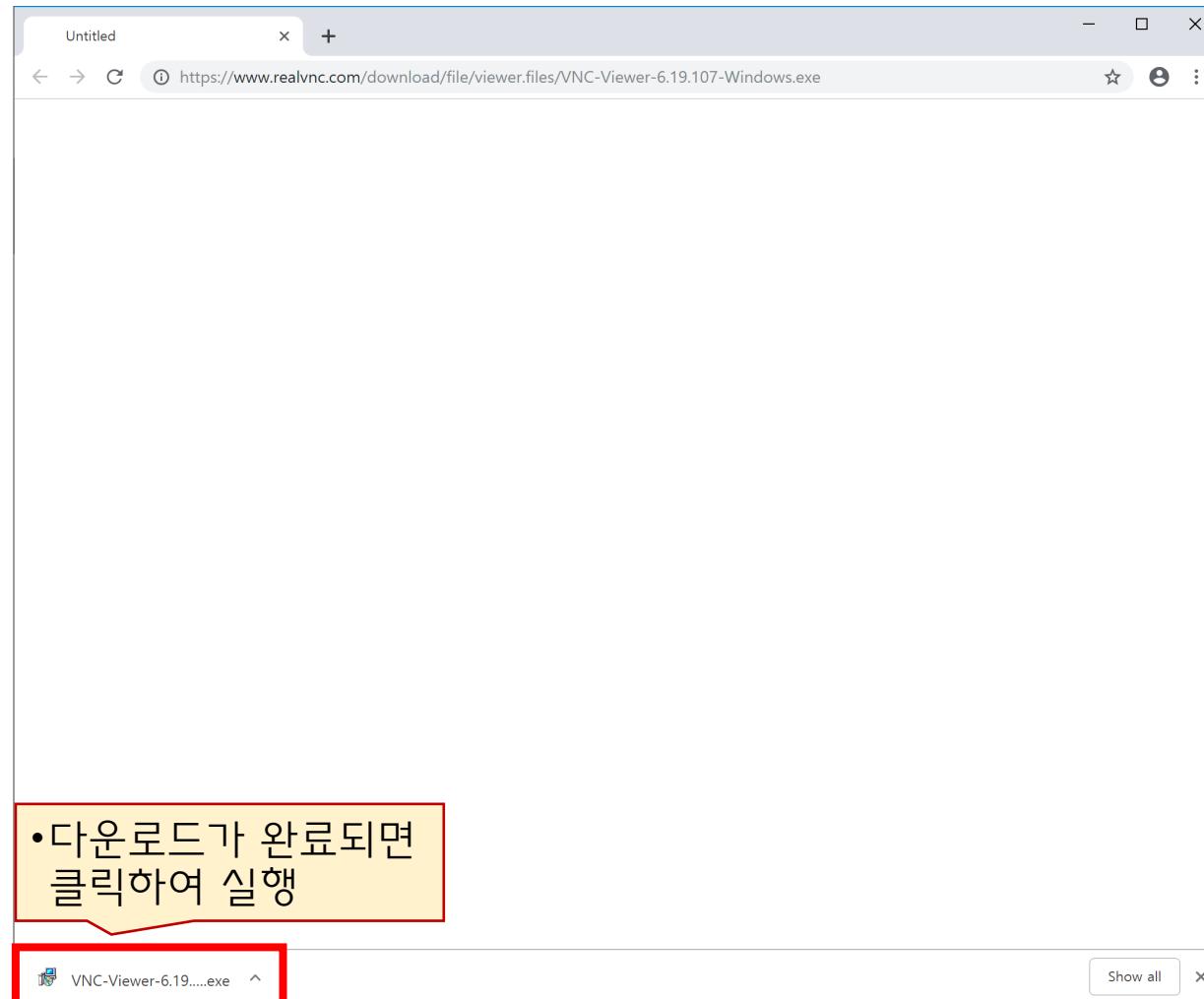
Help

SHA-256: 1f7ea51b3061183fb3b8a49ddc26014439d856eb35417aed0f50332dc868e7c

¹ <https://www.realvnc.com/en/connect/download/viewer/>

VNC 클라이언트 RealVNC¹

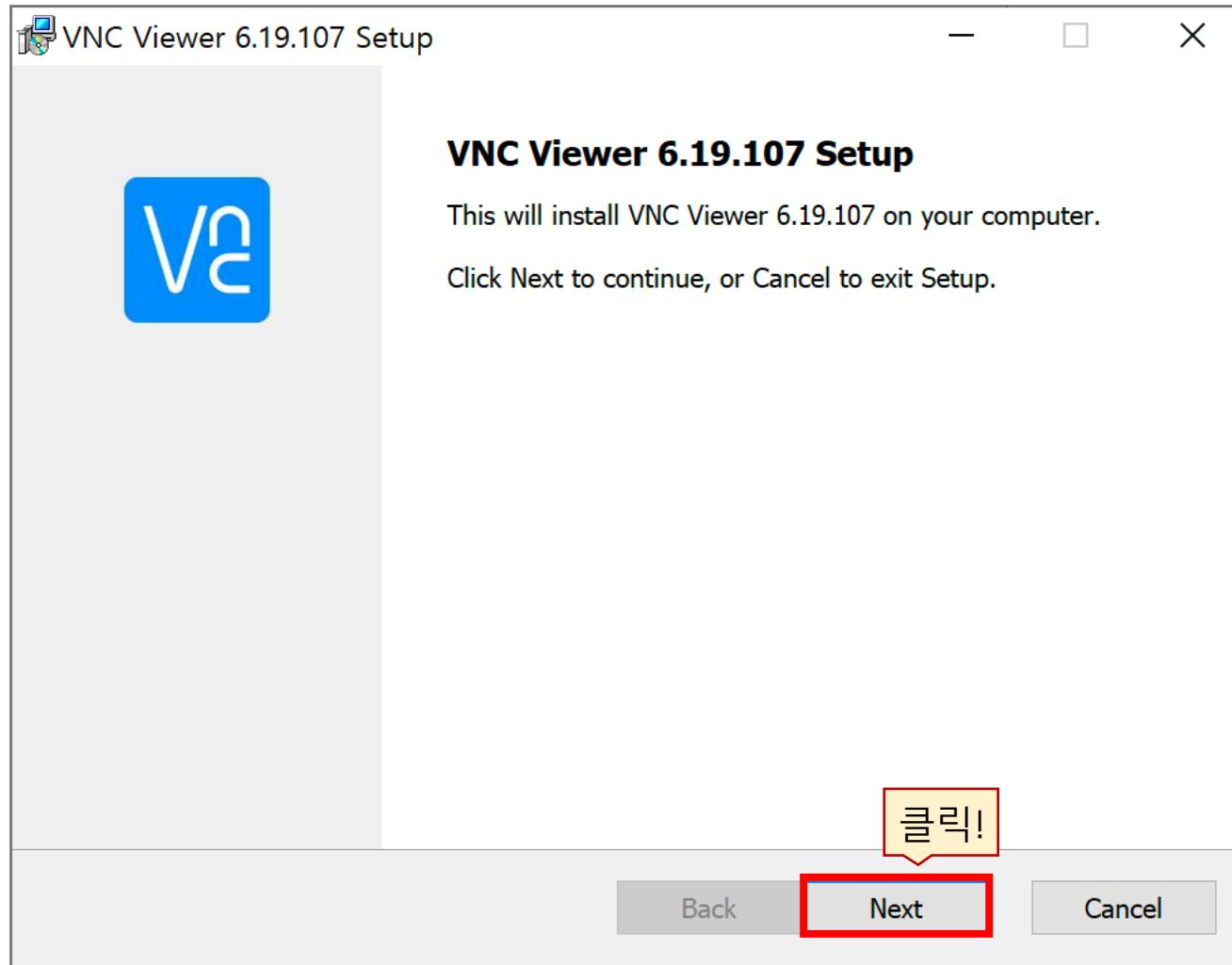
■ RealVNC 다운로드



¹ <https://www.realvnc.com/en/connect/download/viewer/>

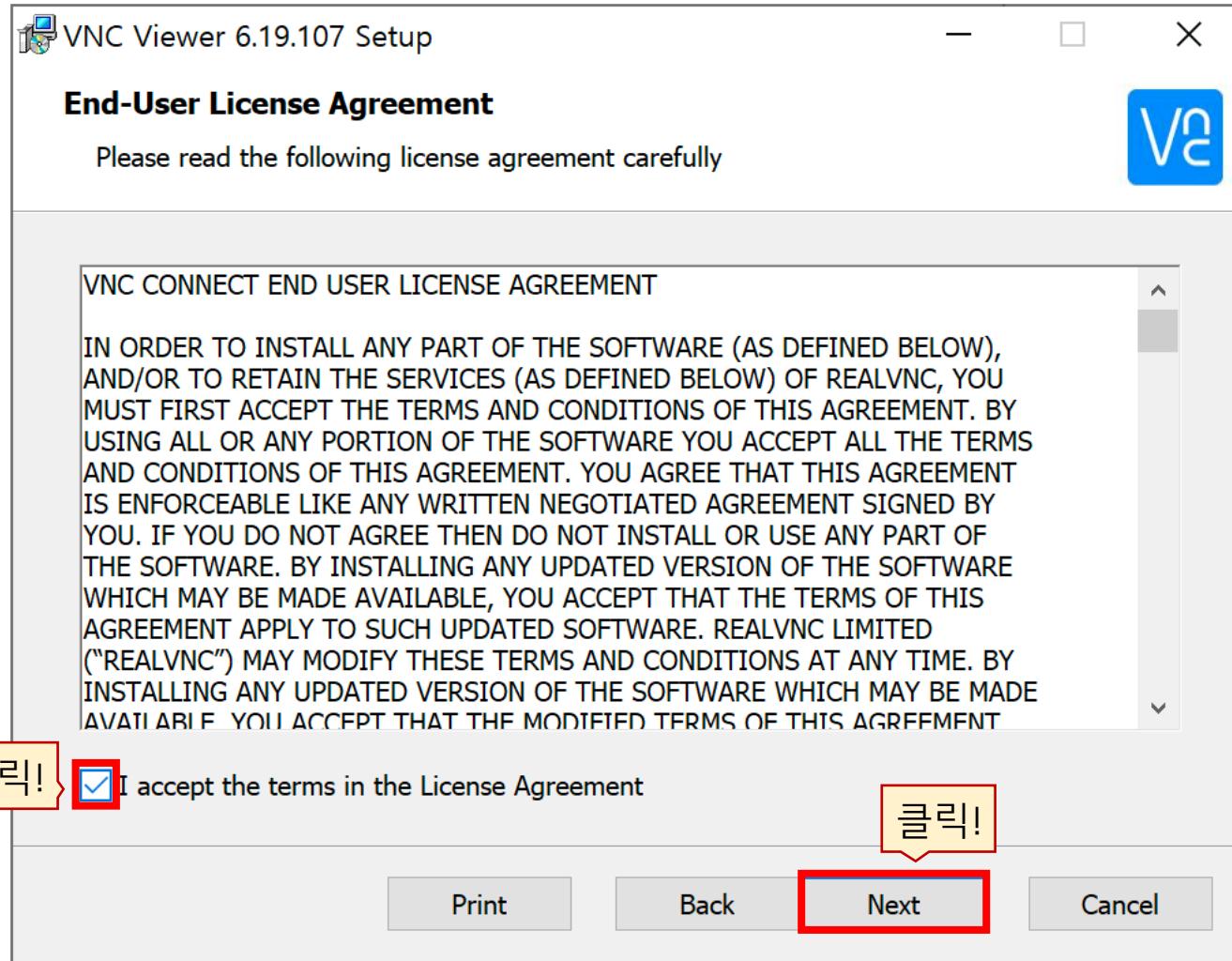
VNC 클라이언트 RealVNC

■ RealVNC 설치



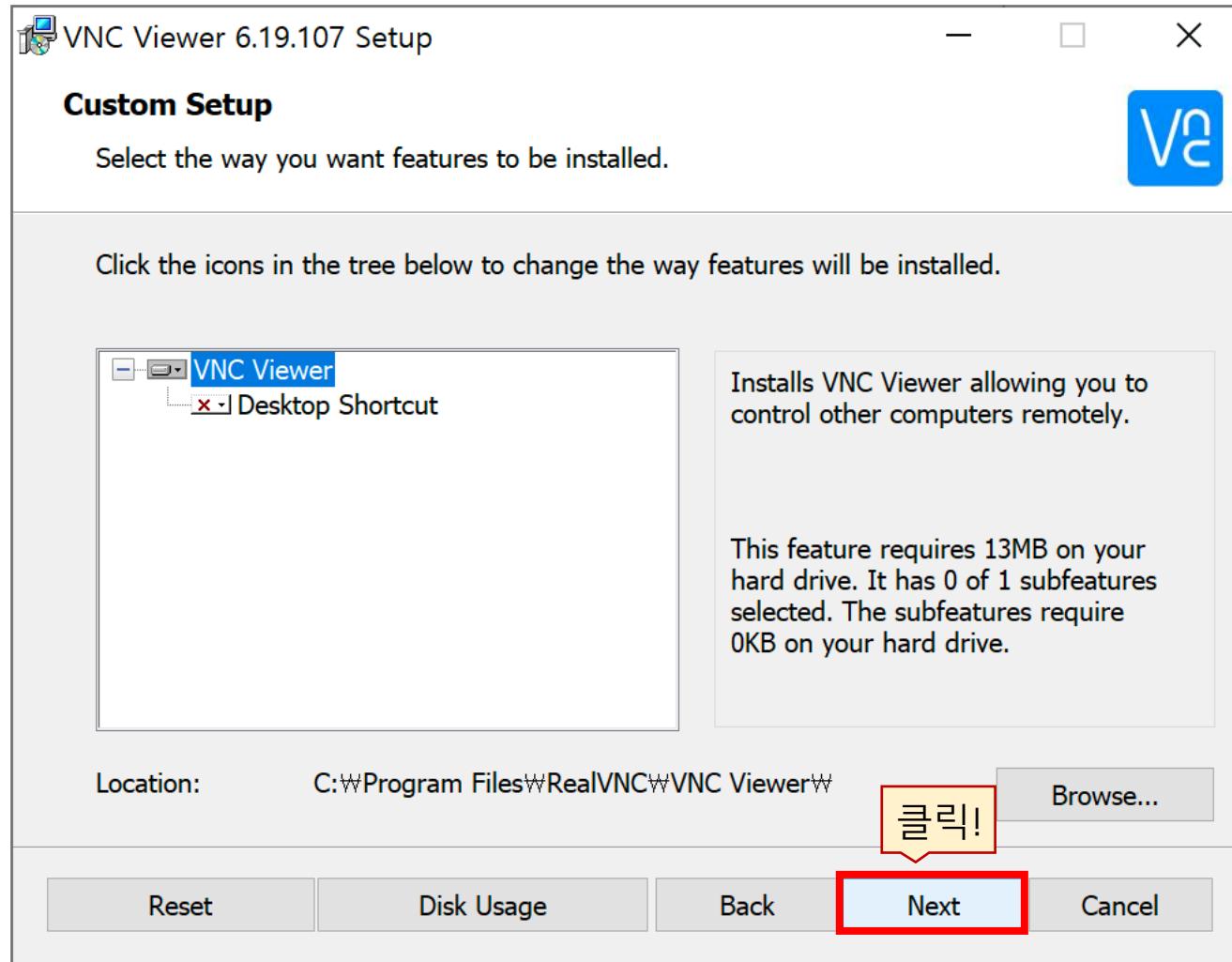
VNC 클라이언트 RealVNC

■ RealVNC 설치



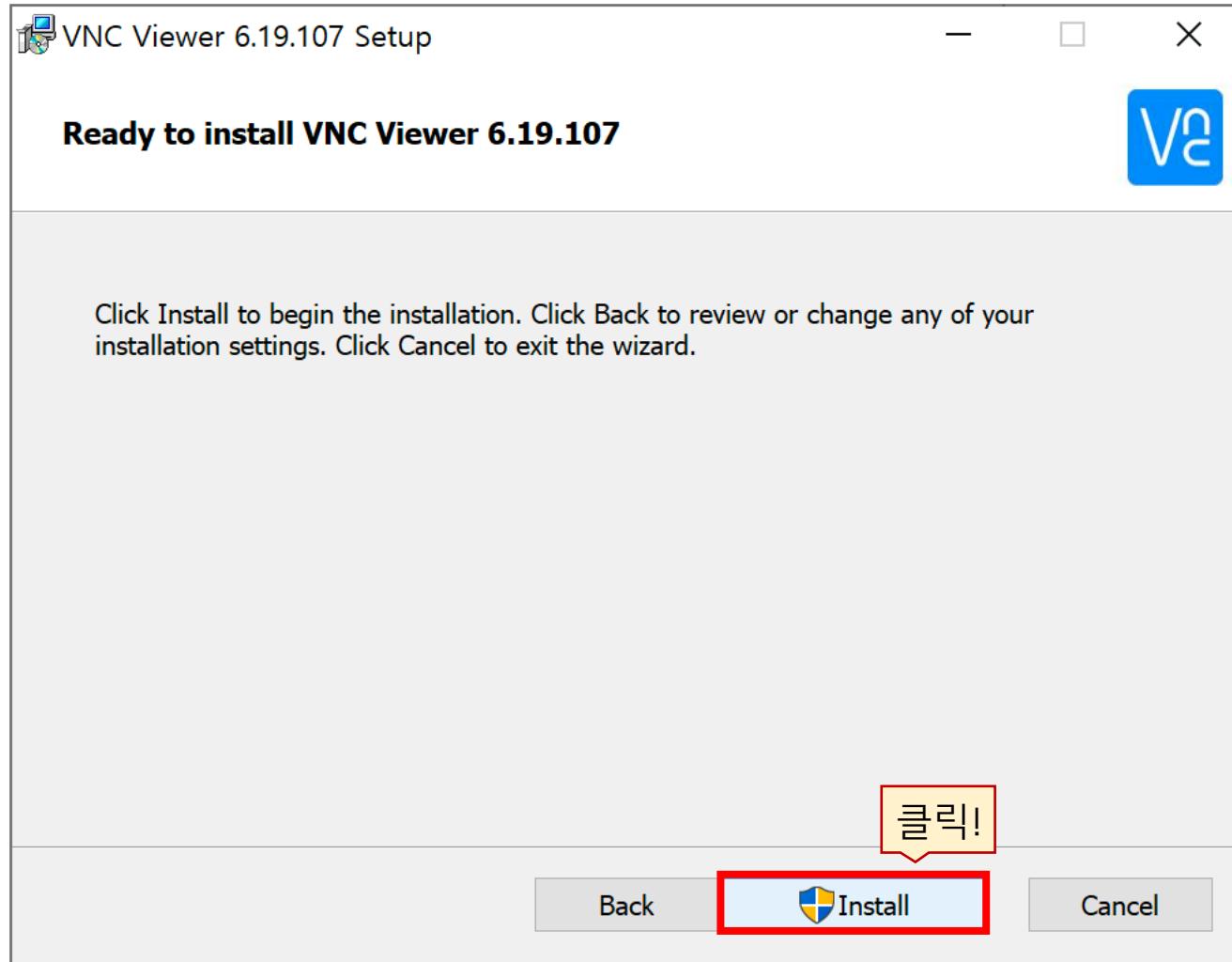
VNC 클라이언트 RealVNC

■ RealVNC 설치



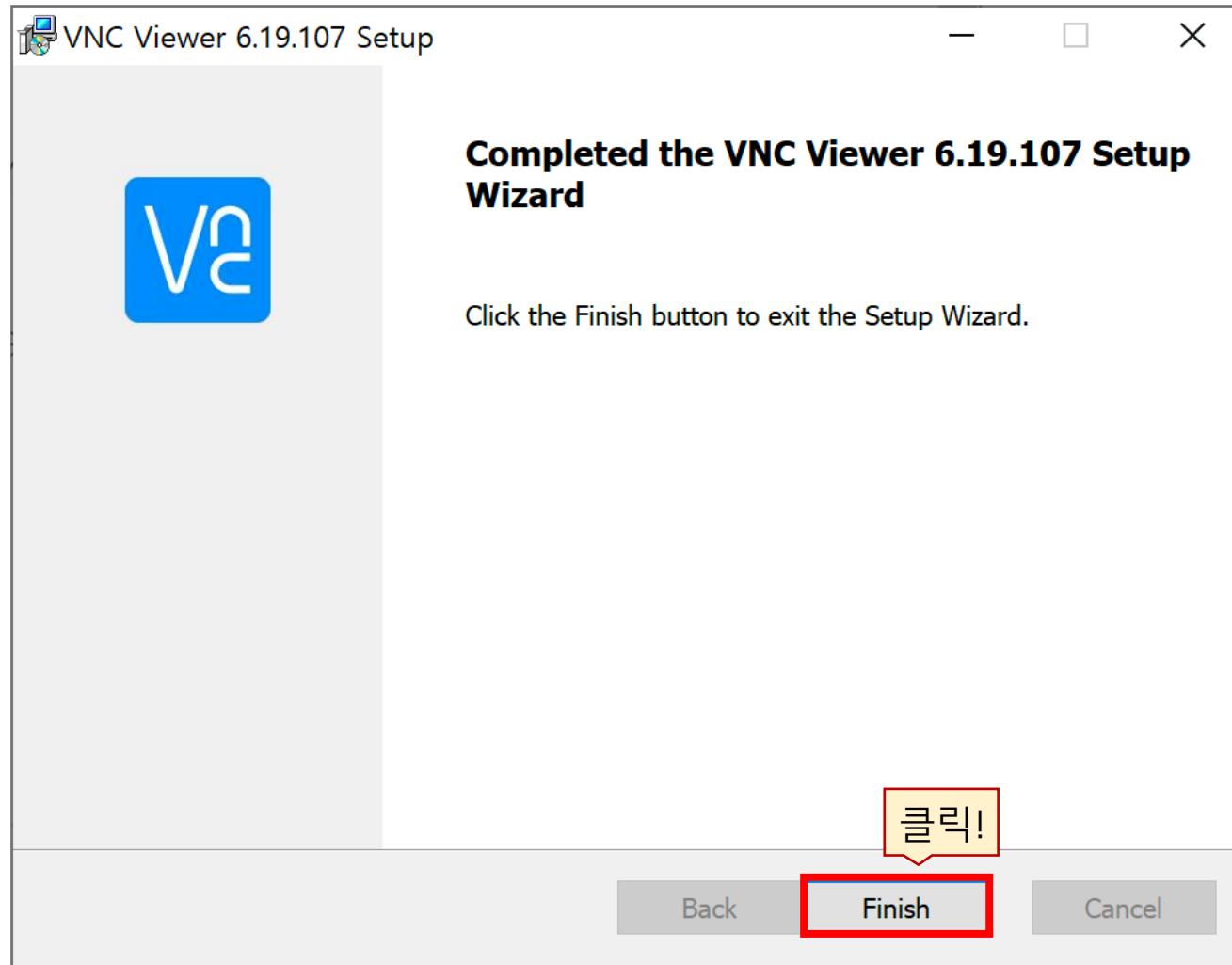
VNC 클라이언트 RealVNC

■ RealVNC 설치



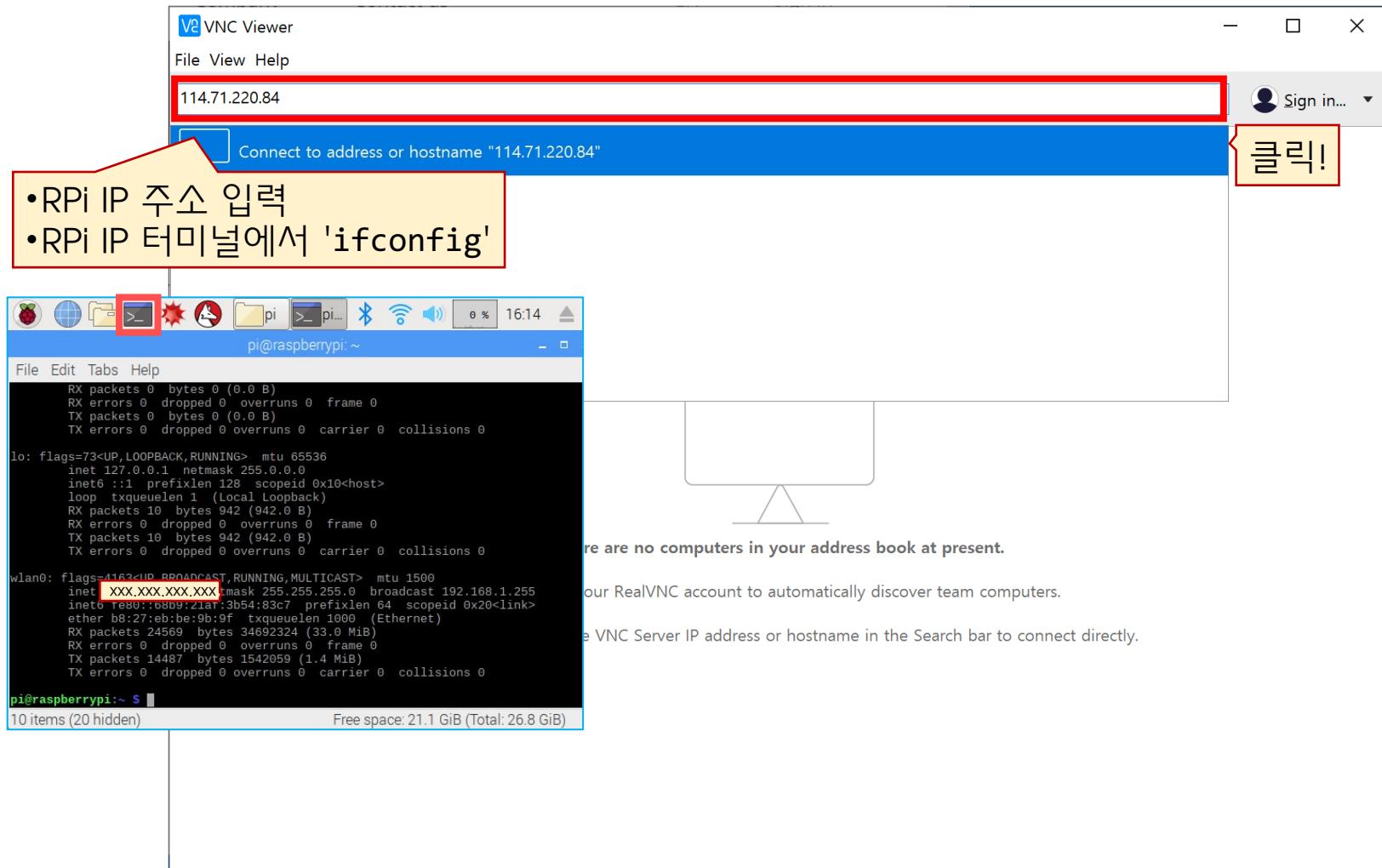
VNC 클라이언트 RealVNC

■ RealVNC 설치



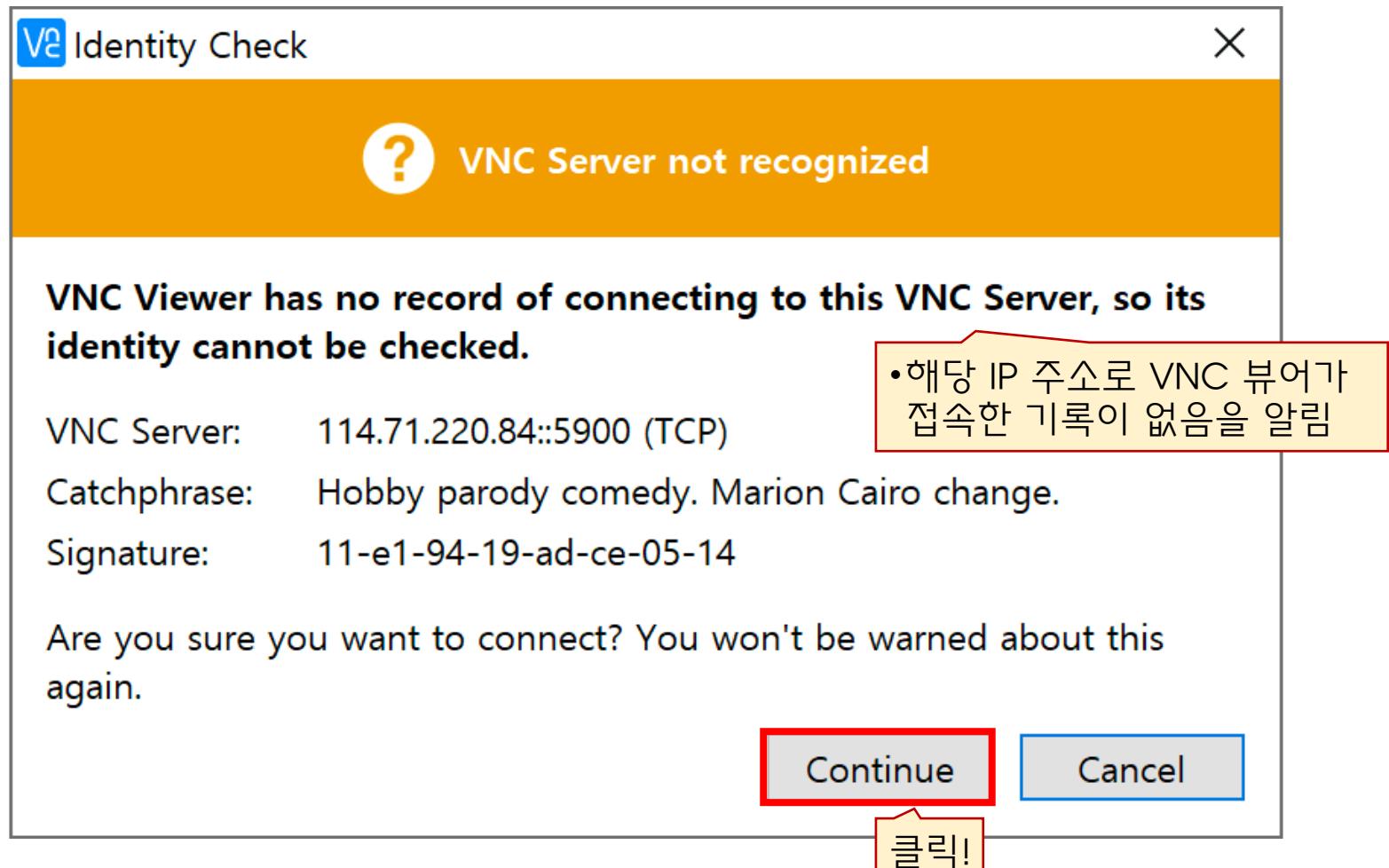
VNC 클라이언트 RealVNC

■ RealVNC 실행



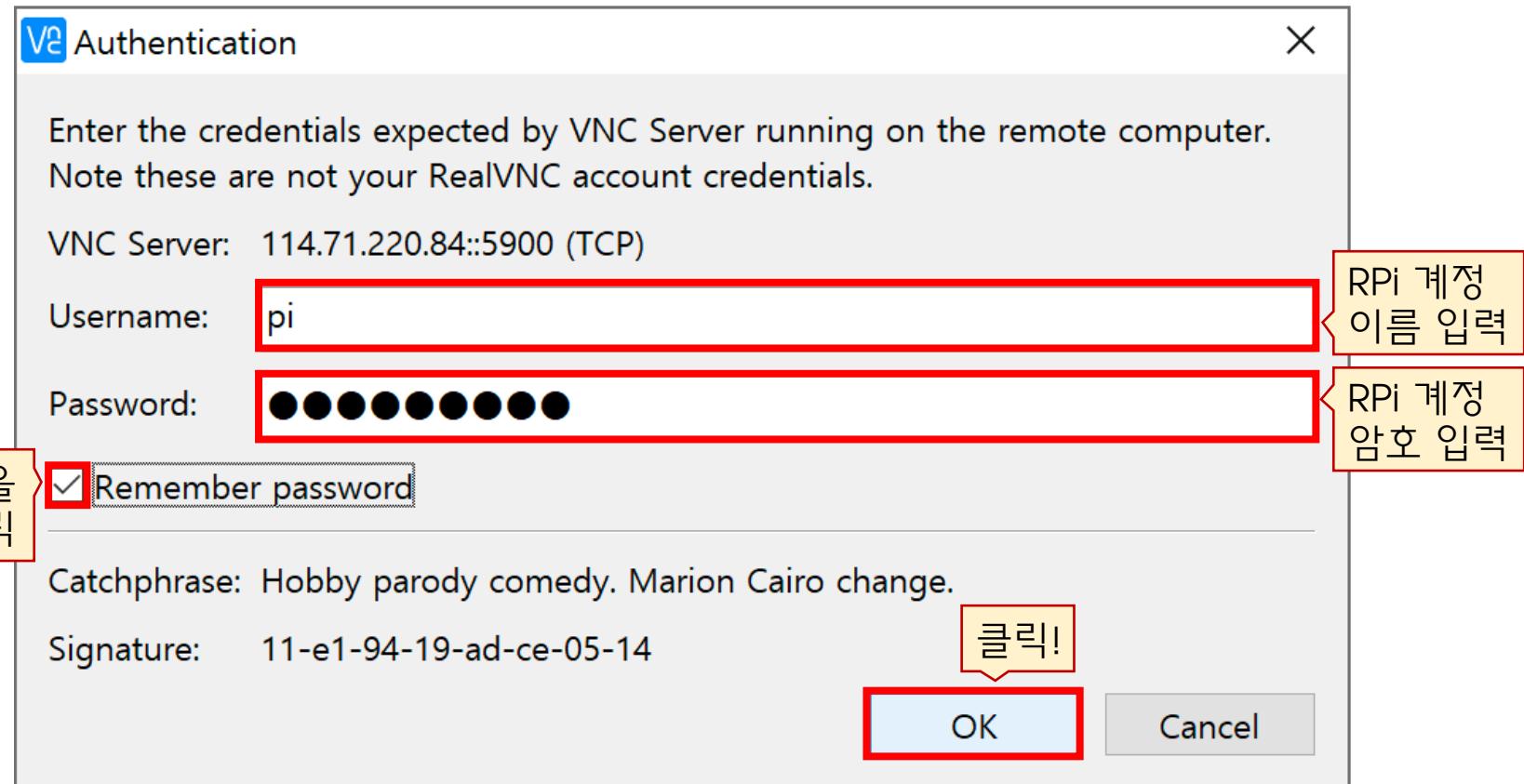
VNC 클라이언트 RealVNC

■ RealVNC 실행



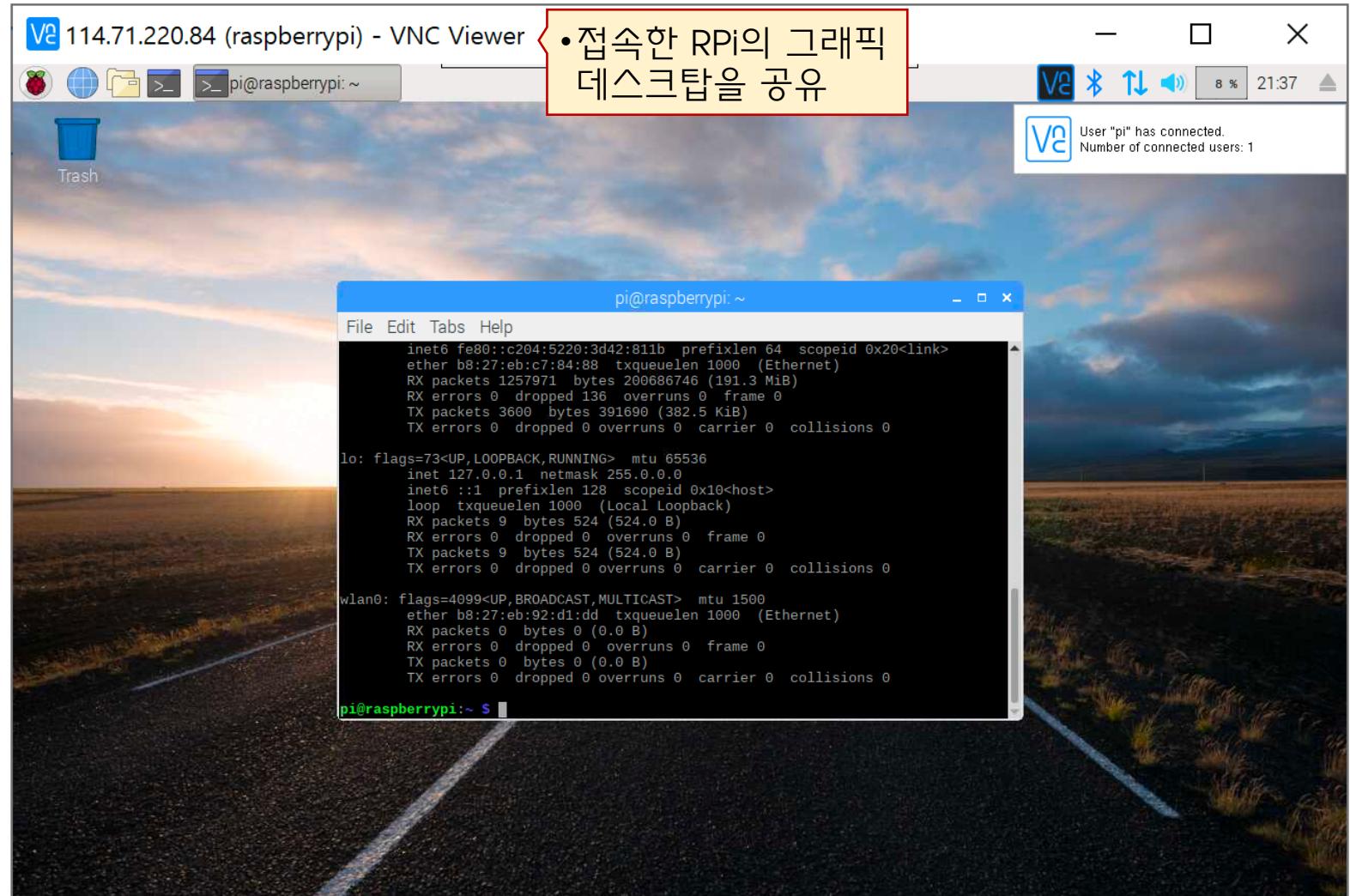
VNC 클라이언트 RealVNC

■ RealVNC 실행



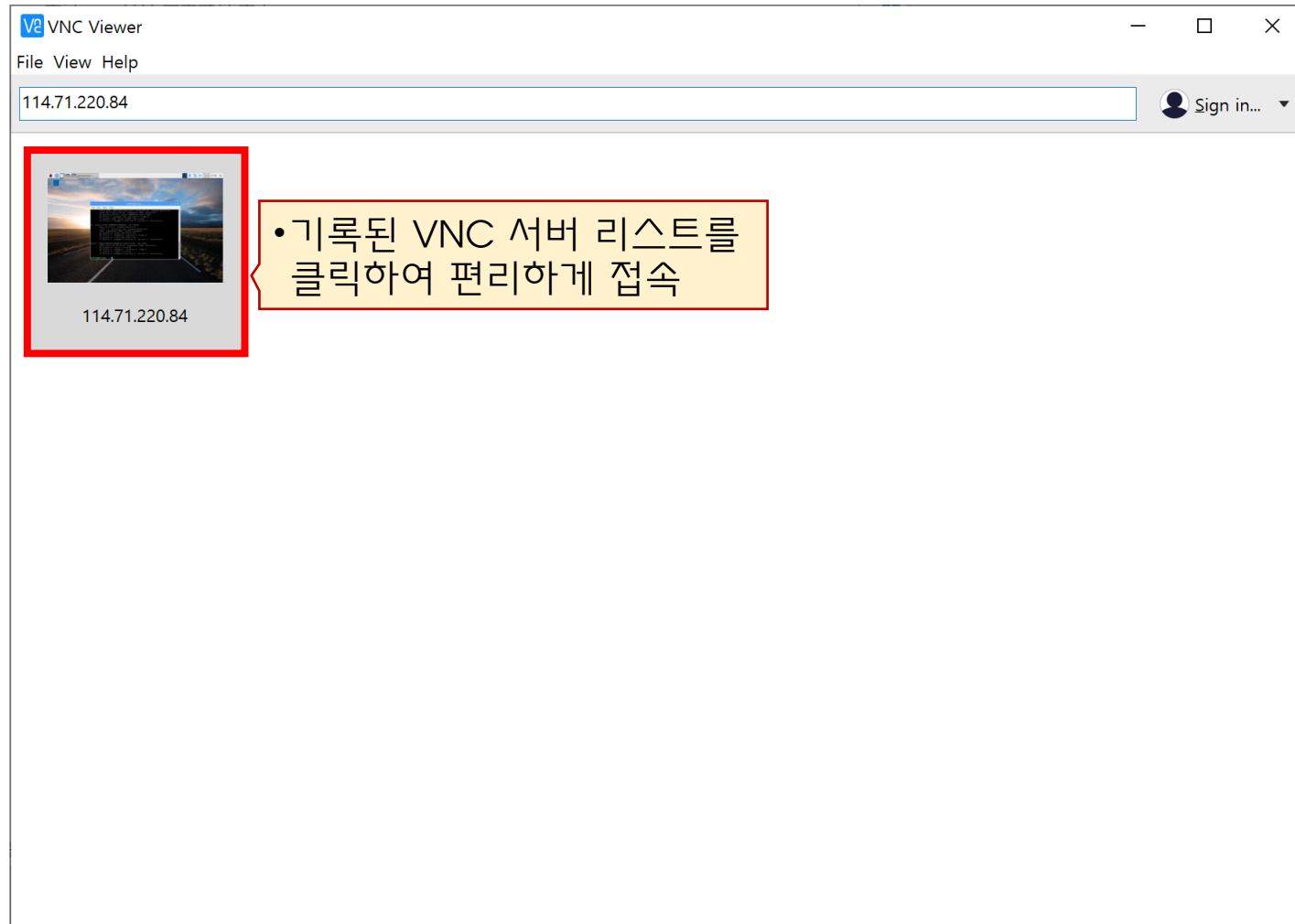
VNC 클라이언트 RealVNC

■ RealVNC 실행



VNC 클라이언트 RealVNC

■ RealVNC 실행



RPi Camera

카메라 모듈

RPi 카메라 소개와 규격

■ RPi 카메라 모듈 V2¹

- RPi 카메라 직렬 인터페이스 (CSI: camera serial interface) 지원
- Sony IMX219 8-메가픽셀 센서
 - ✓ 기존 5-메가픽셀 OmniVision OV5647 센서
- 1080p30, 720p60, VGA90 비디오 모드 지원
- 15 cm 리본 케이블로 RPi CSI 포트에 연결
- MMAL과 V4L API 를 사용하여 카메라 접근
 - ✓ MMAL (multimedia abstraction layer)²: 브로드컴 (Broadcom)에서 RPi Videocore IV GPU 사용을 위해 제작한 C 라이브러리
 - ✓ V4L (Video4Linux)³: 리눅스에서 카메라를 이용해 실시간 비디오 캡처를 지원하기 위한 디바이스 드라이버와 관련 API 모음
- 3rd 파티 라이브러리 지원
 - ✓ 예, Picamera 파이썬 라이브러리⁴



¹ <https://www.raspberrypi.org/products/camera-module-v2/>

² <https://github.com/techyian/MMALSharp/wiki/What-is-MMAL%3F>

³ <https://linuxtv.org/downloads/v4l-dvb-apis/uapi/v4l/v4l2.html>

⁴ <https://picamera.readthedocs.io/en/release-1.13/>

RPi 카메라 연결

■ 카메라 케이블 연결

- RPi 'CSI' 포트 보호 필름 제거

카메라와 리본 케이블

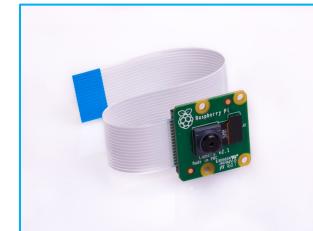


RPi 카메라 연결

■ 카메라 케이블 연결

- RPi 'CSI' 커넥터 하우징 클립 풀기 (당김)

카메라와 리본 케이블



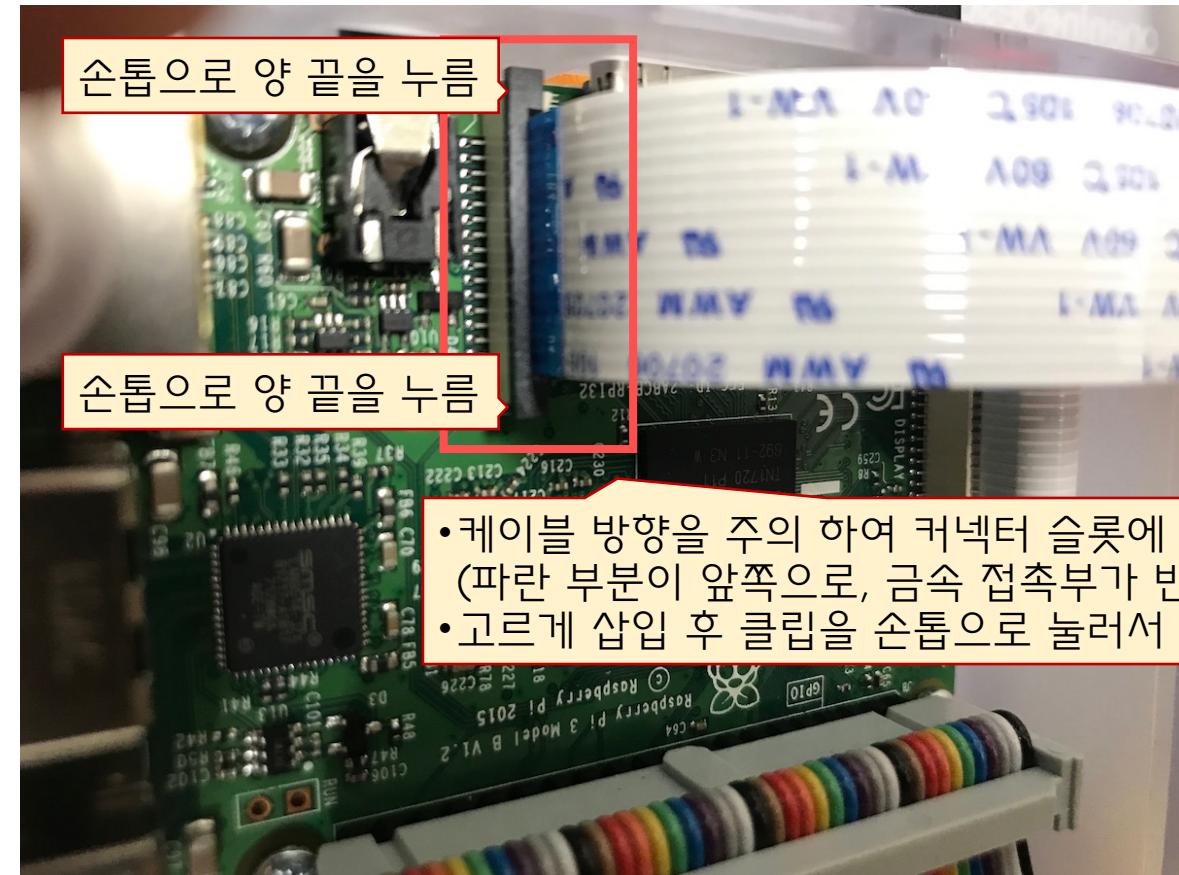
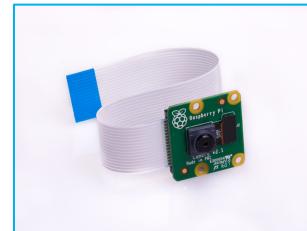
- CSI 커넥터 하우징 클립을 위로 당김
- 케이스가 조립된 상태에서는 **위쪽 부분을 올리기 힘드므로 주의 필요!** (도구 사용)

RPi 카메라 연결

■ 카메라 케이블 연결

- RPi 'CSI' 커넥터 하우징 클립 풀기 (당김)

카메라와 리본 케이블

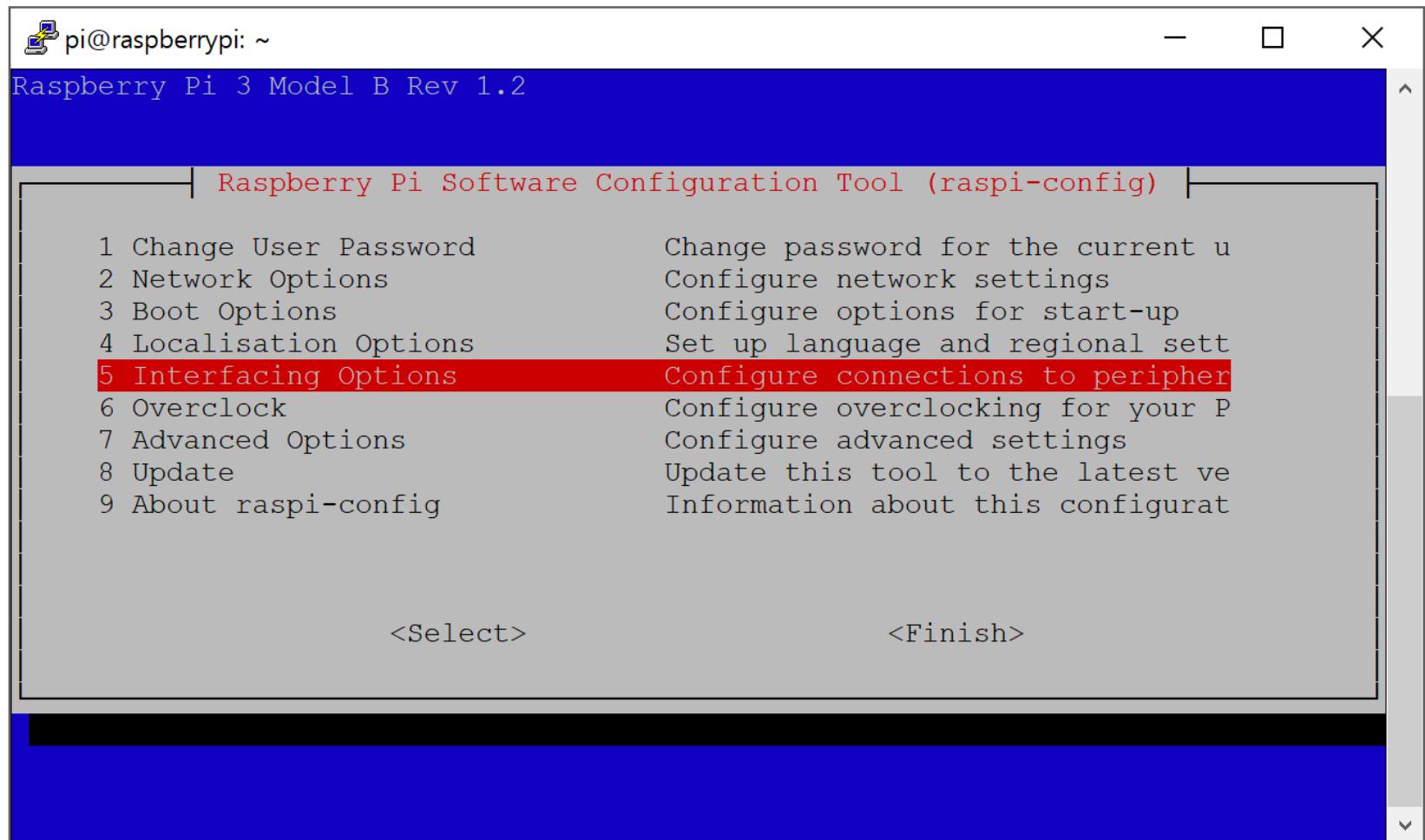


RPi Camera 설정

raspi-config

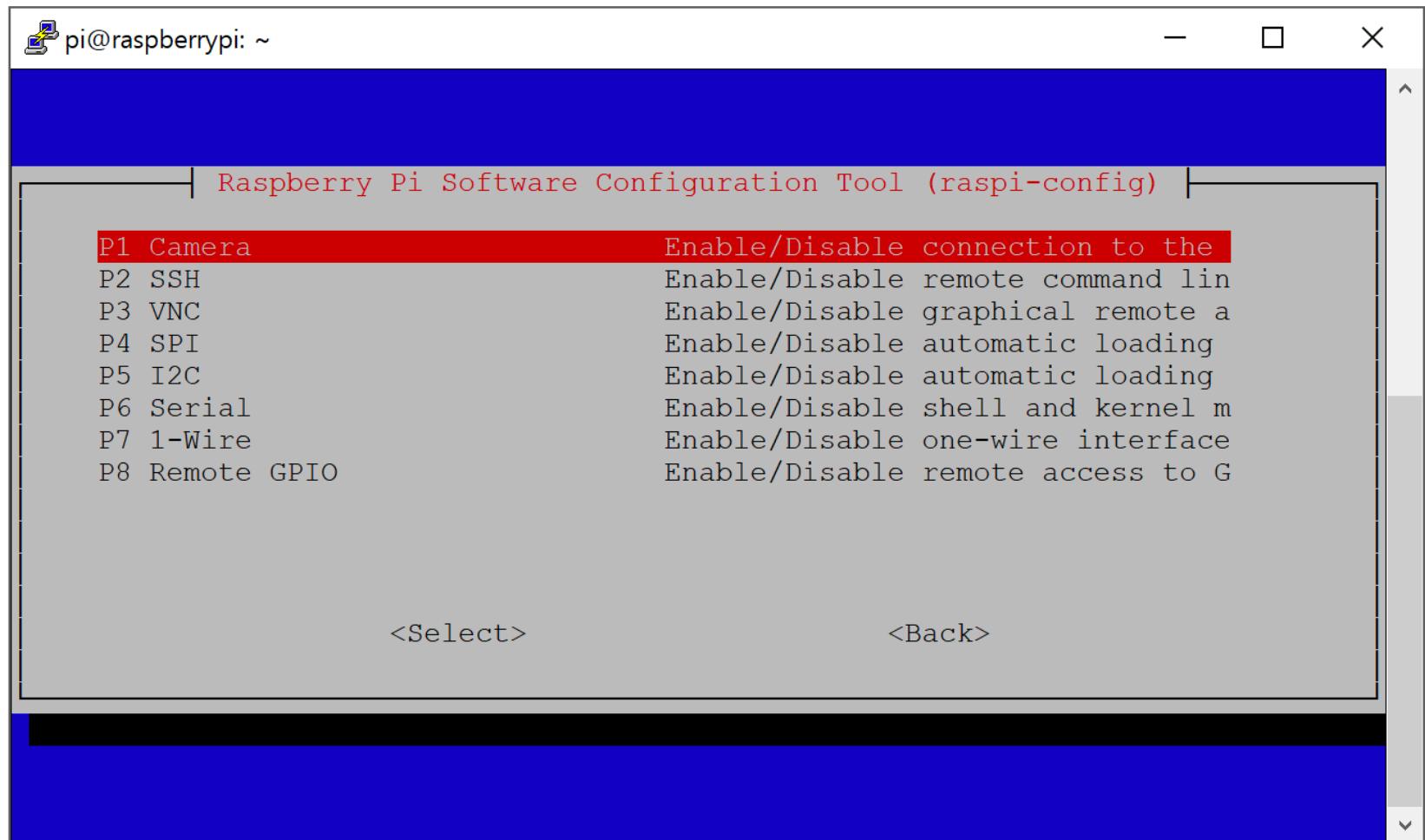
RPi Camera 활성화

■ sudo raspi-config



RPi Camera 활성화

■ sudo raspi-config



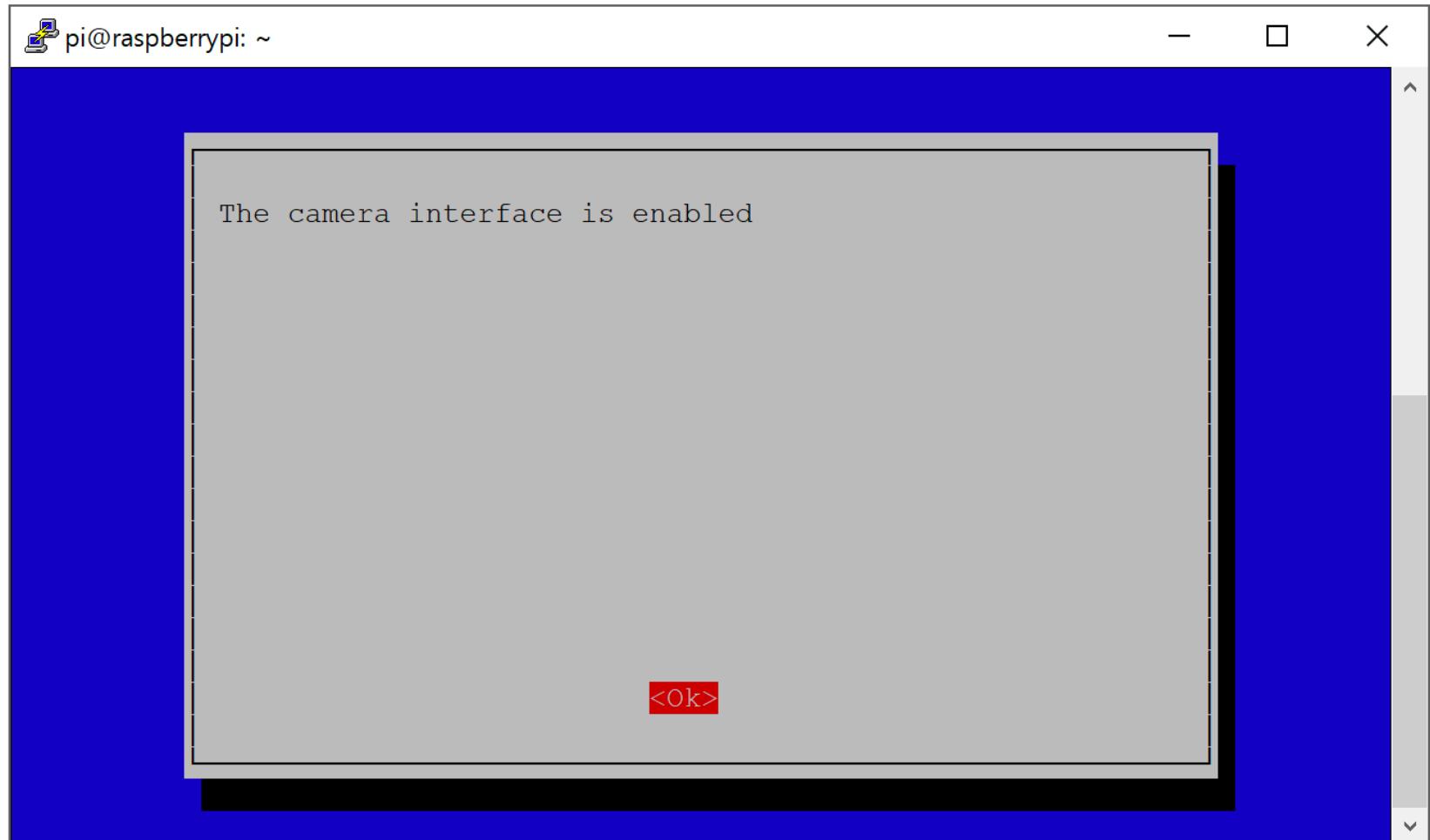
RPi Camera 활성화

■ sudo raspi-config



RPi Camera 활성화

■ sudo raspi-config



RPi Camera 활성화

■ sudo raspi-config



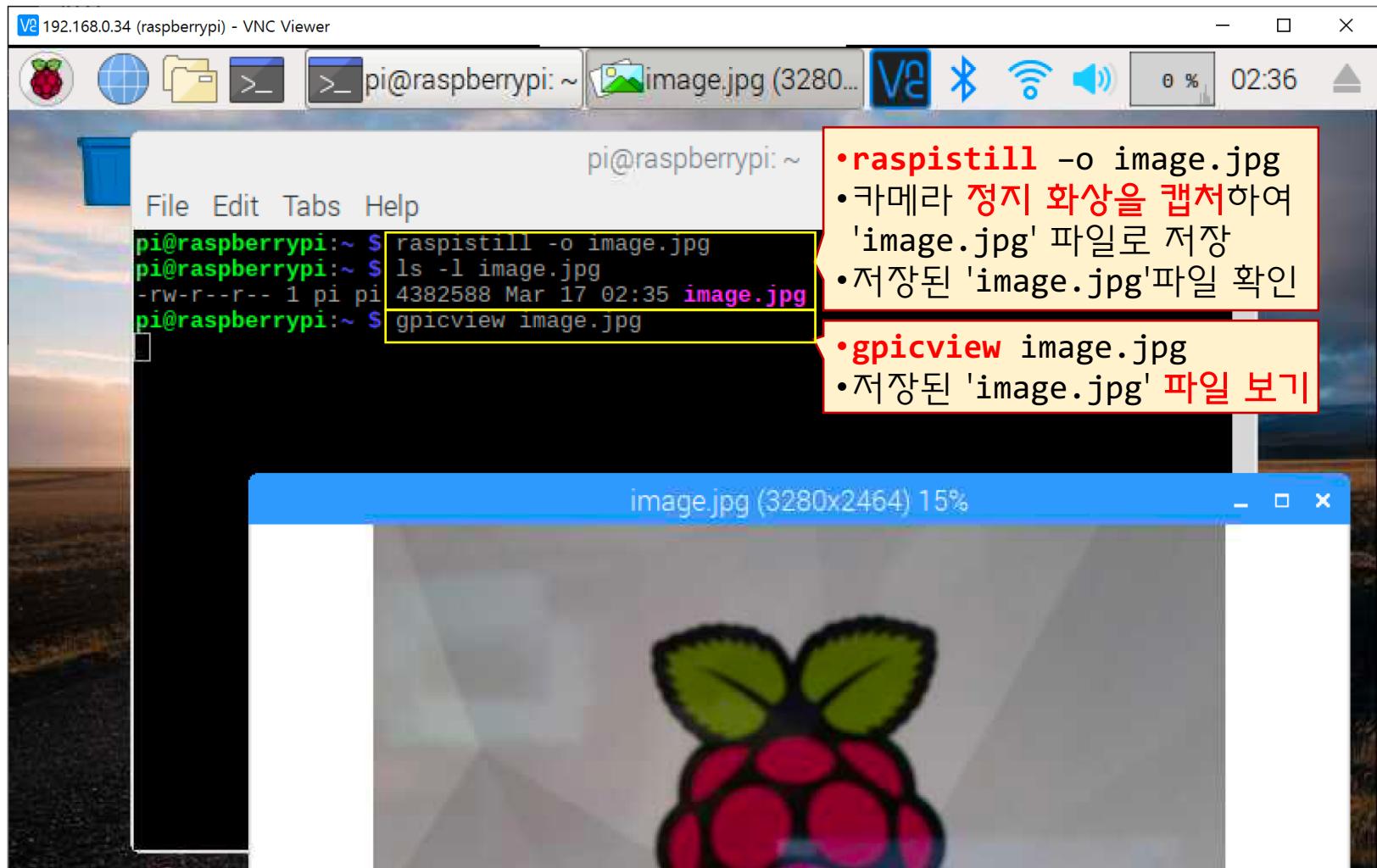
RPi Camera 활용

캡처와 녹화

RPi Camera 활용

PuTTY에서 이미지 캡처가 가능하나 이미지 보기가 불가능 하기 때문에 **VNC 뷰어**로 진행

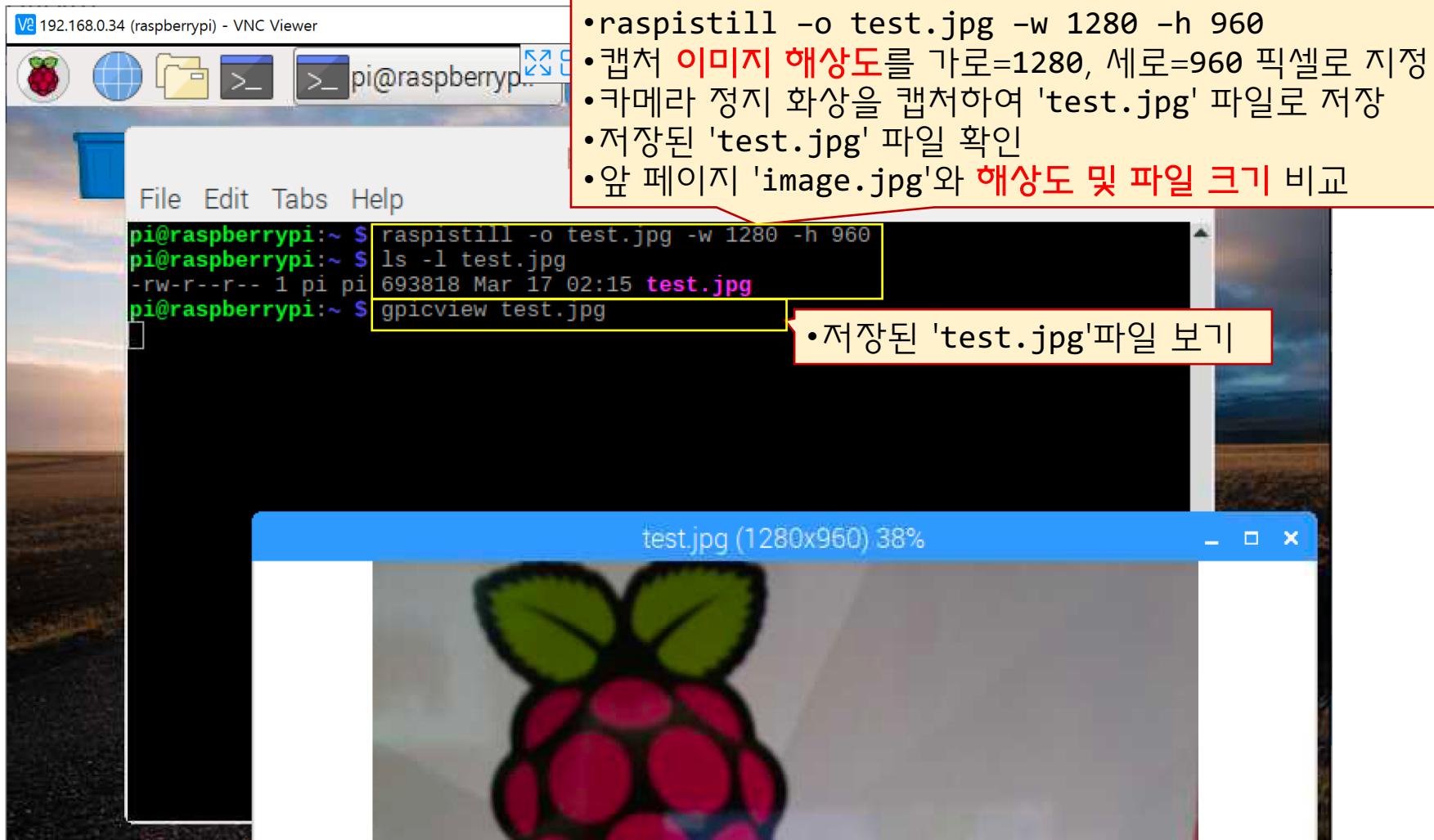
■ 정지 화상 캡처와 캡처 이미지 보기



RPi Camera 활용

PuTTY에서 이미지 캡처가 가능하나 이미지 보기가 불가능 하기 때문에 **VNC 뷰어**로 진행

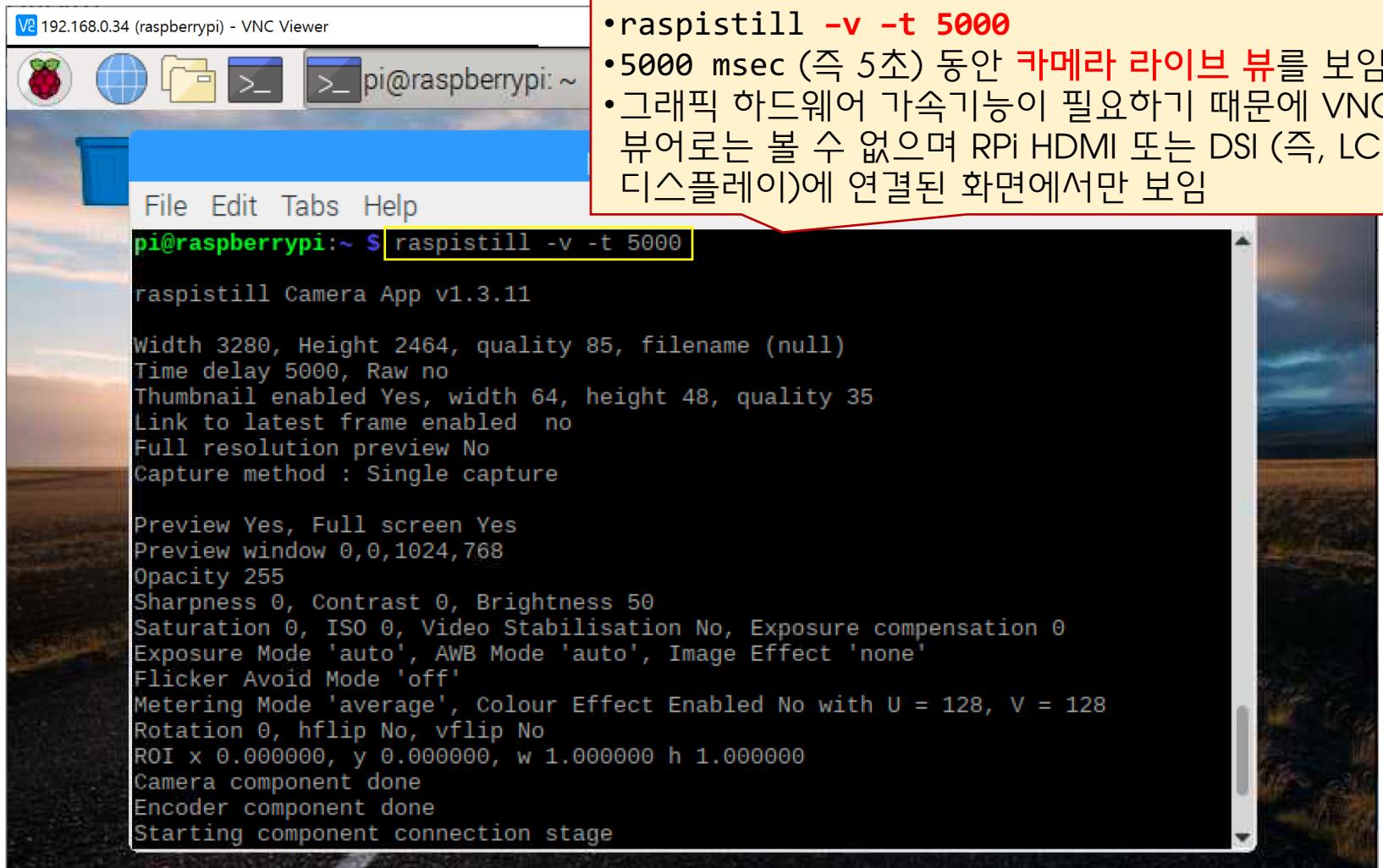
■ 정지 화상 캡처와 캡처 이미지 보기



RPi Camera 활용

카메라 라이브 뷰는 하드웨어 가속때문에
HDMI 또는 DSI 연결 디스플레이가 필요

■ 카메라 라이브 뷰



```

VNC 192.168.0.34 (raspberrypi) - VNC Viewer
File Edit Tabs Help
pi@raspberrypi:~ $ raspistill -v -t 5000

raspistill Camera App v1.3.11

Width 3280, Height 2464, quality 85, filename (null)
Time delay 5000, Raw no
Thumbnail enabled Yes, width 64, height 48, quality 35
Link to latest frame enabled no
Full resolution preview No
Capture method : Single capture

Preview Yes, Full screen Yes
Preview window 0,0,1024,768
Opacity 255
Sharpness 0, Contrast 0, Brightness 50
Saturation 0, ISO 0, Video Stabilisation No, Exposure compensation 0
Exposure Mode 'auto', AWB Mode 'auto', Image Effect 'none'
Flicker Avoid Mode 'off'
Metering Mode 'average', Colour Effect Enabled No with U = 128, V = 128
Rotation 0, hflip No, vflip No
ROI x 0.000000, y 0.000000, w 1.000000 h 1.000000
Camera component done
Encoder component done
Starting component connection stage

```

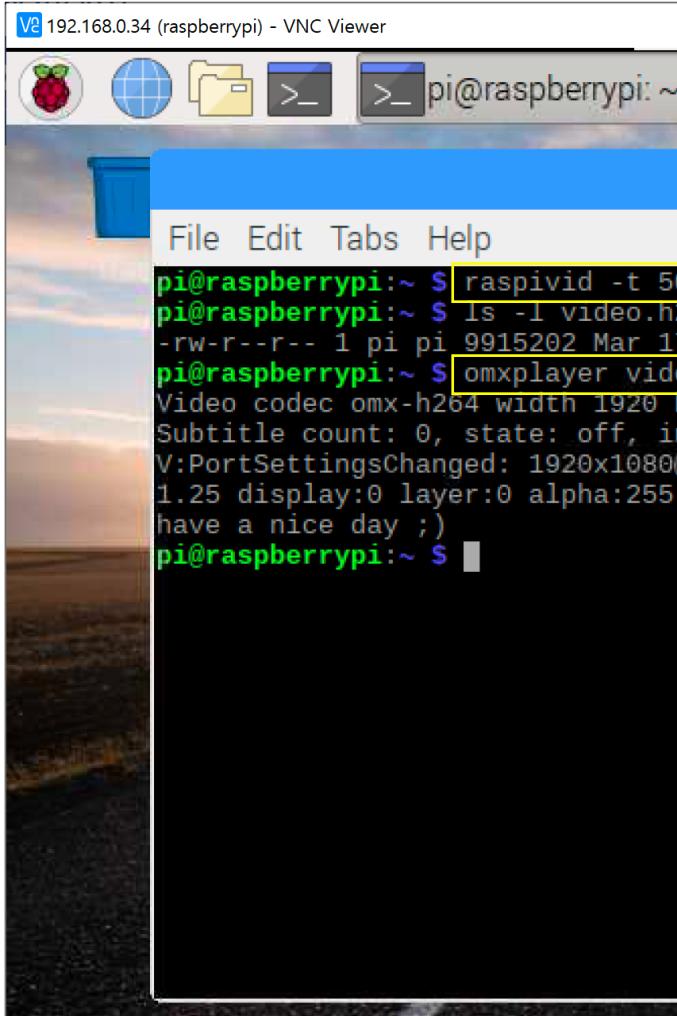
• raspistill -v -t 5000

- 5000 msec (즉 5초) 동안 **카메라 라이브 뷰**를 보임
- 그래픽 하드웨어 가속기능이 필요하기 때문에 VNC 뷰어로는 볼 수 없으며 RPi HDMI 또는 DSI (즉, LCD 디스플레이)에 연결된 화면에서만 보임

RPi Camera 활용 카메라 라이브 뷰 보기

동영상 플레이어는 하드웨어 가속 때문에 HDMI 또는 DSI 연결 디스플레이가 필요

■ 동영상 캡처와 보기



- **raspivid** -t 5000 -o video.h264
- 5000 msec (즉 5초) 동안 카메라 영상을 녹화하여 'video.h264' 파일로 저장
- 동영상 저장 비트율 (bitrate)을 '-b' 옵션을 이용해 조정이 가능
- **raspivid** -t 60000 **-b 8000000** -o video.h264 는 60초 동안 **8Mbit/s** (1MByte/s)로 동영상 녹화

- **omxplayer**를 이용하여 'video.h264' 파일 재생
- 그래픽 하드웨어 가속기능이 필요하기 때문에 VNC 뷰어로는 볼 수 없으며 RPi HDMI 또는 DSI (즉, LCD 디스플레이)에 연결된 화면에서만 보임
- 일반적인 미디어 플레이어에서 재생하기 위한 형식 (예, mp4)으로 변환하기 위해서는 **추가 패키지 활용 필요**

- **raspistill**, **raspivid** 명령을 코드 안에서 호출을 하는 방식으로 **카메라 활용 프로그램** 개발 가능
- 예, 침입 감지 캡처 및 녹화 서비스
- **raspistill**, **raspivid** 명령은 RPi 전용이므로 확장성과 보다 많은 API 사용을 위해서는 **Video4Linux 활용¹** 필요
- OpenCV² 등 이미지/동영상 분석 오픈 소스 라이브러리를 활용하여 얼굴/자세 인식 프로그램 개발 가능

¹ 익스플로링 라즈베리 파이, p660

² <https://opencv.org/>

Summary

- PIR sensor with RPi
- Touch sensor with RPi
- Ambient sound sensor with RPi (digital output)
- Ultrasonic sensor with RPi
- Temperature/humidity sensor with RPi
- Timing functions in wPi library
- VNC in RPi
- Camera serial interface in RPi

Thank you

Questions?

Contact: eclass.sch.ac.kr
(순천향대학교 학습플랫폼 LMS)