ESP32 WROOM U32 Wifi+Bluetooth Configuration

Producer : Yıldırım Elektronik

* Özgün Çevik
* Hızır Arda Can (Stajer 07-08.2025)

Date: 18.07.2025

## **End-to-End Guide: Building & Flashing ESP-AT Firmware**

This document walks you through setting up your development environment, configuring the AT firmware, building it, and flashing it onto an ESP32-WROOM-32U module, from zero to final.

### **1. Prerequisites**

* **Ubuntu 22.04** (or similar Linux)

From: <https://ubuntu.com/download>

* **Virtual Machine** or native install with at least 4 GB RAM, 20 GB free disk

From: <https://www.virtualbox.org/>

* **USB-to-UART adapter** and jumper wires

IMPORTANT Note: Use UART1 for connection from ESP32 WROOM 32U to STM32 Board

For ESP: RX = 16 , TX= 17 .

For STM32, it doesn’t matter, but our code configures with respect to UART6, so RX = PC7, TX PC6.

* ESP32-WROOM-32U development board

### **🔧 How to Install Core Packages and Set Serial Access on Ubuntu (Step-by-Step)**

#### **📌 Step 1: Open the Terminal**

There are two easy ways to open the Terminal.

##### **🔸 Method 1: Using the Mouse (Right Click)**

1. Move your mouse to an **empty area** on your desktop (no icons).
2. **Right-click** with your mouse.
3. From the menu that appears, click **“Open Terminal”**.

##### **🔸 Method 2: From the Application Menu**

1. Click the **"Show Applications"** button at the bottom-left (looks like a grid of 9 dots).
2. In the search bar, type **“Terminal”**.
3. Click the **Terminal** icon to open it

##### **🔸 Method 3: Use Shortcut**

1. CTRL+ALT+T

#### **📌 Step 2: Update the Package List *(Optional)***

Now we will update your system to make sure you get the latest package versions.

💡 **Note**:  
 If you **just installed Ubuntu** for the first time, this step is usually **not necessary**. But if Ubuntu has been installed for a while, it's **a good idea** to run this to refresh the list of available software.

To update the package list, type (or copy and paste) the following command into the terminal, then press **Enter**:

sudo apt update

* It will ask for your password. Type your **computer password** (you won't see anything as you type — that's normal), then press **Enter** again.

You can skip this step if your system is new, but it’s still **recommended** to do it if you're not sure.

#### **📌 Step 3: Install Core Development Packages**

Next, install the required tools for development.

Copy and paste this long command into the terminal, then press **Enter**:

sudo apt install git wget flex bison gperf python3 python3-pip python3-setuptools cmake ninja-build ccache libffi-dev libssl-dev dfu-util

* If it asks "Do you want to continue? [Y/n]", press **Y** and then **Enter**.

#### **📌 Step 4: Allow Serial Port Access (Dialout Group)**

This step allows your computer to talk to devices through USB or serial ports (like ESP32 or Arduino boards).

Copy and paste the following command into the terminal and press **Enter**:

sudo usermod -a -G dialout $USER

* OR you can write your user name instead of “$USER”

Ex: sudo usermod -a -G dialout ozgun

#### **📌 Step 5: Log Out and Log Back In**

To apply the group change:

1. Click on your **profile picture** at the top-right corner of the screen.
2. Choose **"Log Out"**.
3. After logging out, **log back in** with your username and password.

💡 **Note**: Restarting the computer also works and applies the changes.

### 

### 

### **2. Clone & Install ESP-IDF**

**Get the correct IDF version** (v5.0.1):

cd ~

git clone -b v5.0.1 --recursive https://github.com/espressif/esp-idf.git

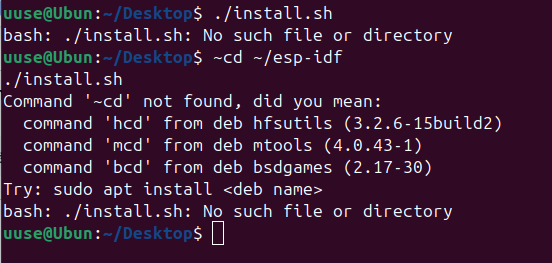
**Install tools & Python venv**:  
cd ~/esp-idf

./install.sh

**Possible Errors:**

Error 1: Wrong Directory:

bash: ./install.sh: No such file or directory



Solution 1 :

**Navigate to the esp-idf directory** Open your file explorer and find the folder named esp-idf.

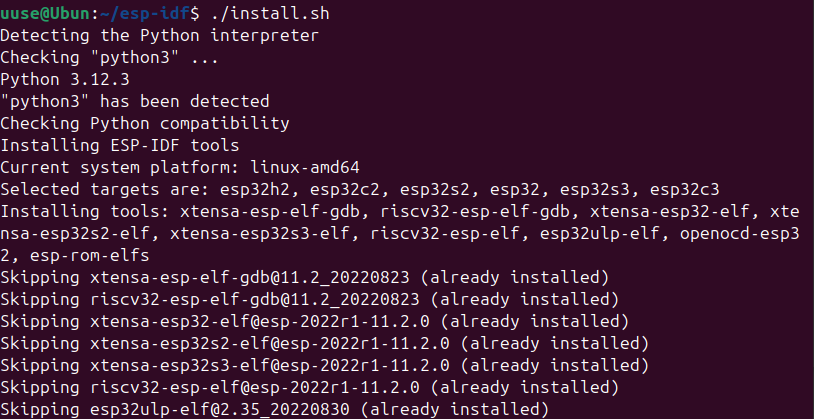
**Open Terminal in that directory**

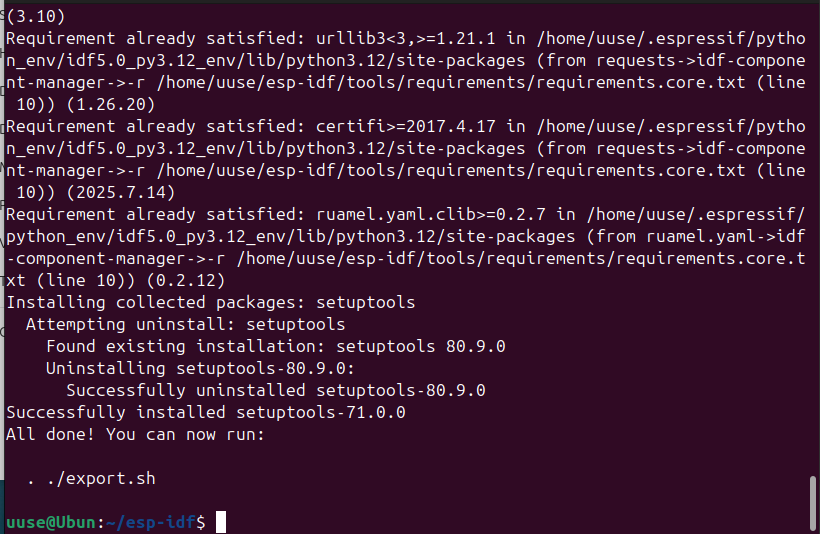
* **Left-click** on the esp-idf folder or an empty space inside it to select it.
* Then **right-click** (or use the menu) and choose **Open in Terminal**.

*This will open a terminal window with the current path set to the esp-idf folder.*

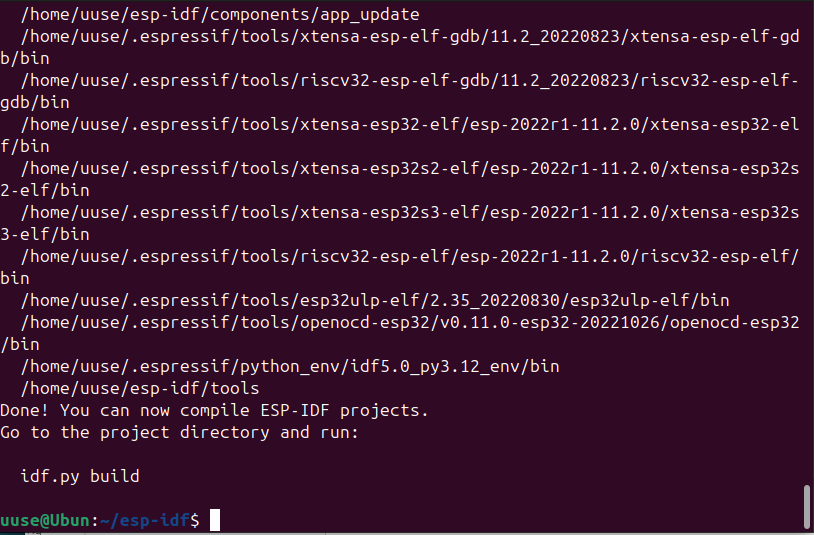
**Run the installation script** In the terminal, type the following command and press Enter:

./install.sh





**Export environment**:  
. ./export.sh



After running the screen like this.

Error 2: A fatal error occurred: No module named 'xlrd'

Solution 1 :

Run :

pip install xlrd

OR

pip3 install xlrd

According to your Python version.

Try to run again:

./install.sh

### **3. Clone & Checkout ESP-AT**

**Clone the AT firmware**: (It can take some time according to your computer and internet)   
cd ~

git clone https://github.com/espressif/esp-at.git

cd esp-at

**Checkout desired AT release** (v3.2.0.0):  
git checkout v3.2.0.0

git submodule update --init --recursive

**Verify the ESP-IDF submodule** points to v4.4 (for AT v3.2.0.0):  
cd esp-idf

git checkout v5.0.1

. ./export.sh

cd ..

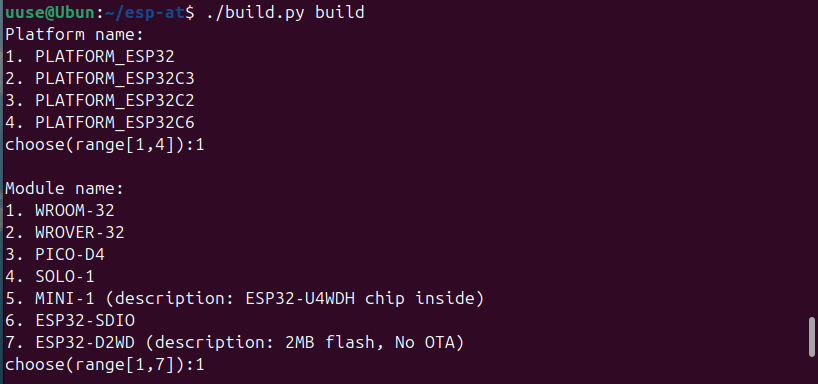
### **4. Build AT Firmware (build)**

Run:

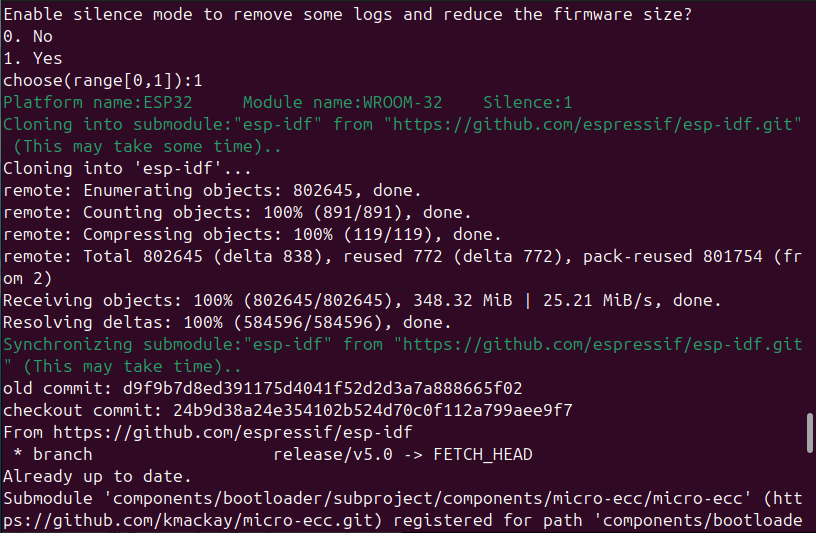
./build.py build

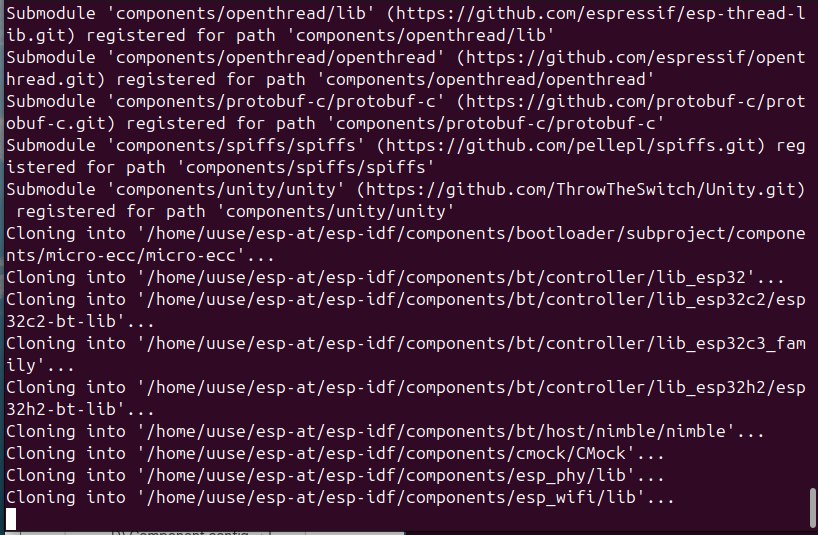
#### **A) Select your ESP Model**

* We selected Platform ESP32; therefore, write 1.
* Also We selected ESP32 WROOM 32; therefore, write 1.

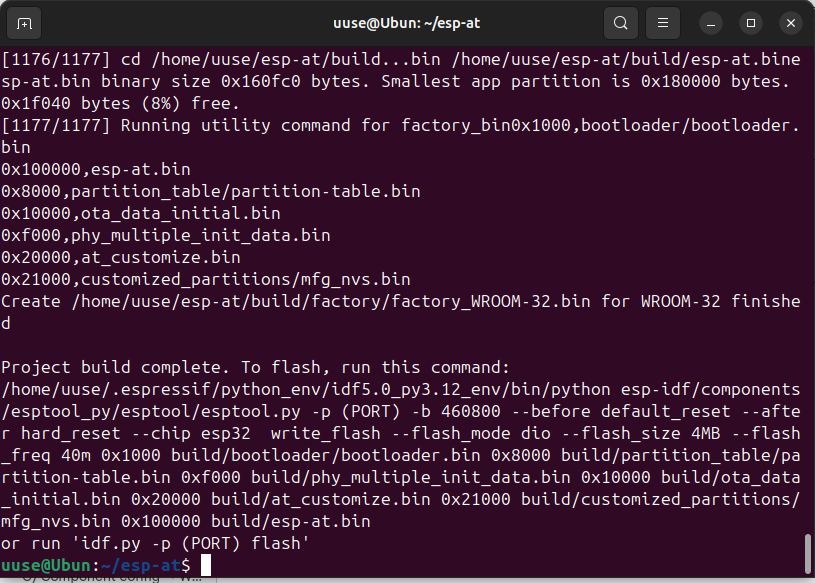


IMPORTANT: Choose silence mode as 1, Silence Mode configuration is not accessible after this comment. It can cause IRAM Overflow.





* If it is stuck in the wifi library, wait (If you have a bad pc, open 1 episode series :D ); this part may take a while. Don’t Panic.

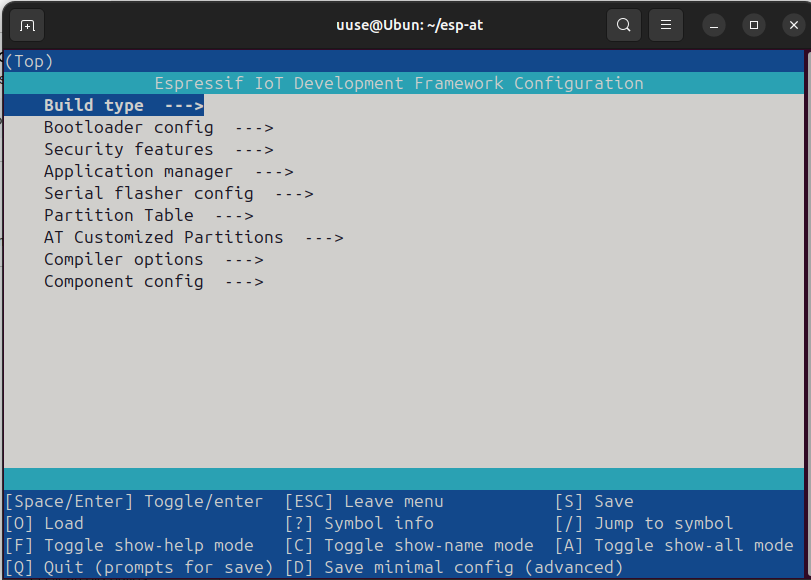


After build you can directly flash (As a default configuration, just WIFI is enabled) or you can follow step 4.

### **4. Configure AT Firmware (menuconfig)**

Run:

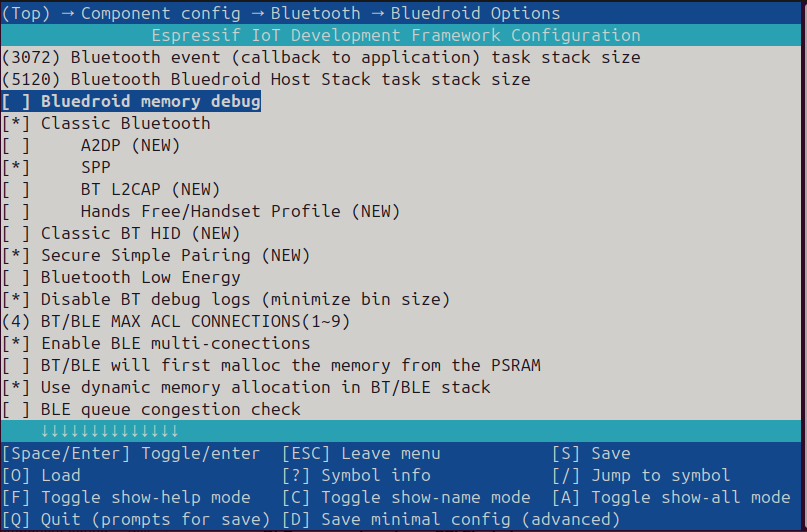
./build.py menuconfig



1. **Go to Component config** Navigate into the **Component config** menu.
2. **Go to Bluetooth** Inside Component config, select **Bluetooth**.
3. **Check Bluetooth has a \*** Make sure **Bluetooth** is selected (marked with an asterisk \*).
4. **Go to Host** Inside Bluetooth, enter the **Host** submenu.
5. **Check Bluedroid-dualmode has a \*** Verify that **bluedroid-dualmode** is enabled (marked with \*).
6. **Exit from Host** Exit the **Host** submenu.
7. **Go to Bluedroid options** Enter the **Bluedroid options** submenu.
8. **Disable Bluetooth Low Energy** Find the **Bluetooth Low Energy** option and **disable** it (remove the \*).
9. **Enable Classic Bluetooth** Make sure **Classic Bluetooth** is **enabled** (marked with \*).

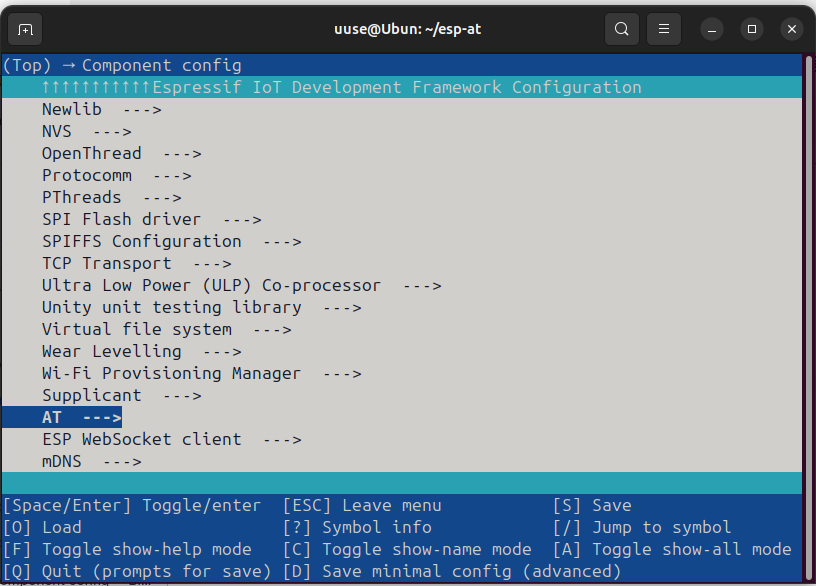
Under **Classic Bluetooth**, enable **SPP** (Serial Port Profile).

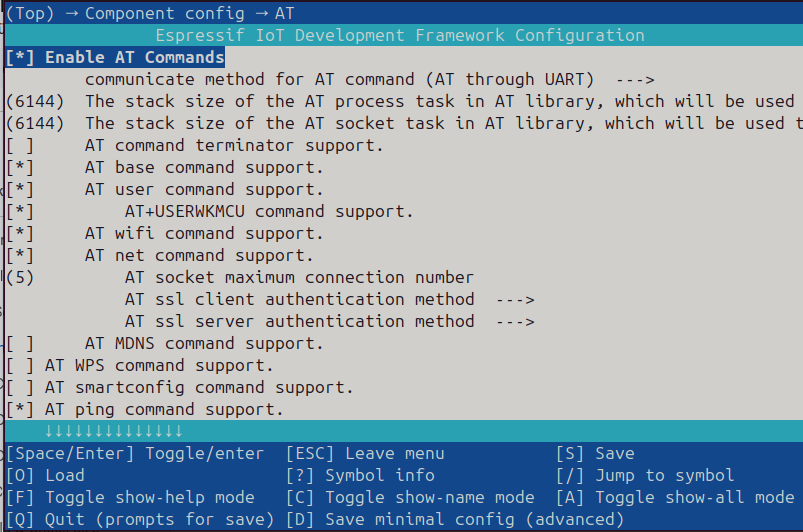
1. **Press “S” and save the configuration** Press the **S** key to save your configuration changes.

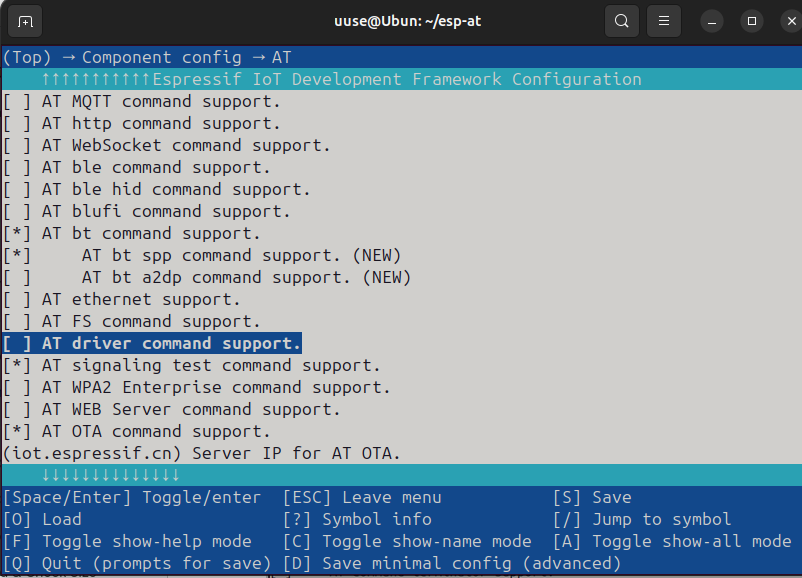


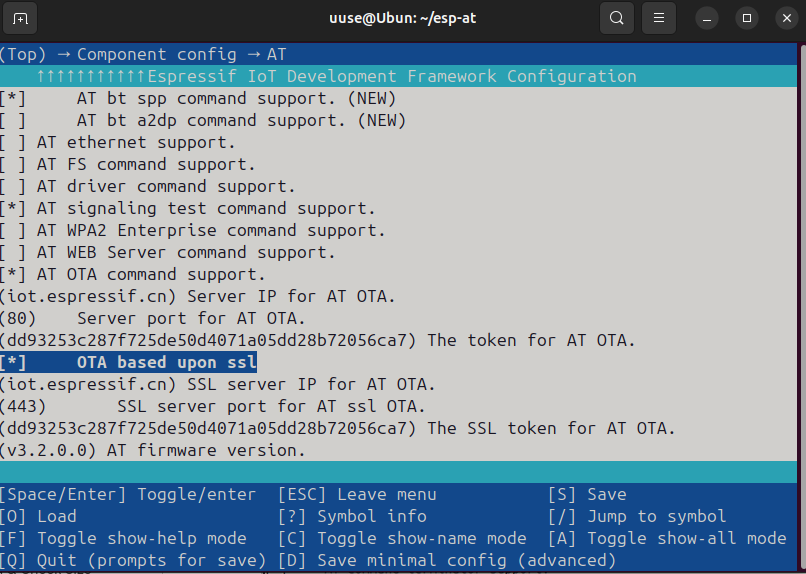
**AT Configuration**

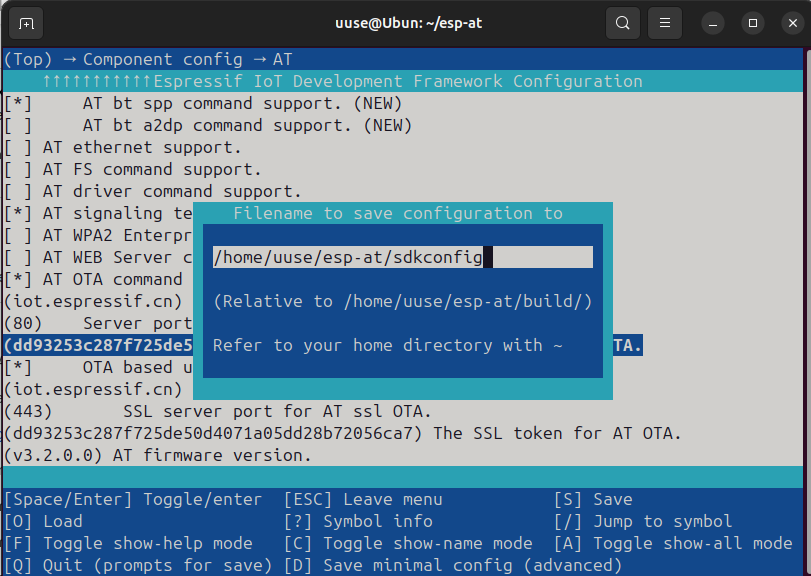
1. Return to Component config menu.
2. Enter the AT menu.
3. Enable AT Base command support (has \*).
4. Enable AT User command support (has \*).
5. Enable AT+USERWKMCU command support (has \*).
6. Enable AT WiFi command support (has \*).
7. Enable AT Net command support (has \*).
8. Enable AT Ping command support (has \*).
9. Enable AT BT SPP command support (has \*).
10. Enable AT Signaling Test command support (has \*).
11. Enable AT OTA command support (has \*).
12. Save and Exit

****

****

****

****

****

### **5. Build & Check Size**

#### 5.1 Build firmware:

./build.py build

#### 5.2 Check memory usage:

-OPTIONAL

./build.py size-components

Ensure **IRAM** usage is under ~64 KiB with no overflow.

### **6. Flash & Monitor**

Connect your ESP32 board via USB-to-UART. Then:

idf.py -p /dev/ttyUSB0 -b 115200 flash monitor

* If permission denied: sudo usermod -a -G dialout $USER, log out/in.
* If you want to just flash use this code :

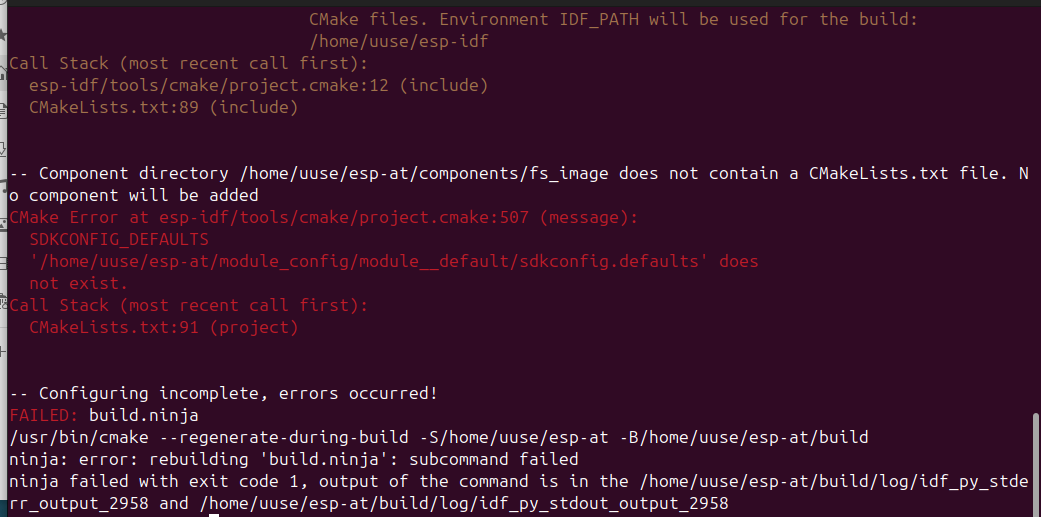
idf.py -p /dev/ttyUSB0 -b 115200 flash monitor

**Monitor controls**:

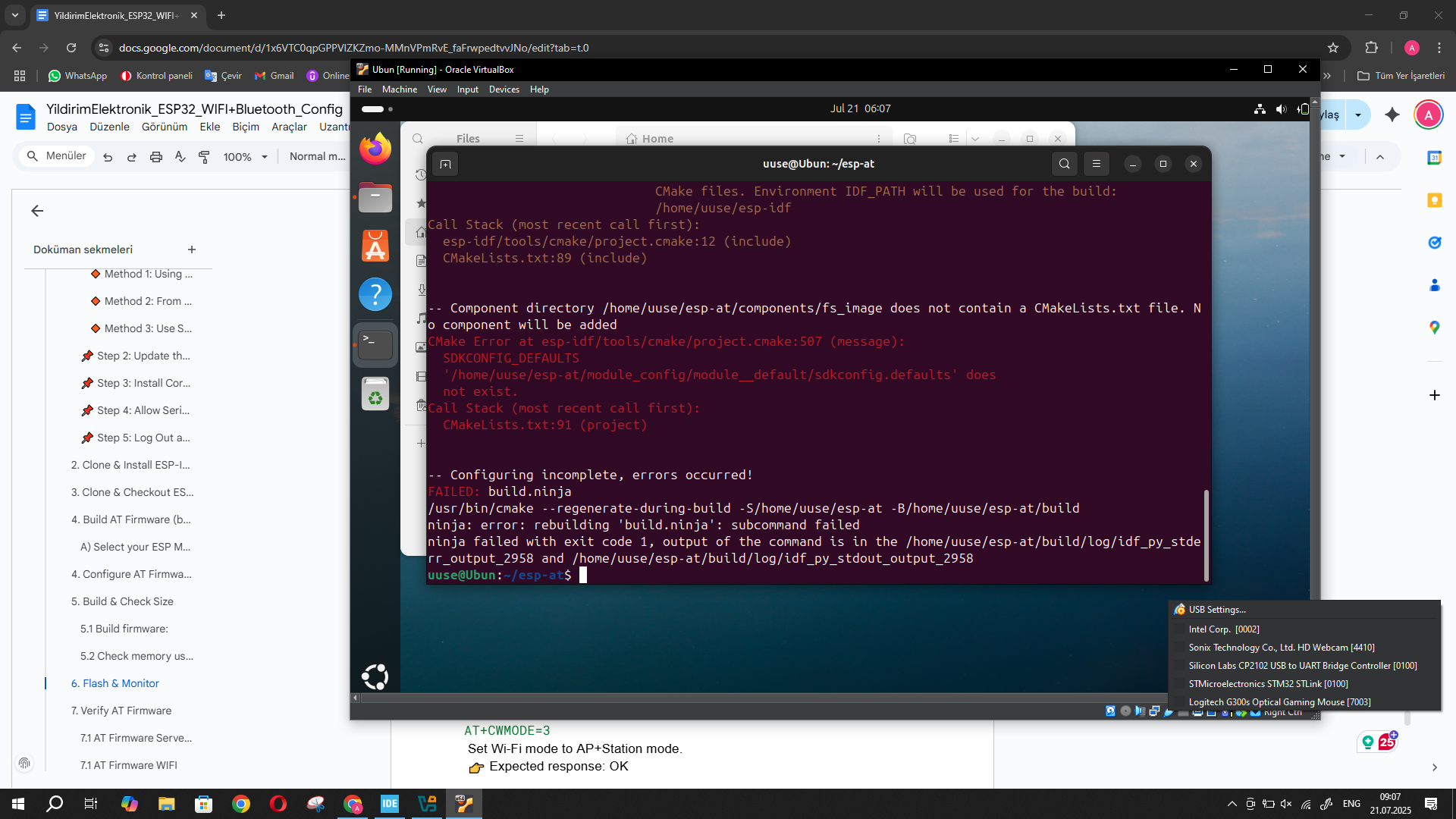
* Exit: **Ctrl-] (US) (Ubuntu has US keyboard so , press CTRL+Ü)**

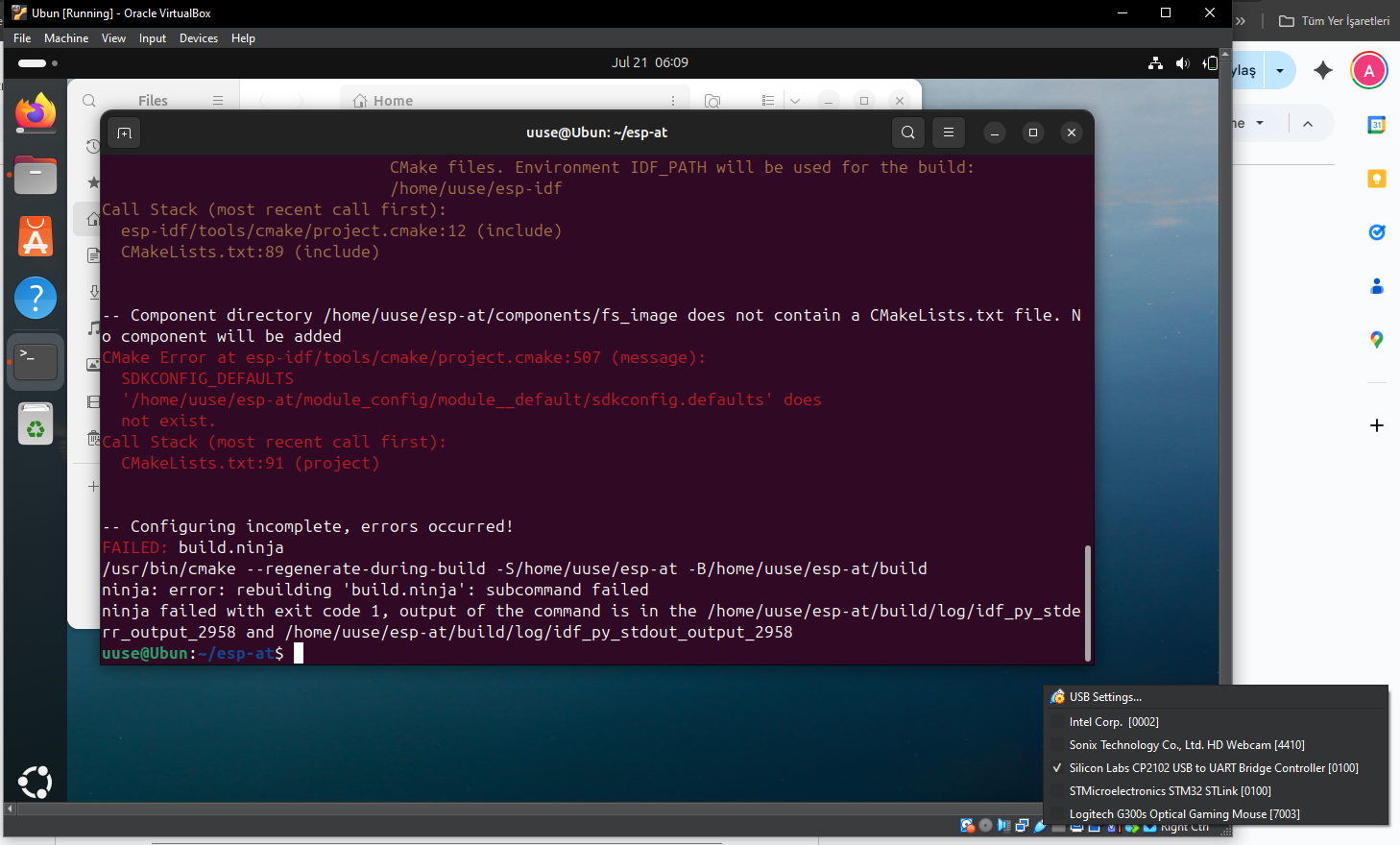
**Possible Errors:**

Error 1: USB port not connected.

****

**Solution:**

****



To resolve the error, right-click on the USB icon located at the bottom right corner of the screen. In the window that appears, locate the device and check the box next to its name to enable it. Then, run the command again.

idf.py -p /dev/ttyUSB0 -b 115200 flash monitor

### **7. Verify AT Firmware**

#### 7.1 AT Firmware Server Wifi

AT  
 Basic communication check — verifies if ESP32 is responding.  
 👉 Expected response: OK

AT+RST  
 Resets the module to clear previous states.  
 👉 Expected response: OK (followed by reboot messages)

AT+CWMODE=3  
 Set Wi-Fi mode to AP+Station mode.  
 👉 Expected response: OK

AT+CWSAP="Ozgun","123456789",6,3,4,0  
 Configure Access Point SSID, password, channel, encryption, max connections, and visibility.  
 👉 Expected response: OK

AT+CIPMUX=1  
 Enable multiple TCP connections.  
 👉 Expected response: OK

AT+CIPSERVER=1,5000  
 Start TCP server on port 5000.  
 👉 Expected response: OK

#### 7.1 AT Firmware WIFI

1. **AT** Basic communication check — verifies if ESP32 is responding.  
    👉 **Expected response:** OK
2. **AT+GMR** Displays the firmware and SDK version information.  
    👉 **Expected response:** Firmware version details.
3. **AT+BTINIT=1** Initializes the Bluetooth stack.  
    👉 **Expected response:** OK
4. **AT+BTINIT?** Checks whether the Bluetooth stack is initialized.  
    👉 **Expected response:** +BTINIT:1
5. **AT+BTNAME="ESP32\_Bluetooth"** Sets a visible Bluetooth name for the device.  
    👉 **Expected response:** OK
6. **AT+BTSCANMODE=2** Makes the ESP32 discoverable and connectable.  
    👉 2 → Discoverable and Connectable  
    👉 **Expected response:** OK
7. **AT+BTSCAN=1,1** Starts Bluetooth device scanning (inquiry scan).  
    👉 **Expected response:** Detected devices will be listed as +BTSCAN: ...
8. **AT+BTSPPCONN=<mac\_address>** Connects to a Bluetooth device using its MAC address.  
    👉 **Expected response:** OK or CONNECT
9. **AT+BTSPPSEND=0,5** Begins transmission of 5 bytes over the SPP connection (Link ID 0).  
    👉 This is typically followed by sending the string HELLO.