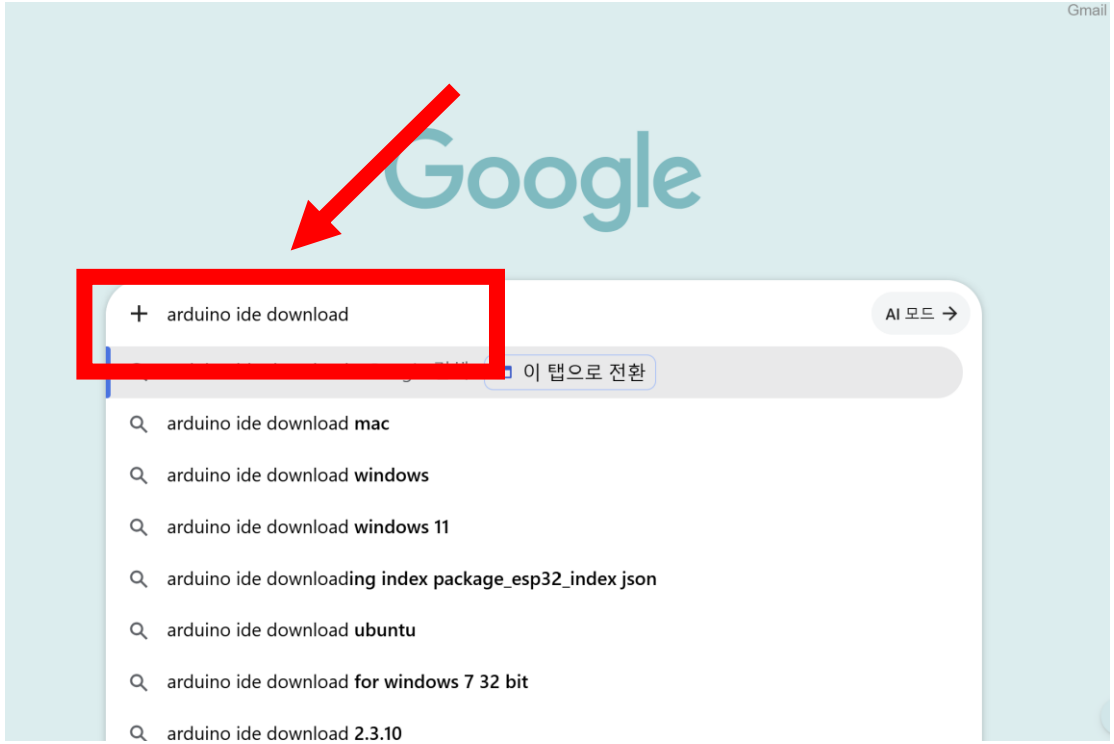
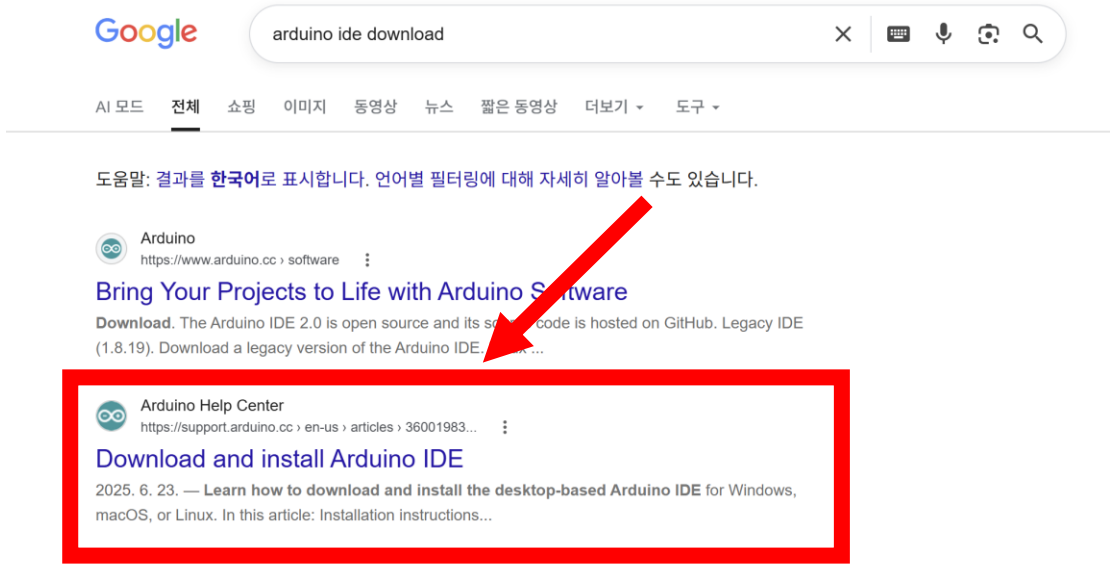
# 0. Setup ESP32 Start in your desktop

## Download Arduino IDE



# 0. Setup ESP32

## Download Arduino IDE

### Download and install Arduino IDE

Learn how to download and install the desktop-based Arduino IDE for Windows, macOS, or Linux.

In this article:

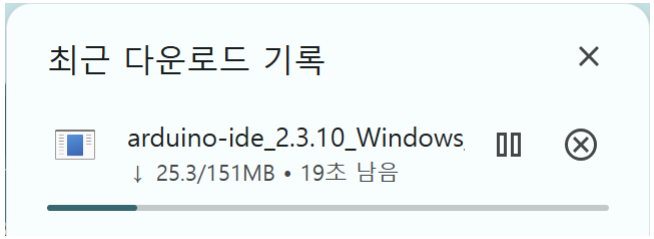
- Installation instructions
- Additional download options
- If you need help

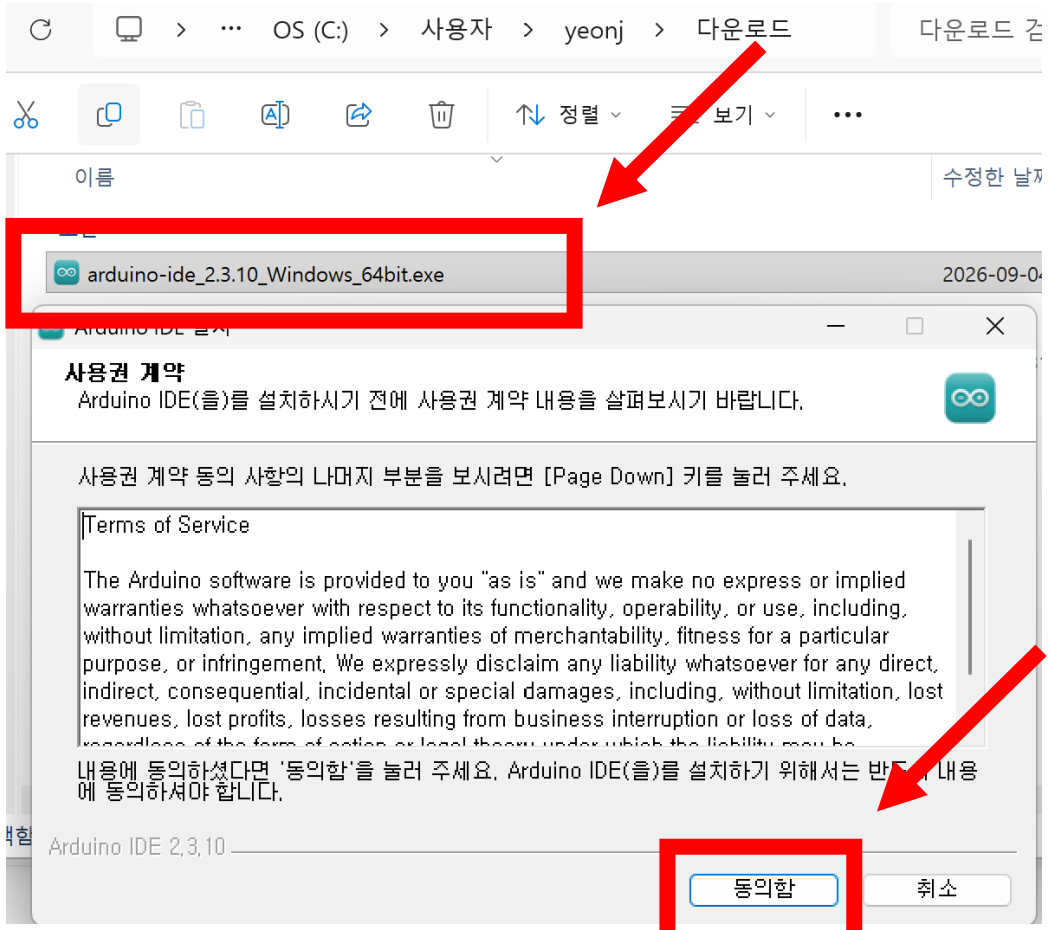
### Installation instructions

**Note**

If you're using a Chromebook, see Use Arduino on Chromebook.

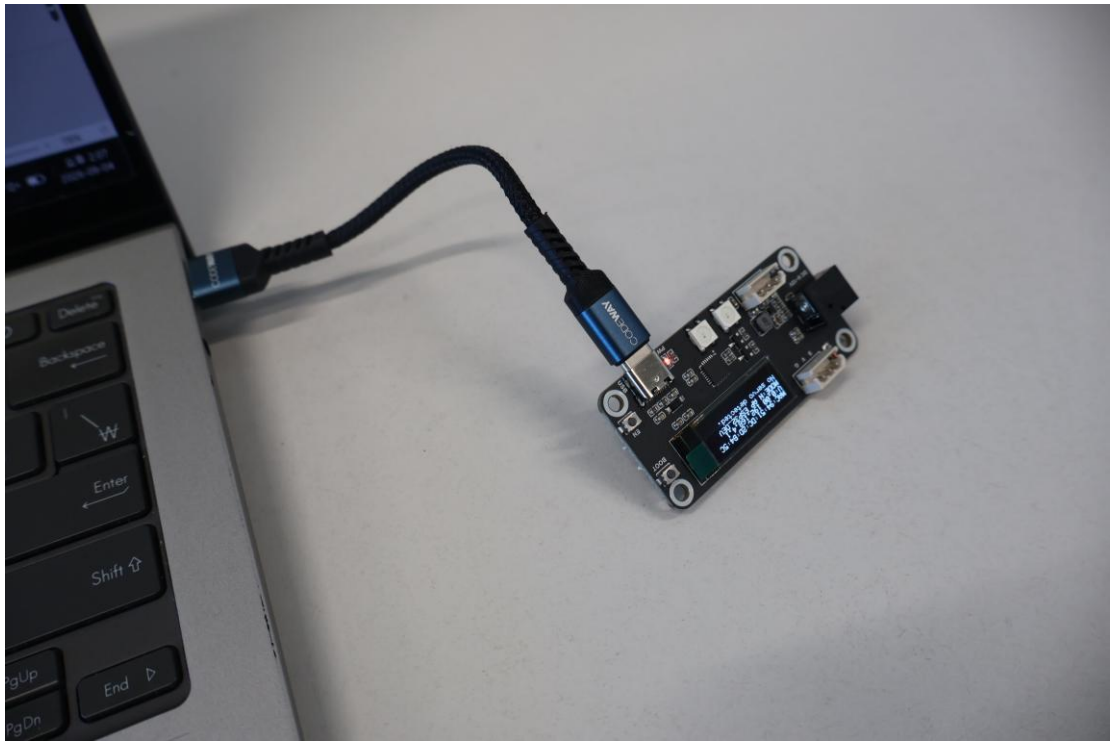
1. Download the latest release (The download will start after you click this link. Check your browser's download manager or the Downloads folder on your computer to find the downloaded file)
2. Double-click the executable (.exe) file.



# 0. Setup ESP32

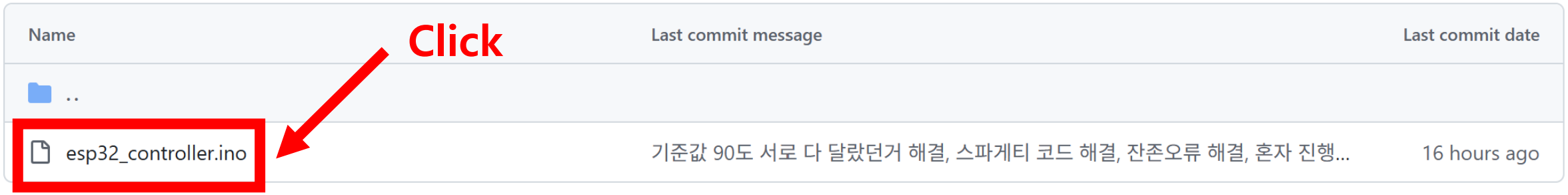
**Before start setting, make the connection between ESP32 to computer**




**And Download Firmware in next pages**

# 0. Setup ESP32

**Go to https://github.com/SeungOhh/oneclickai_drive/tree/main/robodog/esp32_controller**



# 0. Setup ESP32

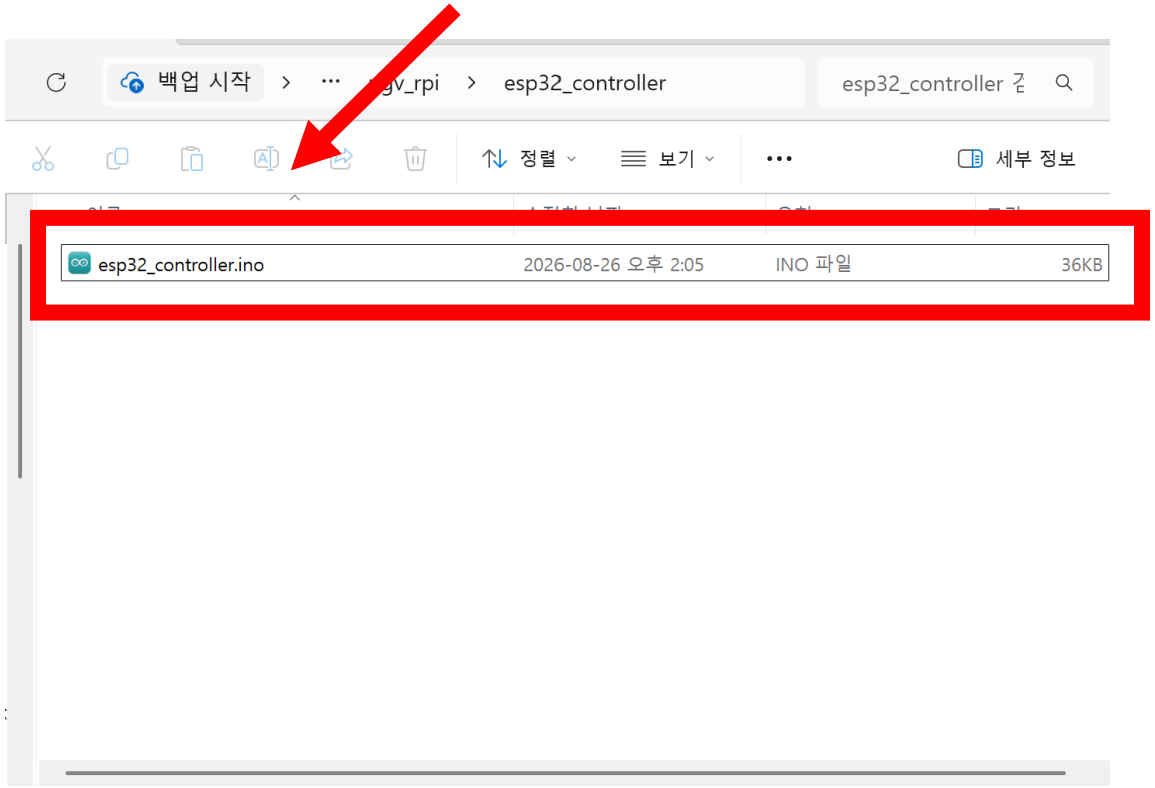


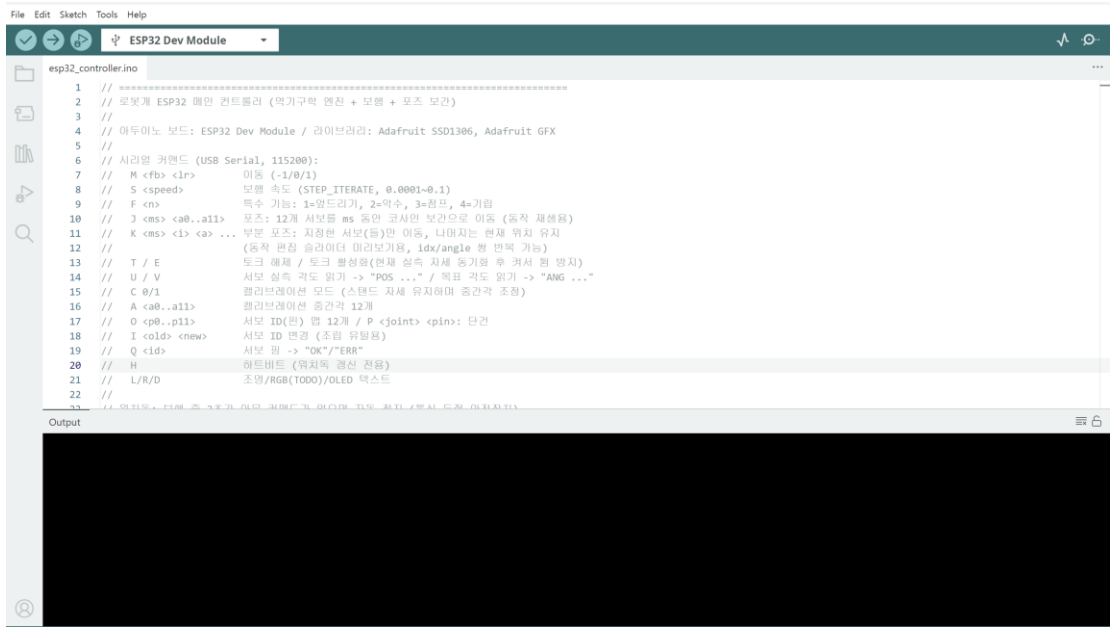

Click

```
1   // ================================================================================
2   // 로봇개 ESP32 메인 컨트롤러 (역기구학 엔진 + 보행 + 포즈 보간)
3   //
4   // 아두이노 보드: ESP32 Dev Module / 라이브러리: Adafruit SSD1306, Adafruit GFX
5   //
6   // 시리얼 커맨드 (USB Serial, 115200):
7   //   M <fb> <lr>          이동 (-1/0/1)
8   //   S <speed>            보행 속도 (STEP_ITERATE, 0.0001~0.1)
9   //   F <n>                특수 기능: 1=엎드리기, 2=악수, 3=점프, 4=기립
10  //   J <ms> <a0..a11> [mode]
11  //                        포즈: 12개 서보를 ms 동안 보간 이동 (동작 재생용)
12  //                        mode 0=코사인(기본) 1=선형 2=ease-in 3=ease-out
13  //   K <ms> <i> <a> ...   부분 포즈: 지정한 서보(들)만 이동, 나머지는 현재 위치 유지
14  //                        (동작 편집 슬라이더 미리보기용, idx/angle 쌍 반복 가능)
15  //   T / E                토크 해제 / 토크 활성화(현재 실측 자세 동기화 후 켜서 튐 방지)
16  //   U / V                서보 실측 각도 읽기 -> "POS ..." (응답 없는 서보는 "?")
17  //                        / 목표 각도 읽기 -> "ANG ..."
18  //   C 0/1                캘리브레이션 모드 (스탠드 자세 유지하며 중간각 조정)
19  //   A <a0..a11>          캘리브레이션 중간각 12개
20  //   O <p0..p11>          서보 ID(핀) 맵 12개 / P <joint> <pin>: 단건
```

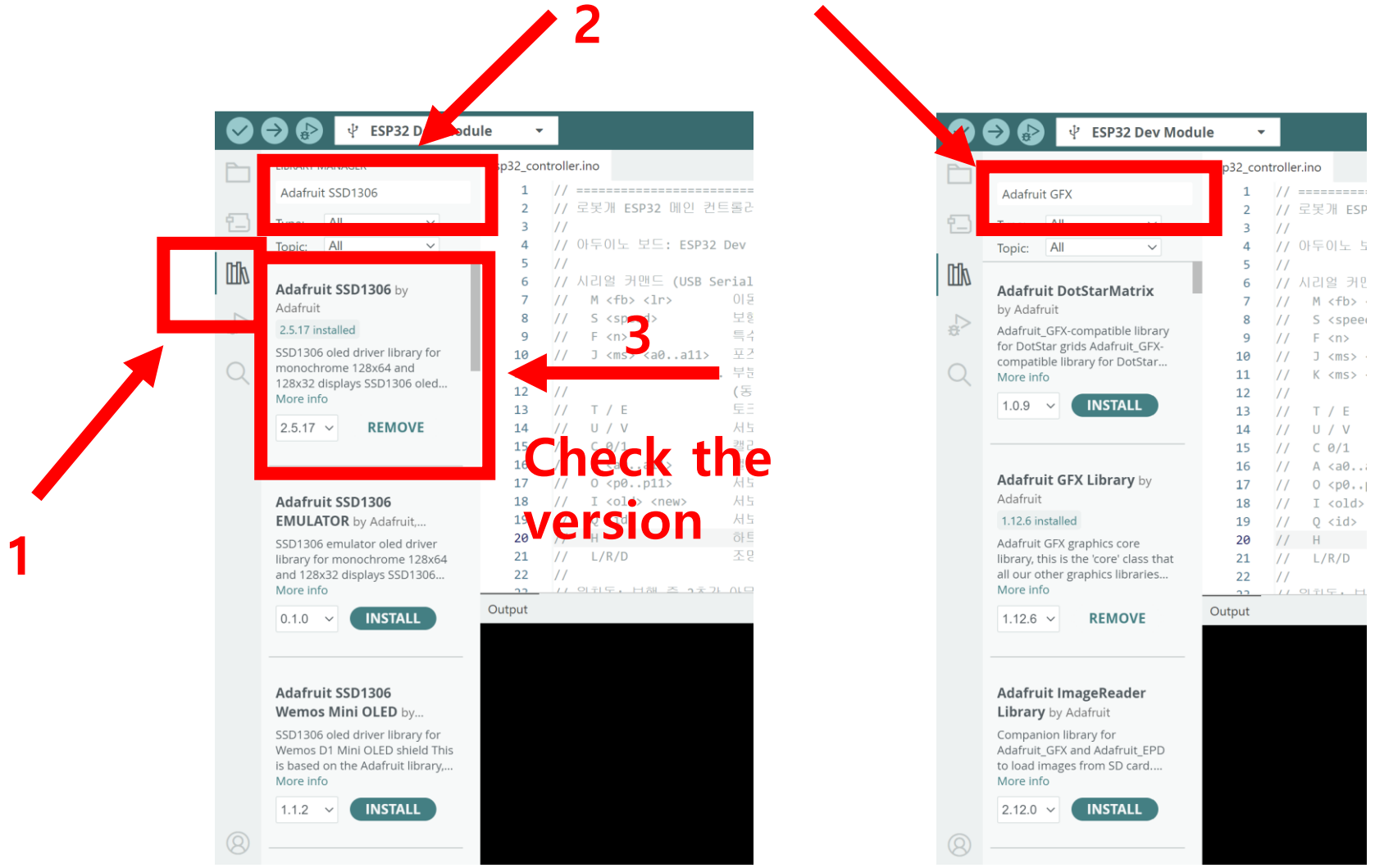
# 0. Setup ESP32

## In Download Folder

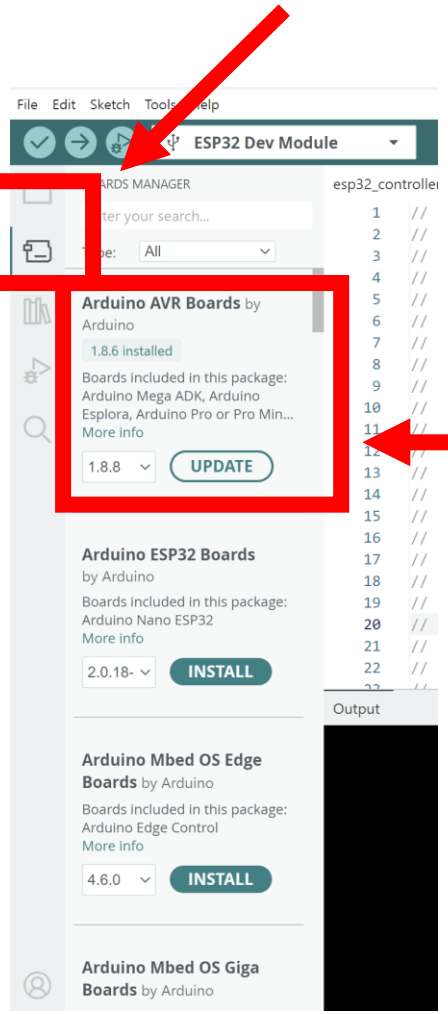


# 0. Setup ESP32



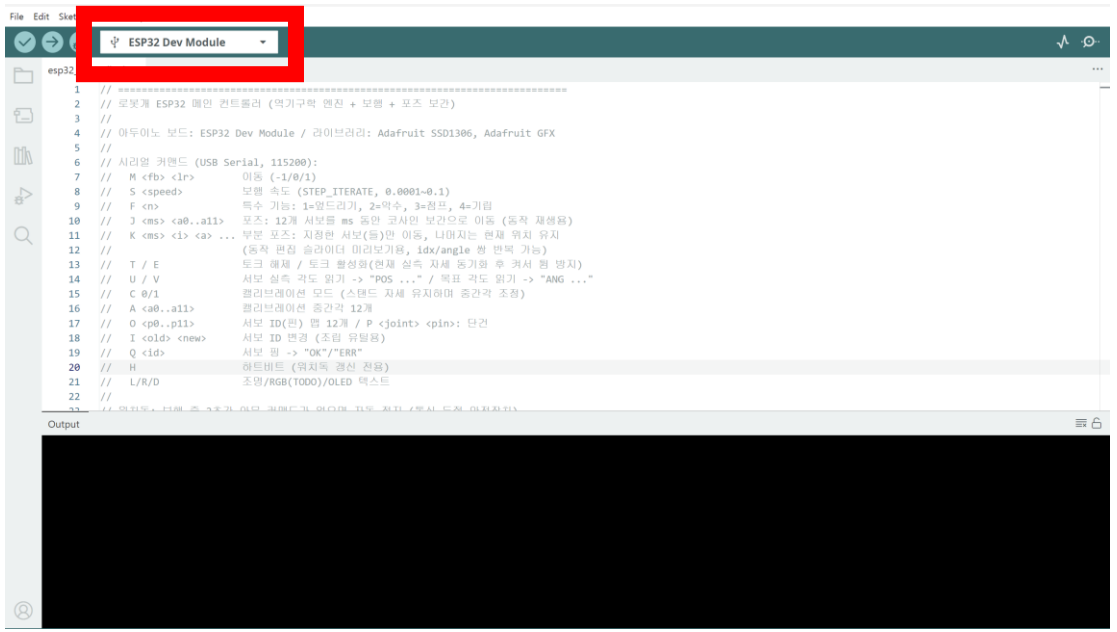
# 0. Setup ESP32

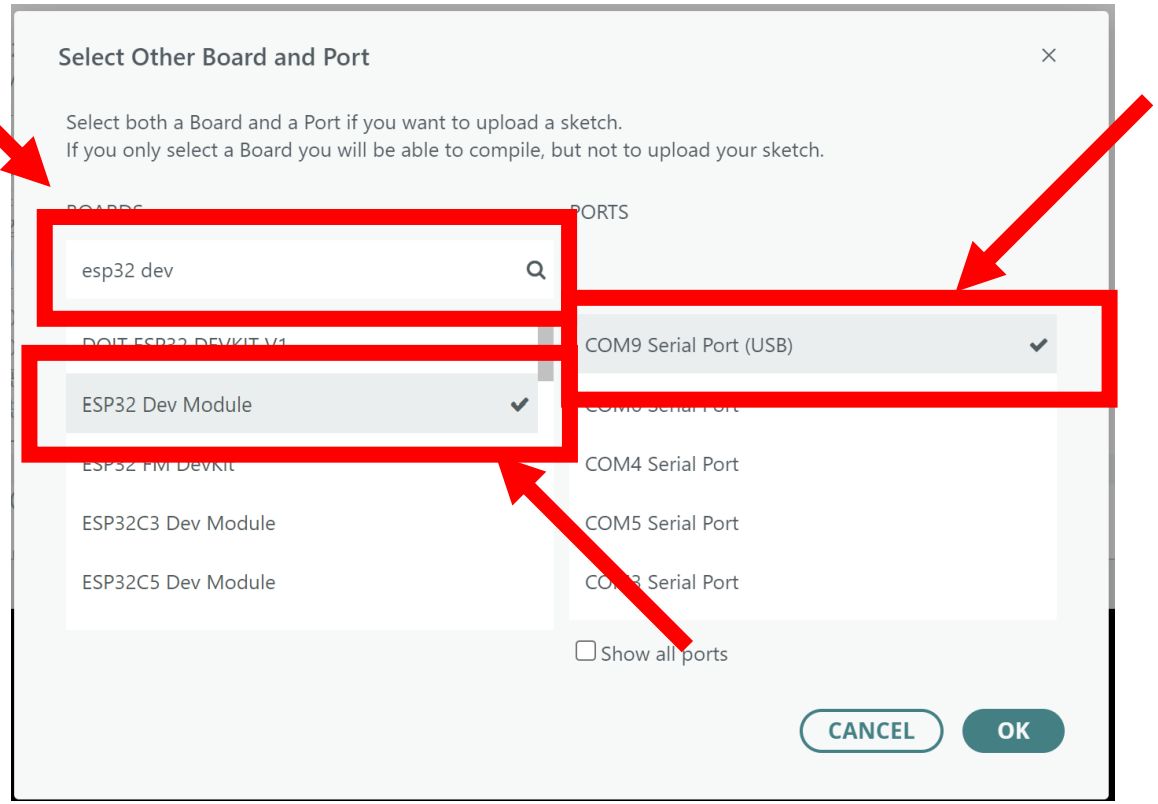



If you can see this, it means done

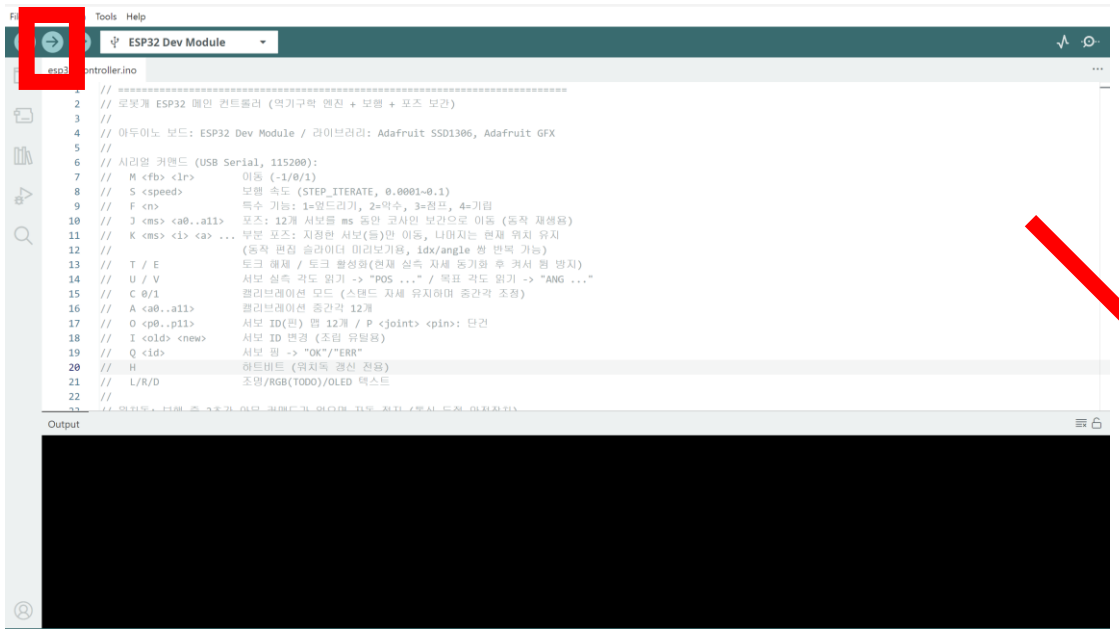
# 0. Setup ESP32

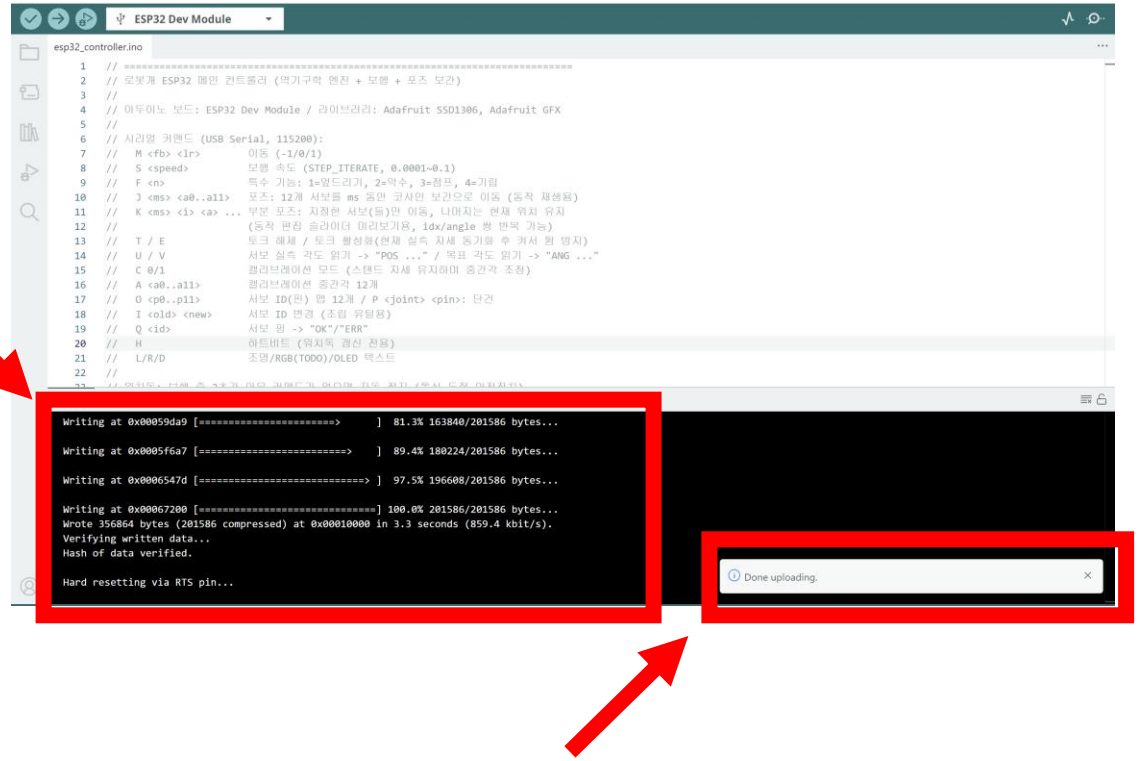
## Select the port where the USB is connected



# 0. Setup ESP32

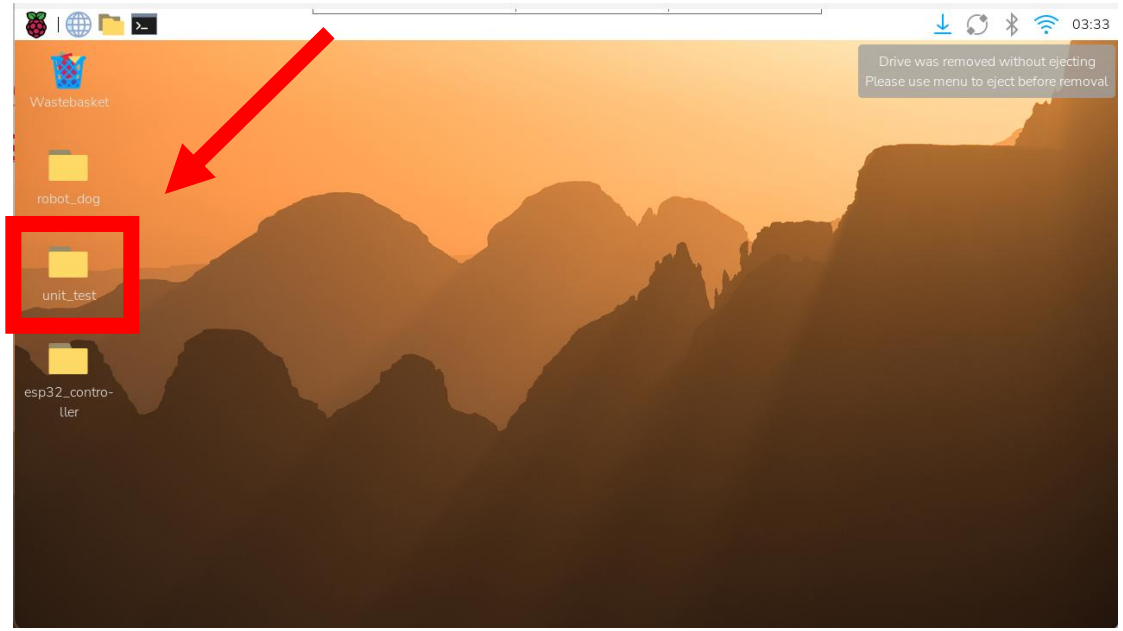


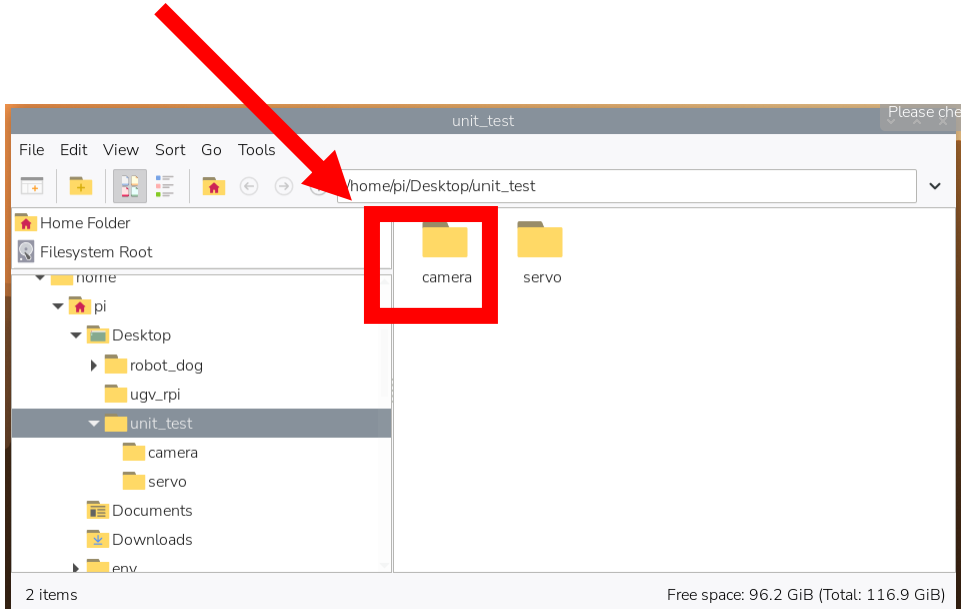



# 1. Camera test

## unit_test/camera

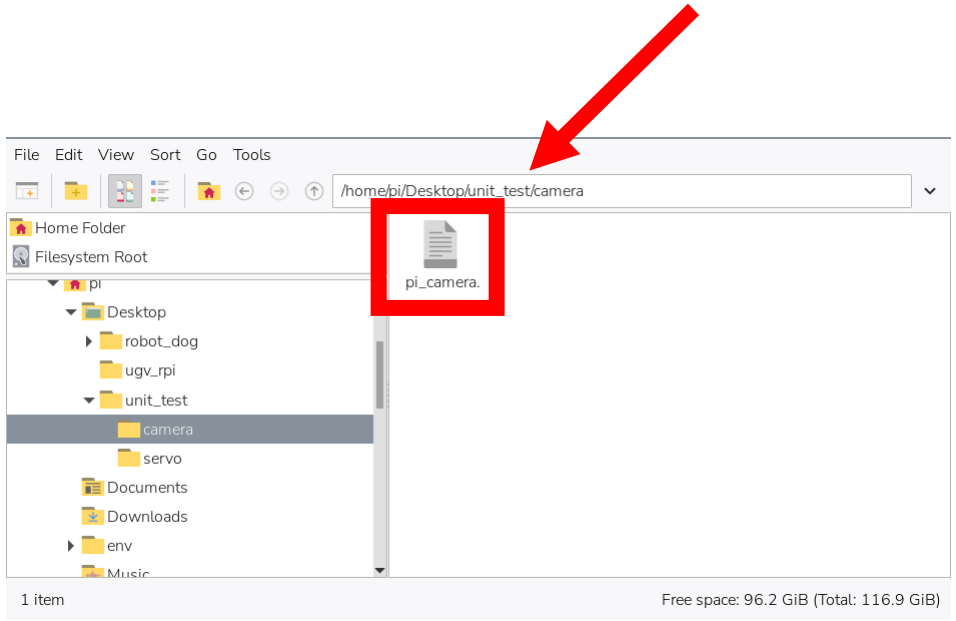
## 2 X Leg type B



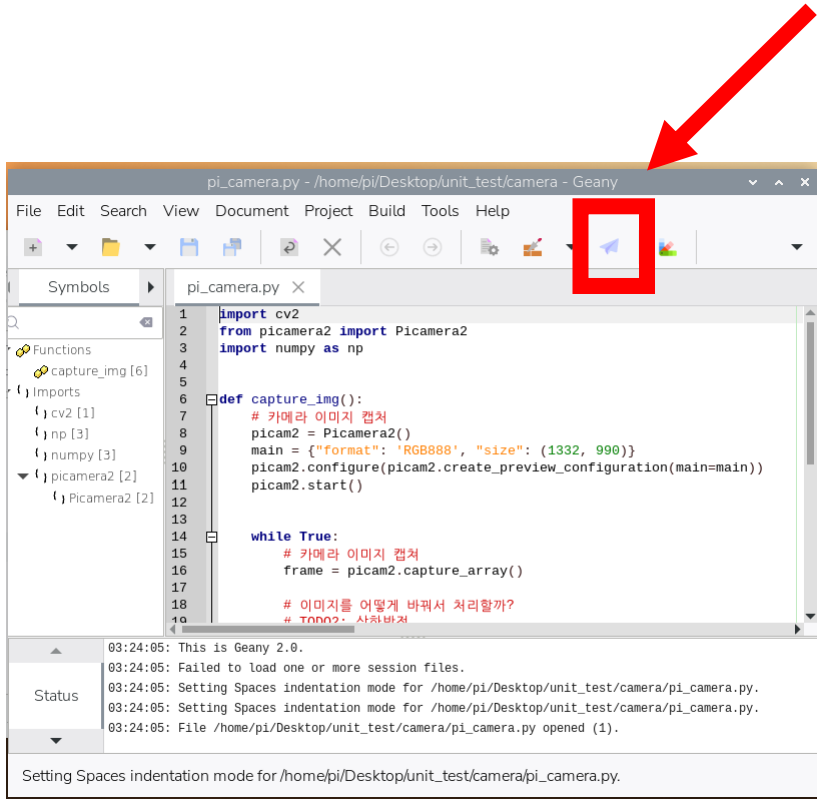



# 1. Camera test

## unit_test/camera

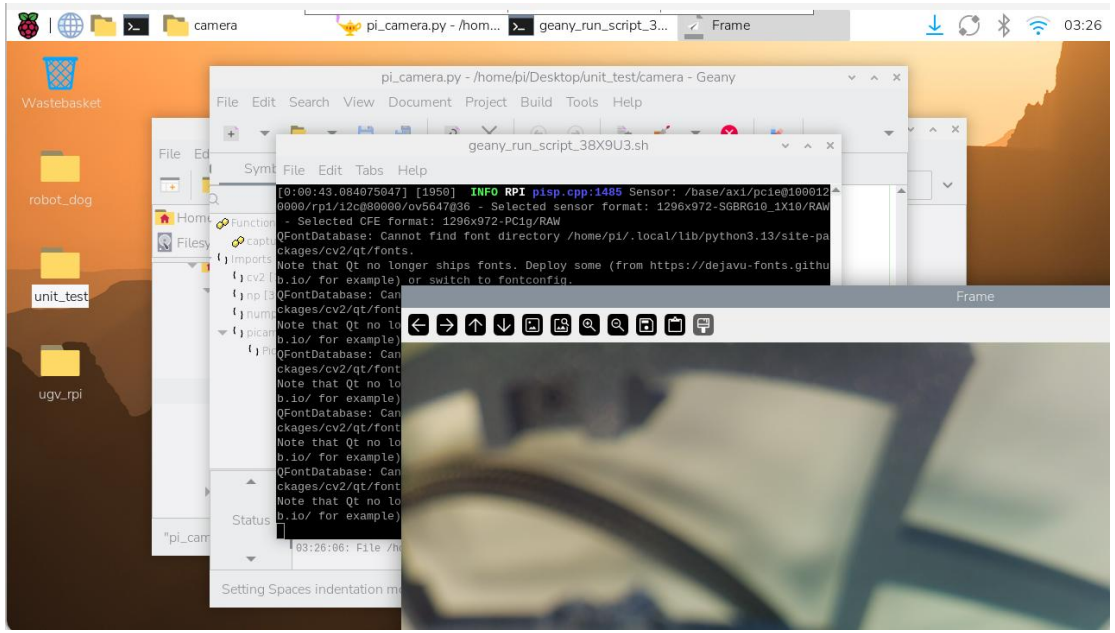



## 2 X Leg type B



# 1. Camera test

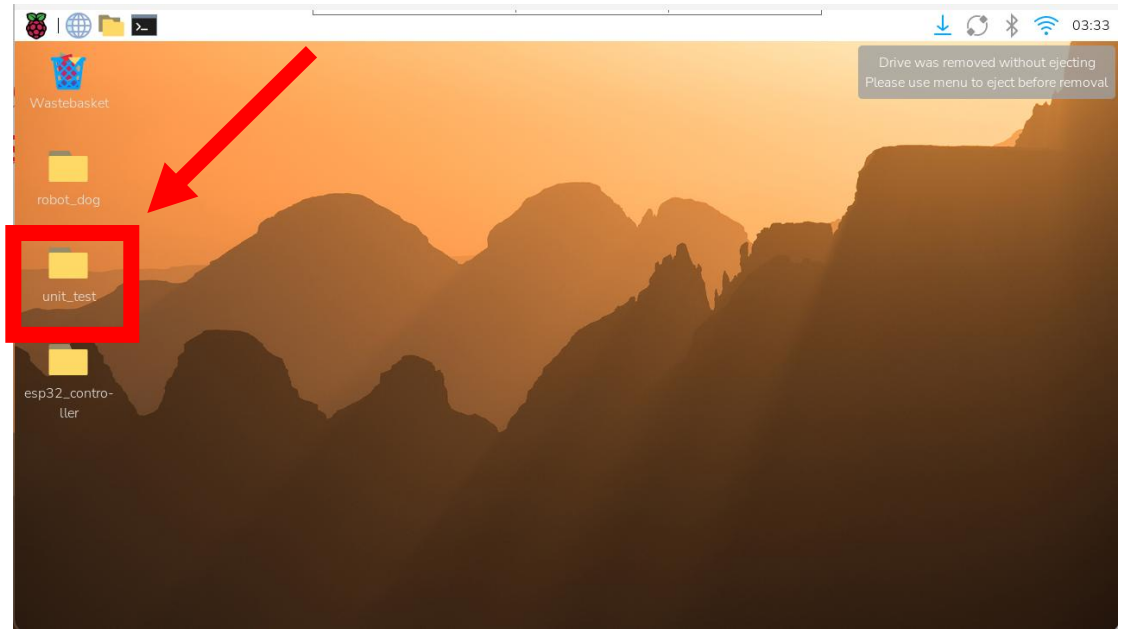
## unit_test/camera

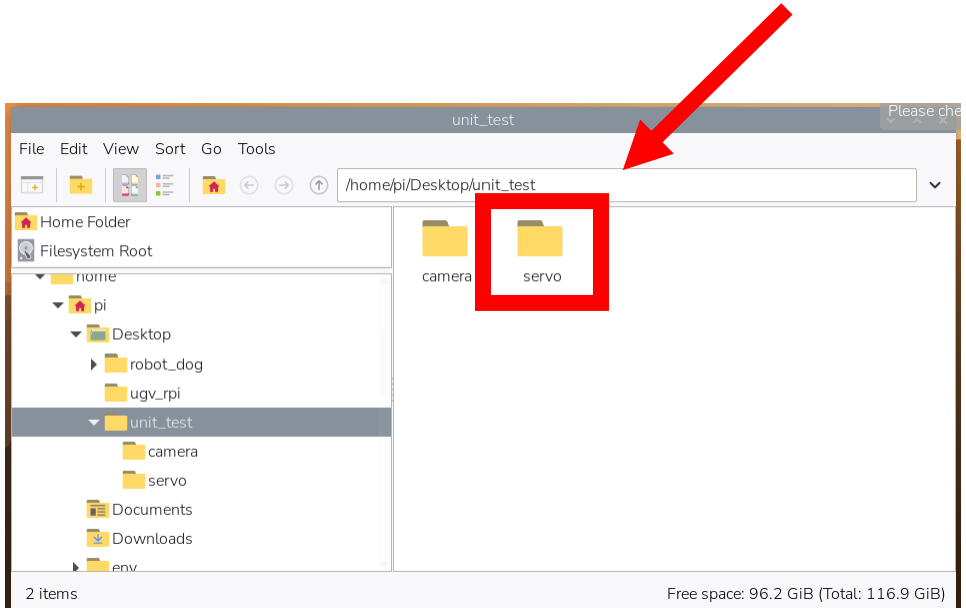



**If you can see the camera screen clearly when you run it, it means success.**
**If it's out of focus, please rotate the lens to adjust it.**

# 2. Servo test

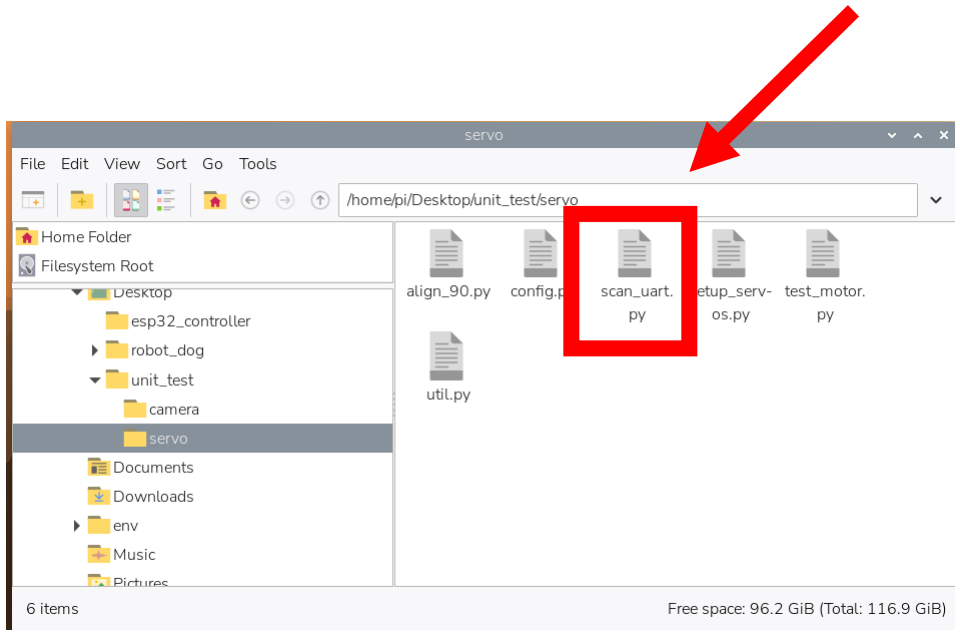
**We need to disassemble and rebuild it for servo calibration because it wasn't done during the last assembly.**

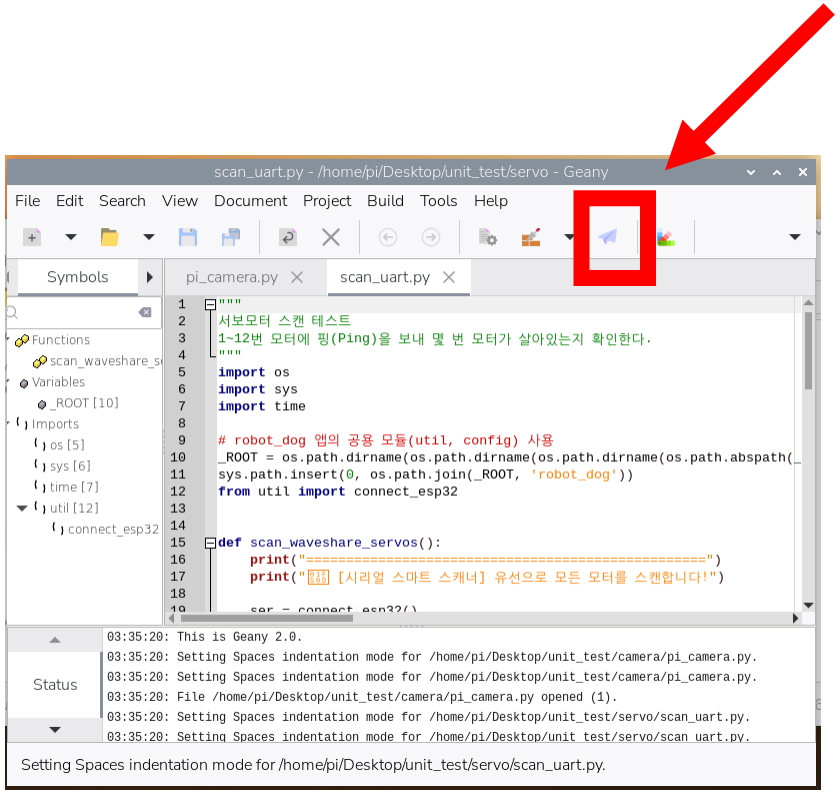


# 2. Servo test

**First, Run Scan_uart.py, you can see that the ESP32 is successfully connected.**

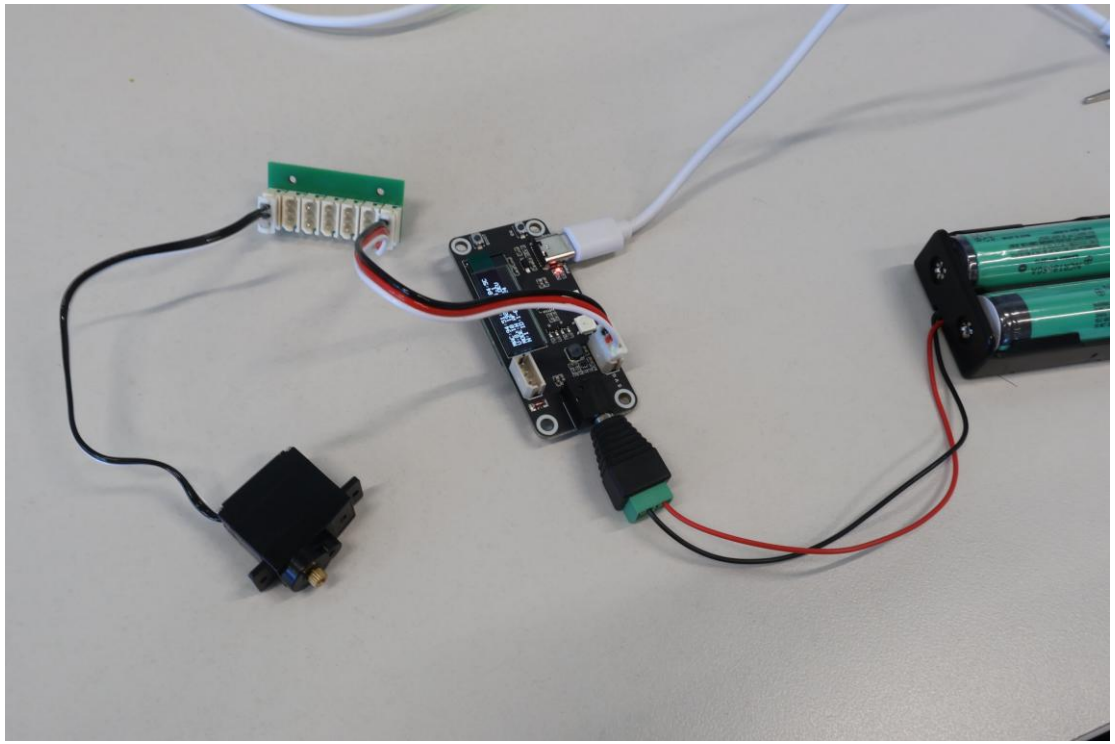



**However, none of the servo motors are showing up. Why is this happening?**

# 2. Servo test

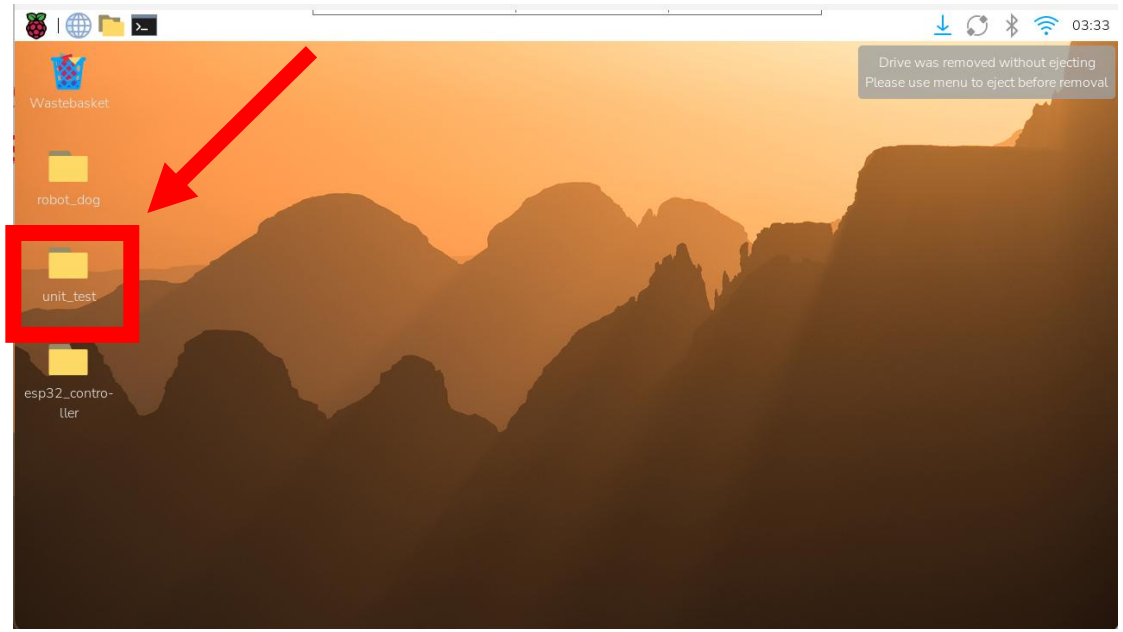
**This is because the servos haven't been assigned IDs yet**
**Let's assign them IDs in numerical order**

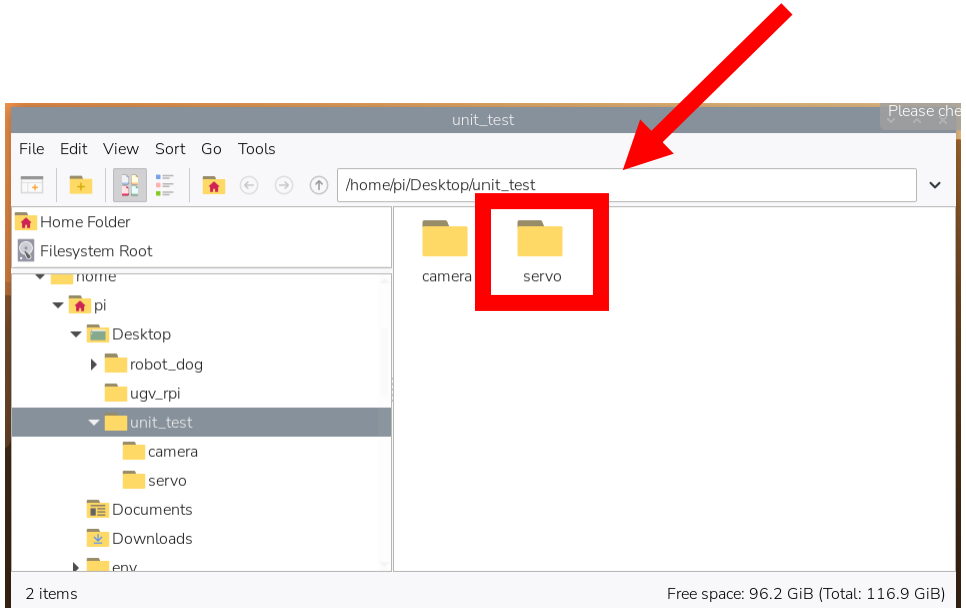


**Connect the ESP to the Raspberry Pi.**

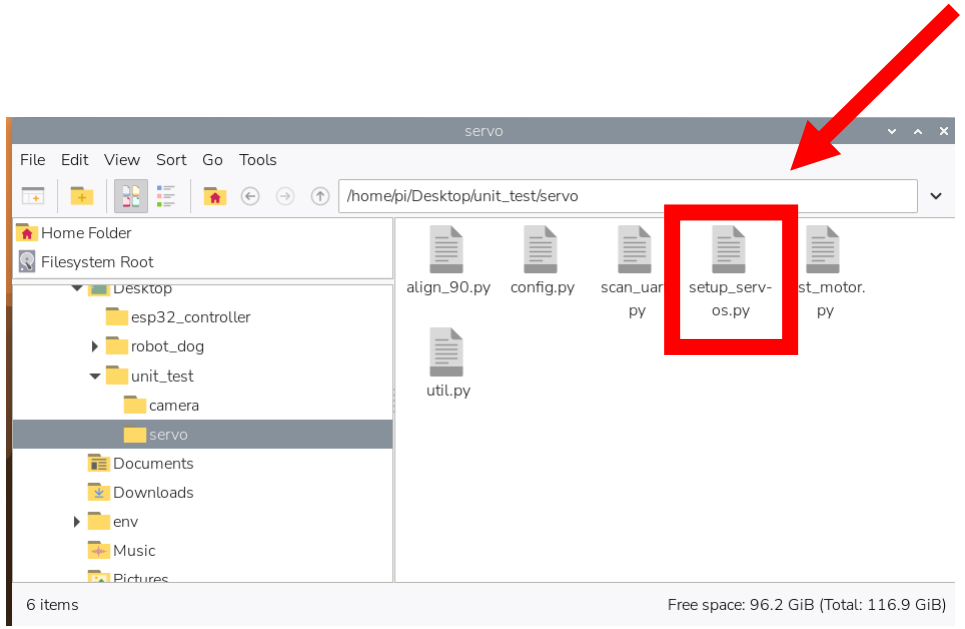
**Only one servo motor should be connected when setting the IDs.**

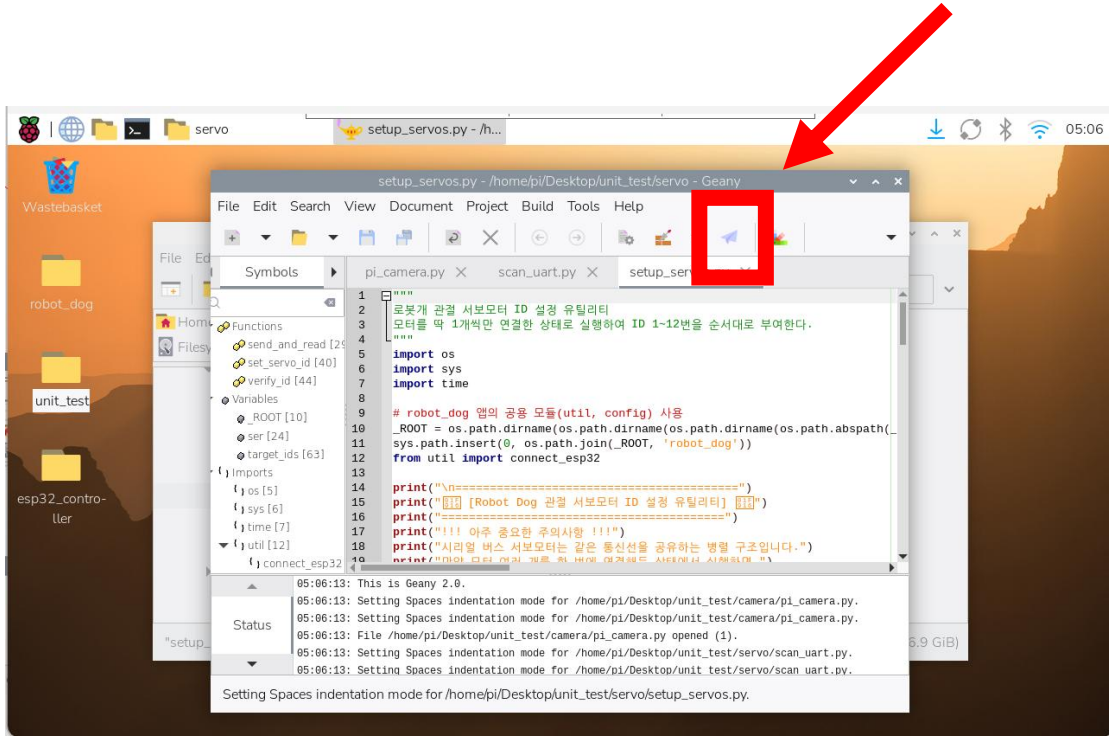
# 2. Servo test


03:33
Drive was removed without ejecting
Please use menu to eject before removal
Wastebasket
robot_dog
unit_test
esp32_contro-
ller


unit_test
File Edit View Sort Go Tools
/home/pi/Desktop/unit_test
Home Folder
Filesystem Root
home
pi
Desktop
robot_dog
ugv_rpi
unit_test
camera
servo
Documents
Downloads
env
camera
servo
2 items
Free space: 96.2 GiB (Total: 116.9 GiB)

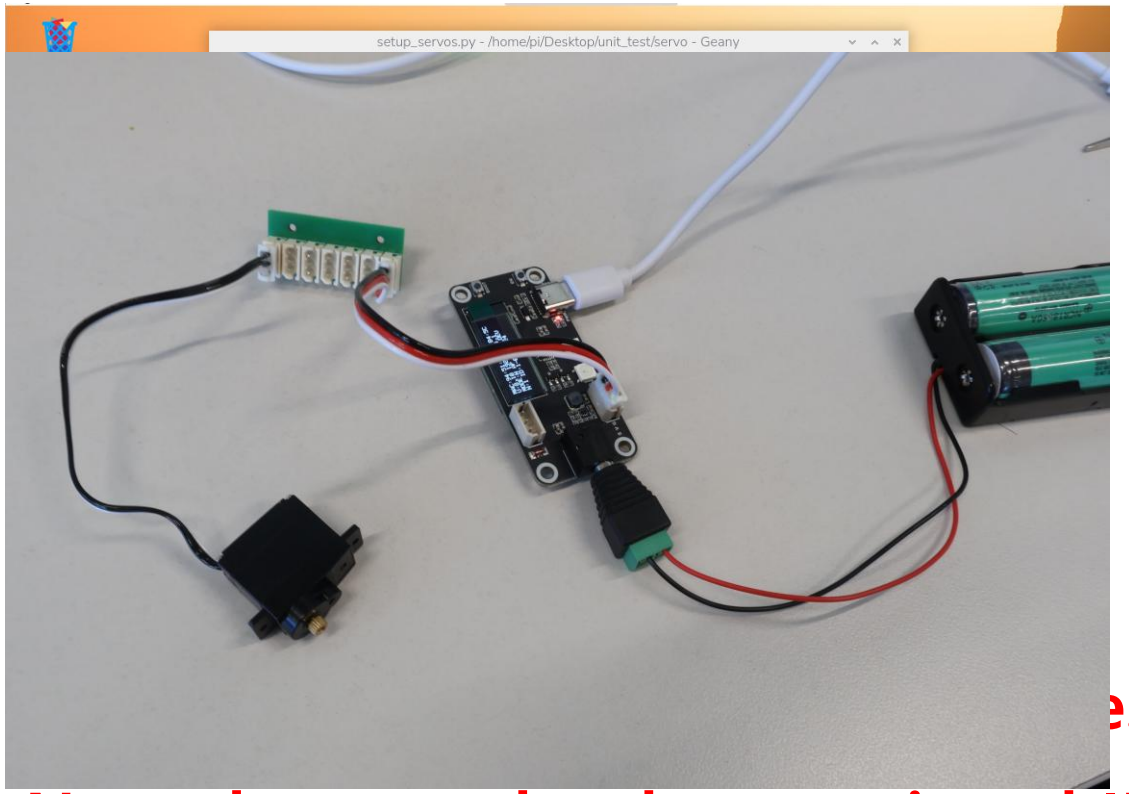
# 2. Servo test


servo
File Edit View Sort Go Tools
/home/pi/Desktop/unit_test/servo
Home Folder
Filesystem Root
Desktop
esp32_controller
robot_dog
unit_test
camera
servo
Documents
Downloads
env
Music
Pictures
align_90.py
config.py
scan_uar
py
setup_serv-
os.py
st_motor.
py
util.py
6 items
Free space: 96.2 GiB (Total: 116.9 GiB)


servo
setup_servos.py - /h...
05:06
Wastebasket
robot_dog
unit_test
esp32_contro-
ller
setup_servos.py - /home/pi/Desktop/unit_test/servo - Geany
File Edit Search View Document Project Build Tools Help
Symbols
pi_camera.py
scan_uart.py
setup_serv
Functions
send_and_read [29
set_servo_id [40]
verify_id [44]
Variables
_ROOT [10]
ser [24]
target_ids [63]
Imports
os [5]
sys [6]
time [7]
util [12]
connect_esp32
"""
로봇개 관절 서보모터 ID 설정 유틸리티
모터를 딱 1개씩만 연결한 상태로 실행하여 ID 1~12번을 순서대로 부여한다.
"""
import os
import sys
import time
# robot_dog 앱의 공용 모듈(util, config) 사용
_ROOT = os.path.dirname(os.path.dirname(os.path.dirname(os.path.abspath(_
sys.path.insert(0, os.path.join(_ROOT, 'robot_dog'))
from util import connect_esp32
print("\n========================================")
print("[] [Robot Dog 관절 서보모터 ID 설정 유틸리티] []")
print("========================================")
print("!!! 아주 중요한 주의사항 !!!")
print("시리얼 버스 서보모터는 같은 통신선을 공유하는 병렬 구조입니다.")
Status
05:06:13: This is Geany 2.0.
05:06:13: Setting Spaces indentation mode for /home/pi/Desktop/unit_test/camera/pi_camera.py.
05:06:13: Setting Spaces indentation mode for /home/pi/Desktop/unit_test/camera/pi_camera.py.
05:06:13: File /home/pi/Desktop/unit_test/camera/pi_camera.py opened (1).
05:06:13: Setting Spaces indentation mode for /home/pi/Desktop/unit_test/servo/scan_uart.py.
05:06:13: Setting Spaces indentation mode for /home/pi/Desktop/unit_test/servo/scan_uart.py.
Setting Spaces indentation mode for /home/pi/Desktop/unit_test/servo/setup_servos.py.

# 2. Servo test

**Run the code with only one servo motor connected.**

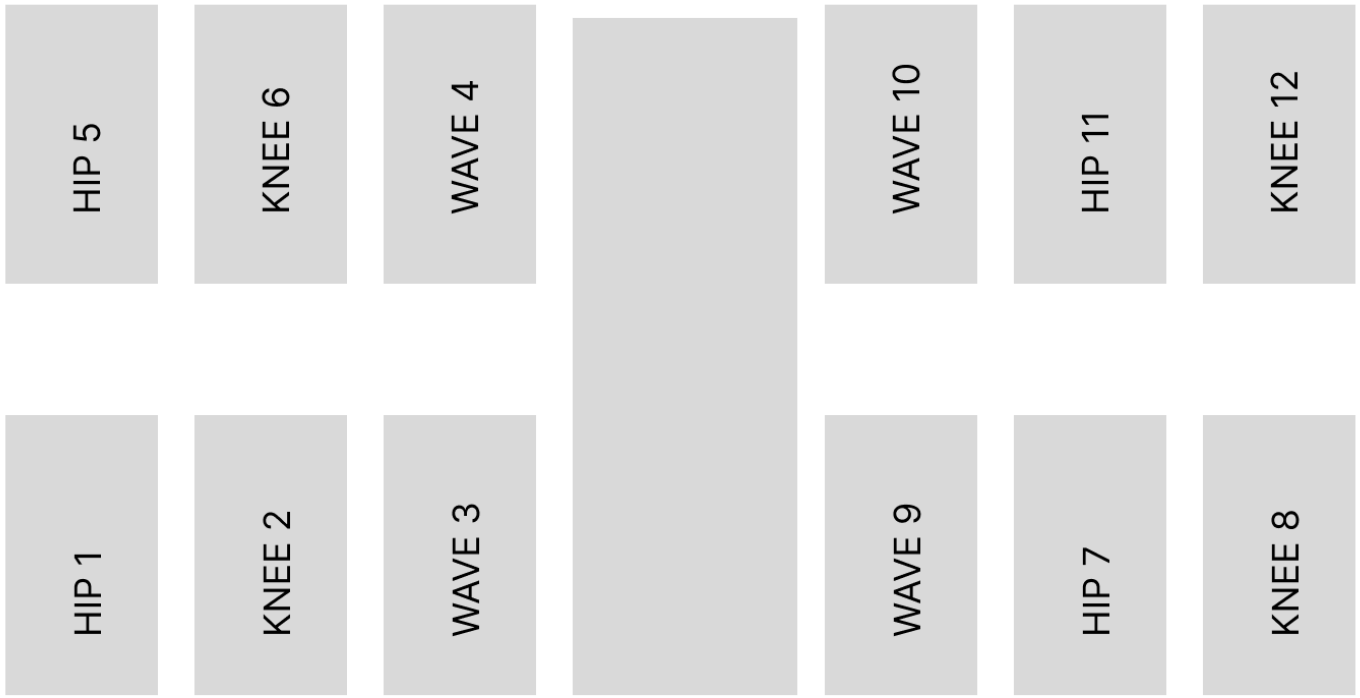



**e.**

**Now, the servo has been assigned ID 1.**

# 2. Servo test

**Next, connect another motor and press Enter again to save it as ID 2.**
**Repeat this process up to ID 12.**

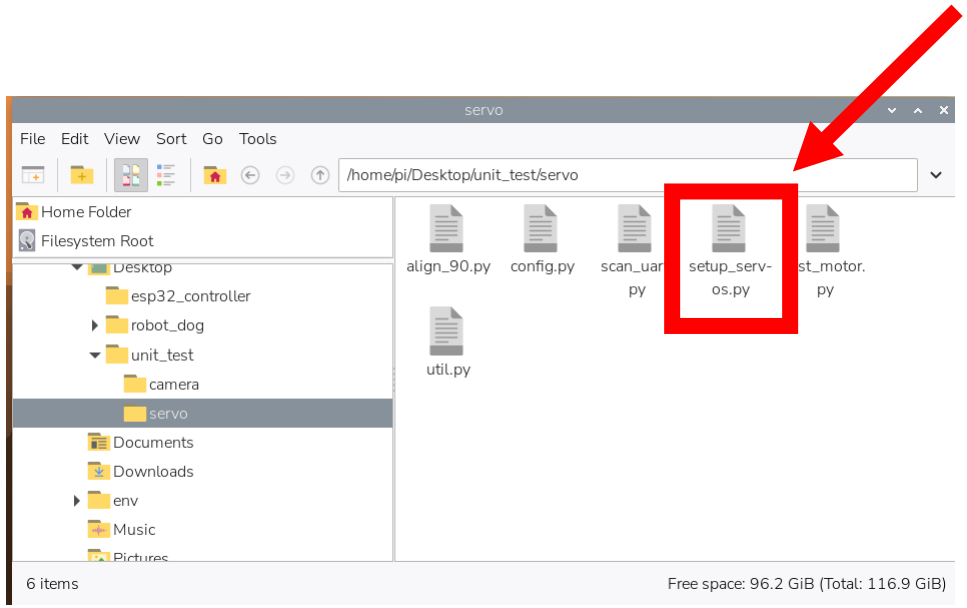


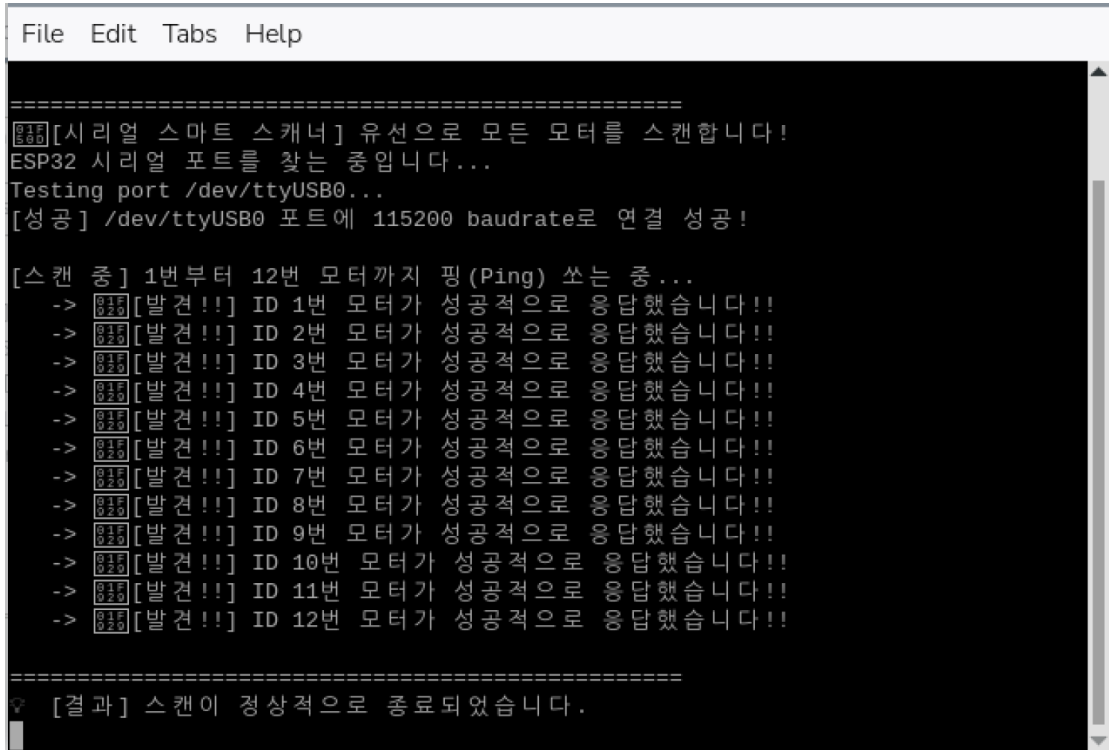

**Camera**

**Please assign the IDs to the servos matching the positions and numbers shown in this diagram. Otherwise, it will not function properly.**

# 2. Servo test

**Once the setup is complete, run the scan_uart code again.**

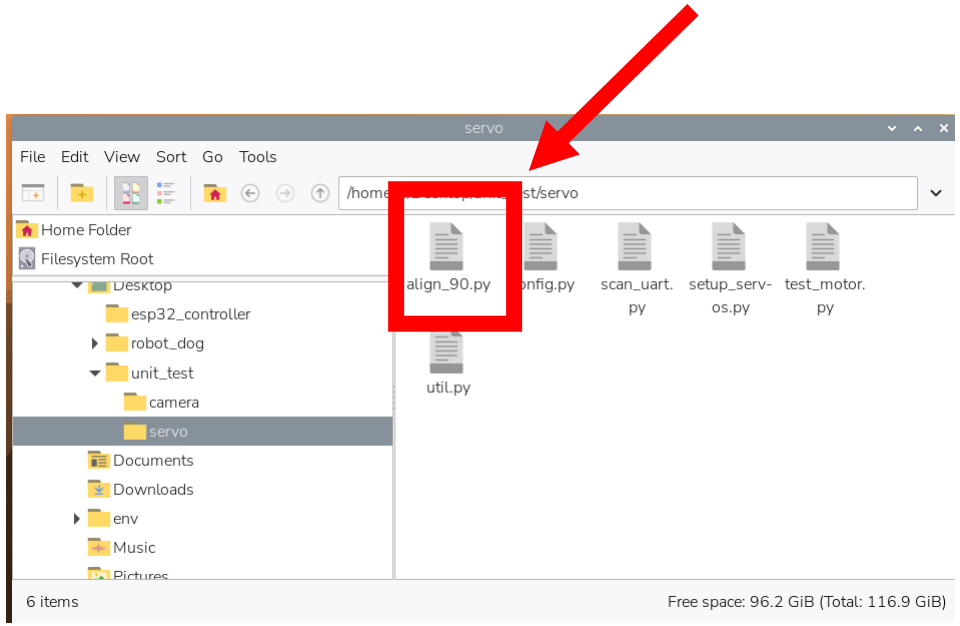



**If you see this log, it is successful.**

# 2. Servo test

**Now, set all the servos to 90 degrees.**
**Please connected all servo once like this**




**Now all the servomotors are set to 90 degrees.**
**Please keep track of the number assigned to each servo.**