

---

**Skuttlebot**

235 E Ocean Blvd  
Suite 800  
Long Beach, CA, 90803

# Skuttlebot Dev Kit Software Instructions

---

We're so glad to have you join us in the development of Skuttlebot! This guide is meant to provide team specific information with regards to the Software pertaining to the Skuttlebot Command Unit- Dev Kit Assembly

<b>What is Skuttlebot?</b>	<b>2</b>
Our mission	2
The team	2
<b>Setup:</b>	<b>3</b>
<b>Updating</b>	<b>4</b>
Command Unit	4
Updating PC	5
<b>Startup</b>	<b>6</b>
Hardware	6
PC	6
<b>Revisions</b>	<b>7</b>

---

## What is Skuttlebot?

### Our mission

Skuttlebot aims to create personal AIs that interact with the user through drones, 'skuttling' between devices. We seek to do this by creating both hardware and offboard software applications that can take advantage of existing systems resources. End result being an AI system that has the owners well-being at heart, able to be a lifelong companion as humanity enters its next wave of tech integration.

### The team

We have a core team led by our leading mad scientist, but we also look strongly to the community to join us in the development of Skuttlebot. The reason for this, is that we strongly believe that AI development should and must be open to as many people as possible. We are developing the framework and the core OS but we encourage developers to create their own applications and subroutines. Our goal is to create an open marketplace for such apps, similar to many other devices.

---

## Setup:

The initial phase of setup involved downloading all of the needed files from Github.

- 1) Navigate to a target location on your file where you want the skuttlebot files to reside.
- 2) If you have not received the SkuttlebotDevKit.bat via a DevKit purchase move to step 3. Otherwise place the SkuttlebotDevKit.bat here and execute it as administrator. It will then proceed to create all needed files automatically. Skip to step 4.
- 3) Go to <https://github.com/skuttlebot/DevKit>, download and extract the files to the target folder.
- 4) Go into Arduino/Skuttleshell and edit the config file to include your local wifi info. This will be used as default. When this is complete, open a terminal from the root folder and use the following command:  
`git update-index --no-assume-unchanged path/to/config.h`  
This will prevent git from tracking this file and overwriting during future updates.
- 5) At this point the folders will be setup as follows:
  - SkuttlebotDevKit (root directory)
    - Arduino (device code)
      - Libraries (needed libraries including customized ones)
      - Skuttleshell (command module specific code)
    - Docs (instruction manuals, STEP files)
    - PC (PC side code)
      - Server (server specific code)
      - UI (UI, renderer, etc)
    - gitignore
    - Install&Update.bat (use this to update to latest released code (note this will overwrite existing code))
    - LICENSE
    - package-lock.json
    - package.json
    - Start.bat (use this to start the PC application)
- 6) Open the Arduino IDE and confirm the following are installed, or manually install the following libraries:

- 
- ☐ Arduino.h
  - ☐ WiFi.h
  - ☐ ESPAsyncWebServer.h
  - ☐ ESPmDNS.h
  - ☐ WiFiUdp.h
  - ☐ ESPAsync\_WiFiManager.h
  - ☐ ArduinoOTA.h
  - ☐ Esp\_camera.h
  - ☐ ESP32Servo.h
  - ☐ math.h

## Updating

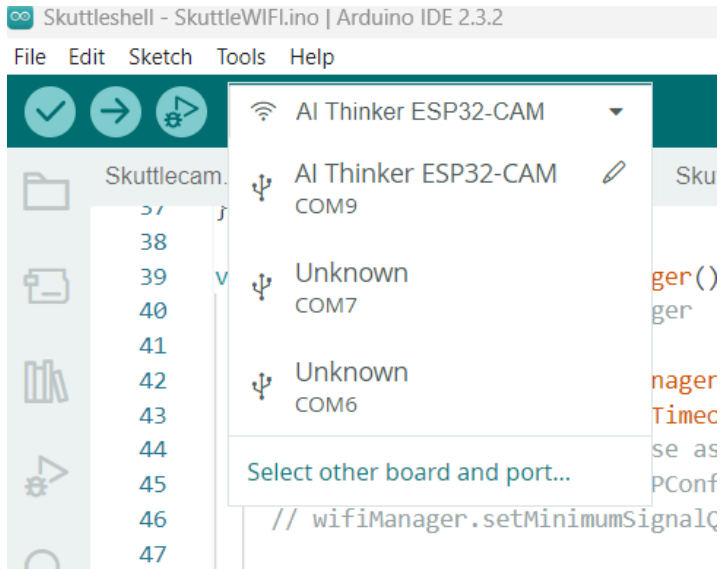
### Command Unit

The command unit is currently programmed in C++, and programmed via the Arduino IDE. The Command unit purchased via the Electronics Kit or Full Assembly, will come with the latest code installed and updates are typically done over the air. The following steps will allow for the device to be updated:

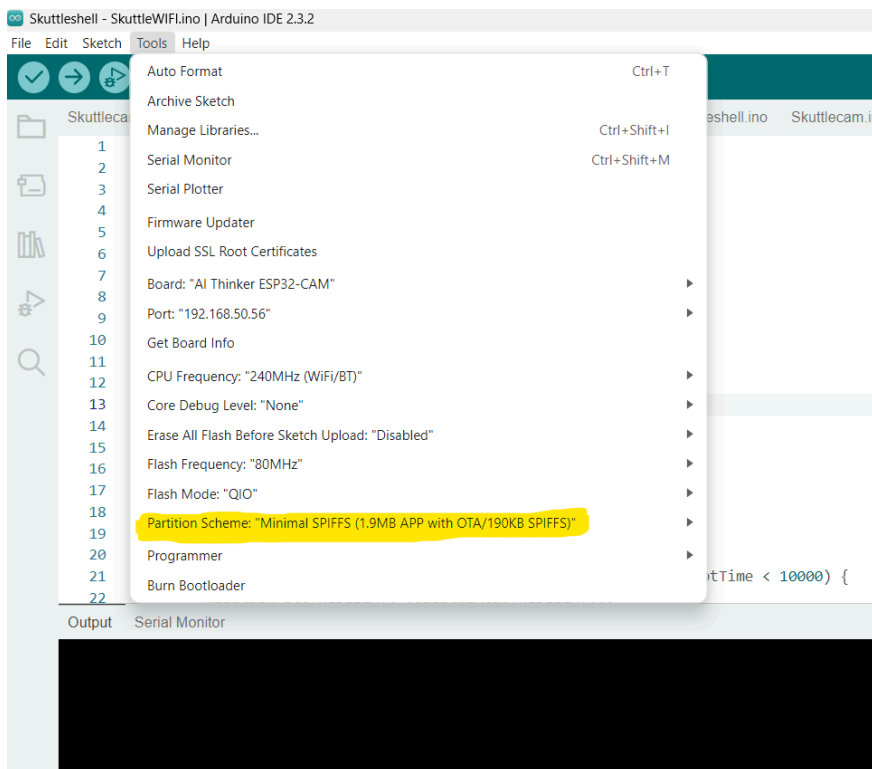
Note, if updating via the OTA has been disabled or it is desired to avoid the OTA feature, start with step 3.

1. With the Command Unit being previously charged, simply turn the device on.
2. The device will attempt to access the local network. If unsuccessful it will default into Access Point mode, creating its own server with the IP 192.168.4.1 from any browser. Enter the following credentials: Password: password on the ESP32AP network to bring up a UI where updated network information may be entered. The device will remember this information until new code is again uploaded, so it is advisable to change the information in the config.h to your local login credentials. Jump to step 4.
3. OPTIONAL This is only needed if the device has been independently purchased or if a prior upload has deactivated the OTA functionality: Connect the USB-TTL adaptor to the ESP32 and the appropriate connector cord to your PC
4. Open your Arduino IDE and then open the device pulldown. If accessing via the USB-TTL be sure to select the correct port the device is on. Otherwise connect to the ESP32

network.



5. Ensure the Partition Scheme is set for “Minimal SPIFFS (1.9MB APP with OTA/190KB SPIFFS):



6. Compile and upload your code normally. The device will automatically store your information, validate the code, prior to utilizing the new code.

## Updating PC

---

To update the PC with the latest release, simply use the Install&Update.bat file, received when obtaining any Dev kit from Skuttlebot.com. Alternatively, files may be downloaded from the Skuttlebot project on github at <https://github.com/skuttlebot/DevKit> Note that a reinstall may overwrite any unique changes made, and it is advisable you backup previous work prior to updating.

## Startup

### Hardware

To start up the device, simply turn on the powerswitch to power the onboard ESP-32. The system will then operate the last uploaded code. If the latest code from the Arduino folder has been updated, the device should first attempt to connect to the local WiFi server, using the credentials in the config.h file.

If unsuccessful, it will default to AP mode, creating its own server. A smartphone or pc may connect to this access point by connecting to the 'ESP32AP' network using the password 'password'. (these may be changed in the SkuttleWiFi.ino). Upon connecting, in any browser, go to the IP: **192.168.4.1**. At this point, the credentials for the local WiFi may be entered. Once the device can connect, it will close the AP and proceed into normal operation.

During normal operation, the device sets up a series of websocket servers and awaits a client. When a client is established communications are set up and the device will send a periodic heartbeat with basic information, while setting up listeners for commands from the PC.

Commands received from the PC are executed depending on the type of command, ranging from movement, audio, or video transmission.

### PC

Starting up the PC side of the device is accomplished by running the Start.bat file from the root directory. This will bring up a very basic Skuttlebot GUI. Details on the GUI will be forthcoming.

---

## Revisions

Draft	Initial Release	5/9/2024