

# HubVolt Setup Guide

---

## Table of Contents

1. Installing Arduino IDE
2. Installing Required Libraries
3. Connecting the HubVolt Device
4. Uploading Code to ESP32
5. Configuration via Access Point
6. Project Code & Libraries

## 1. Installing Arduino IDE

### Step 1: Download Arduino IDE

Visit: <https://www.arduino.cc/en/software>

## Downloads



### Arduino IDE 2.3.6

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

#### DOWNLOAD OPTIONS

**Windows** Win 10 and newer, 64 bits

**Windows** MSI Installer

**Windows** ZIP file

**Linux** AppImage 64 bits (X86-64)

**Linux** ZIP file 64 bits (X86-64)

**macOS** Intel, 10.15: "Catalina" or newer, 64 bits

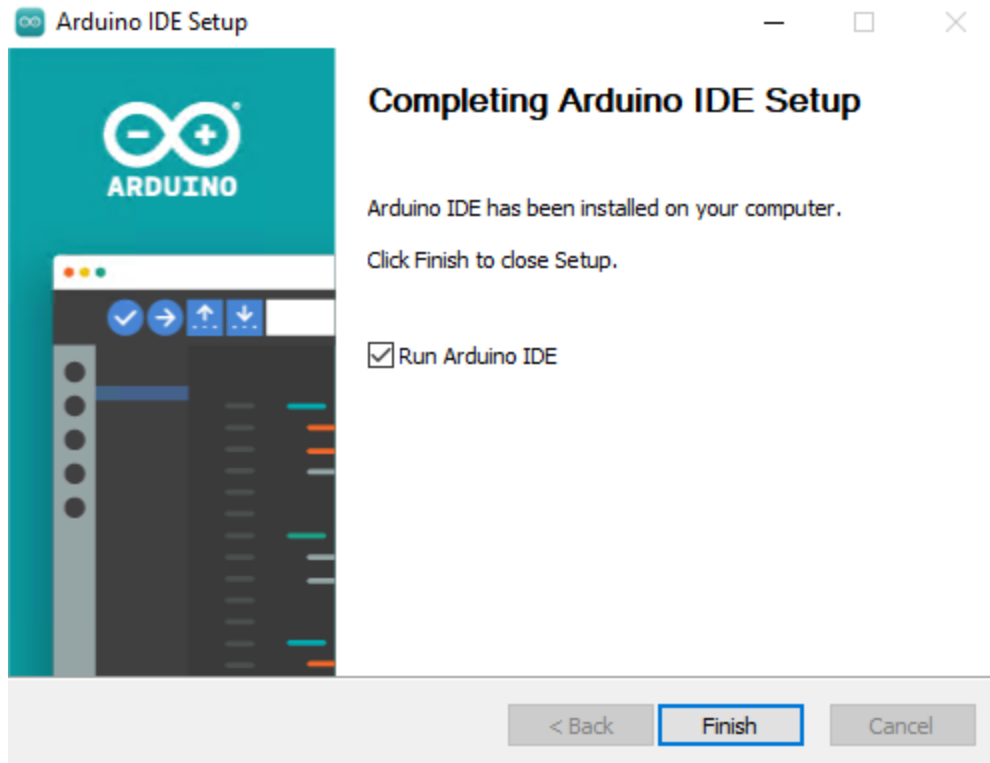
**macOS** Apple Silicon, 11: "Big Sur" or newer, 64 bits

[Release Notes](#)

Download the IDE for your operating system (Windows, macOS, or Linux).

### Step 2: Install and Launch

Run the installer and follow the setup instructions.



Launch the Arduino IDE after installation.

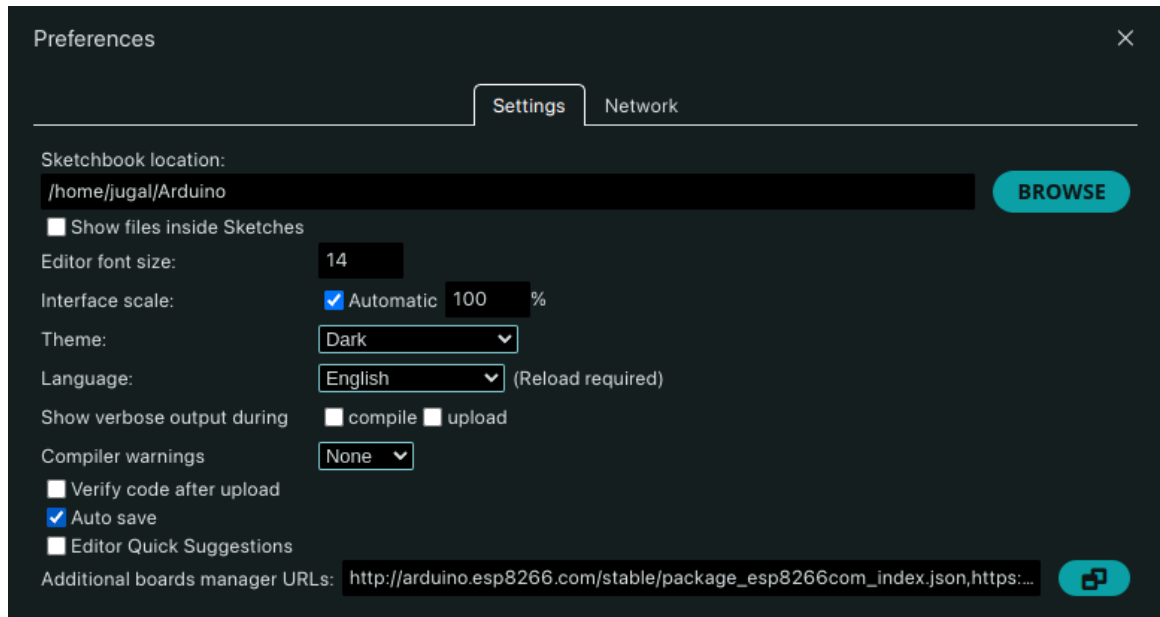
## 2. Installing Required Libraries

### Step 1: Add ESP32 Board Support

Go to File > Preferences.

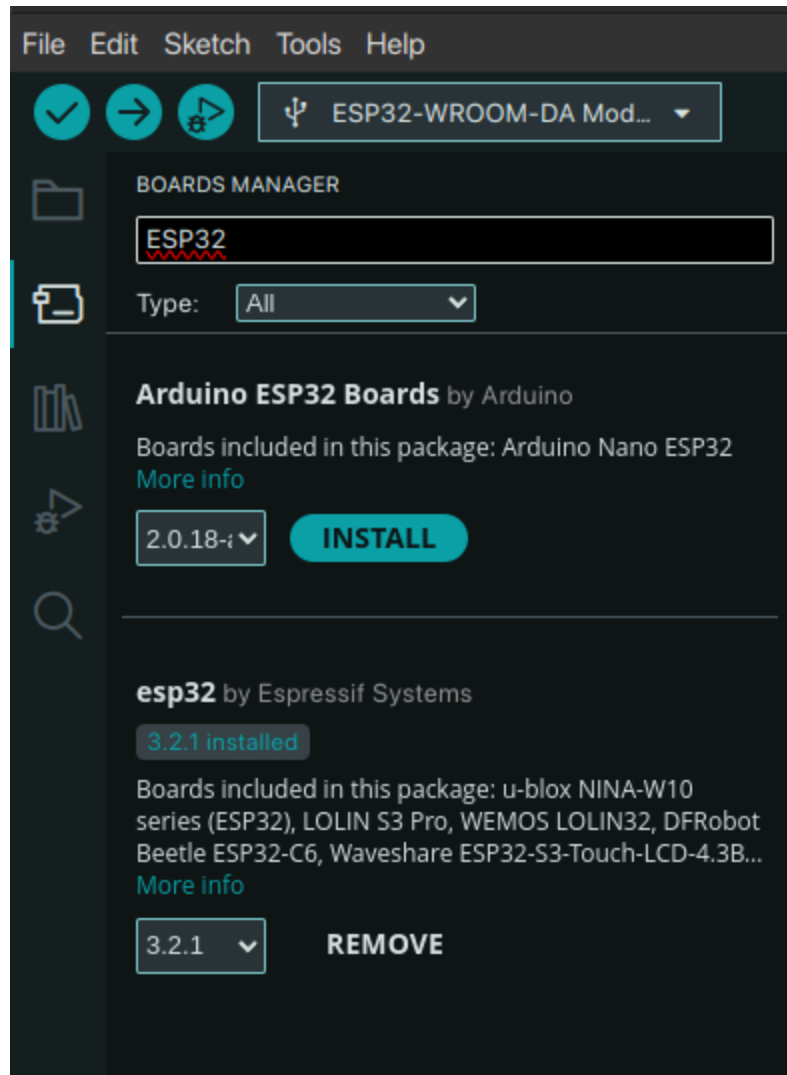
In 'Additional Board Manager URLs', paste:

[https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package\\_esp32\\_index.json](https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json)



## Step 2: Install Board Package

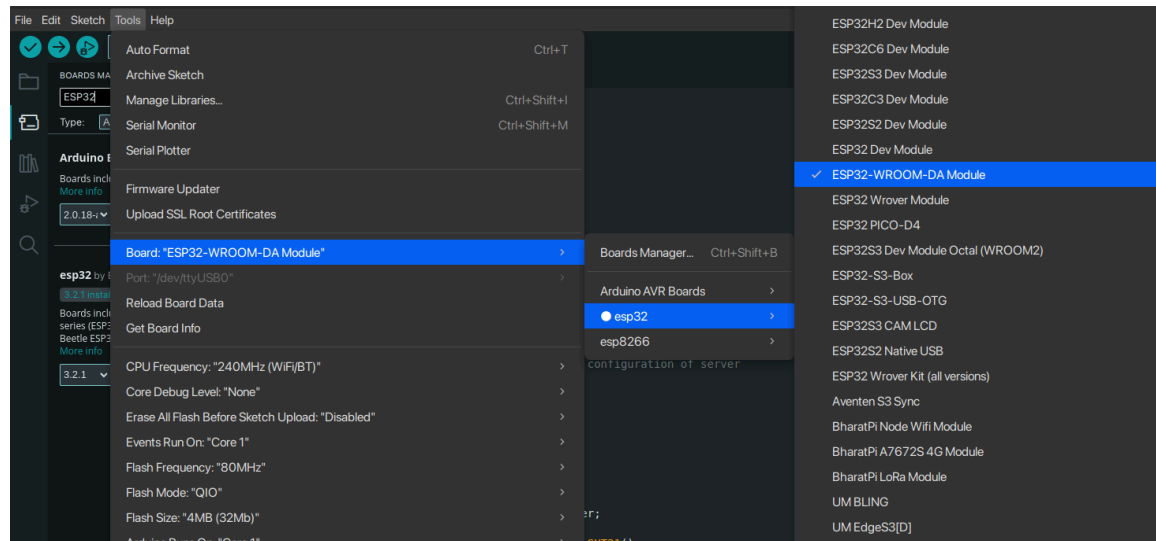
Go to Tools > Board > Boards Manager.



Search for 'esp32' and install 'esp32 by Espressif Systems'.

### Step 3: Select Board

Go to Tools > Board and select 'ESP32 Dev Module'.

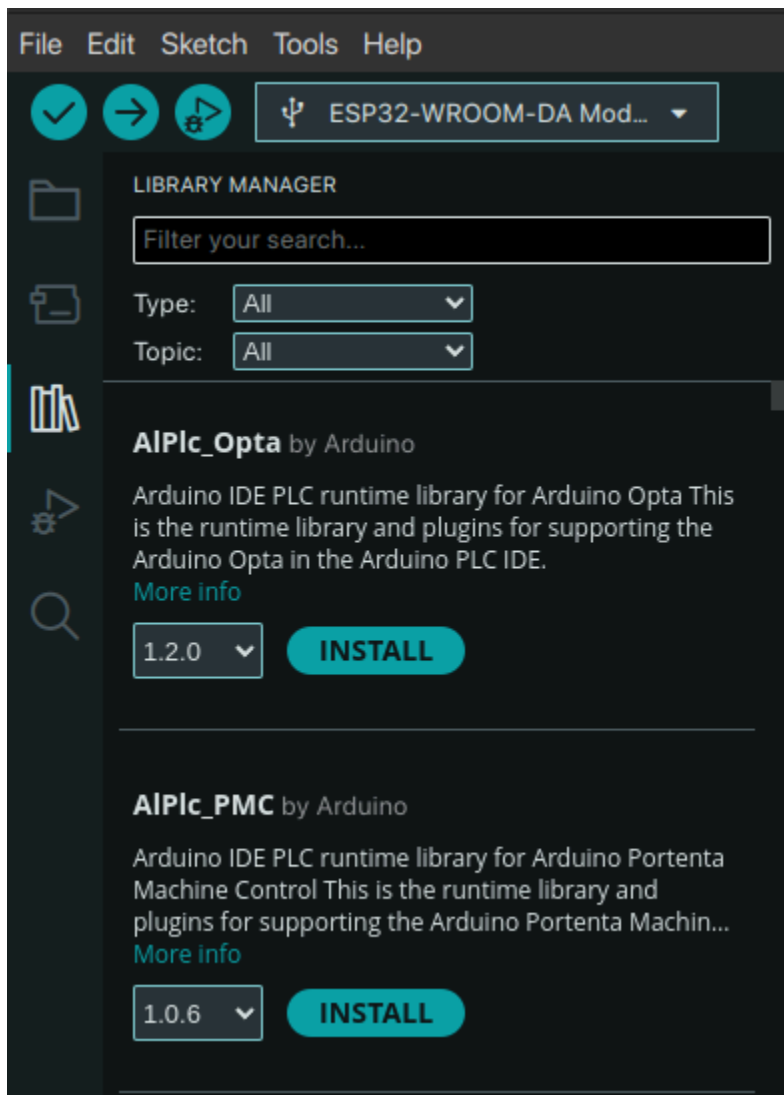


## Step 4: Install Required Libraries

In case of hubvolt there is no need to download the libraries because the libraries used are already installed because of the esp32 by Espressif Systems board manager package. In case of any trouble this is for the reference which library is used for what purpose and further you can see their documentation at the official website of ESP32 Documentation.

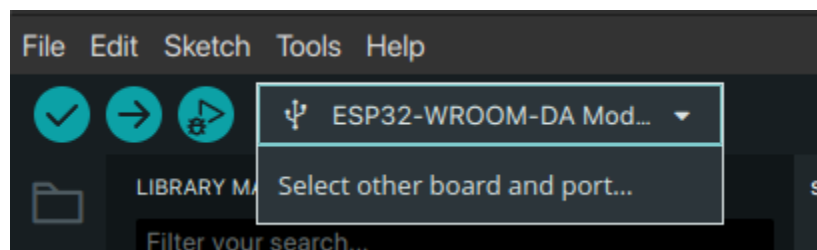
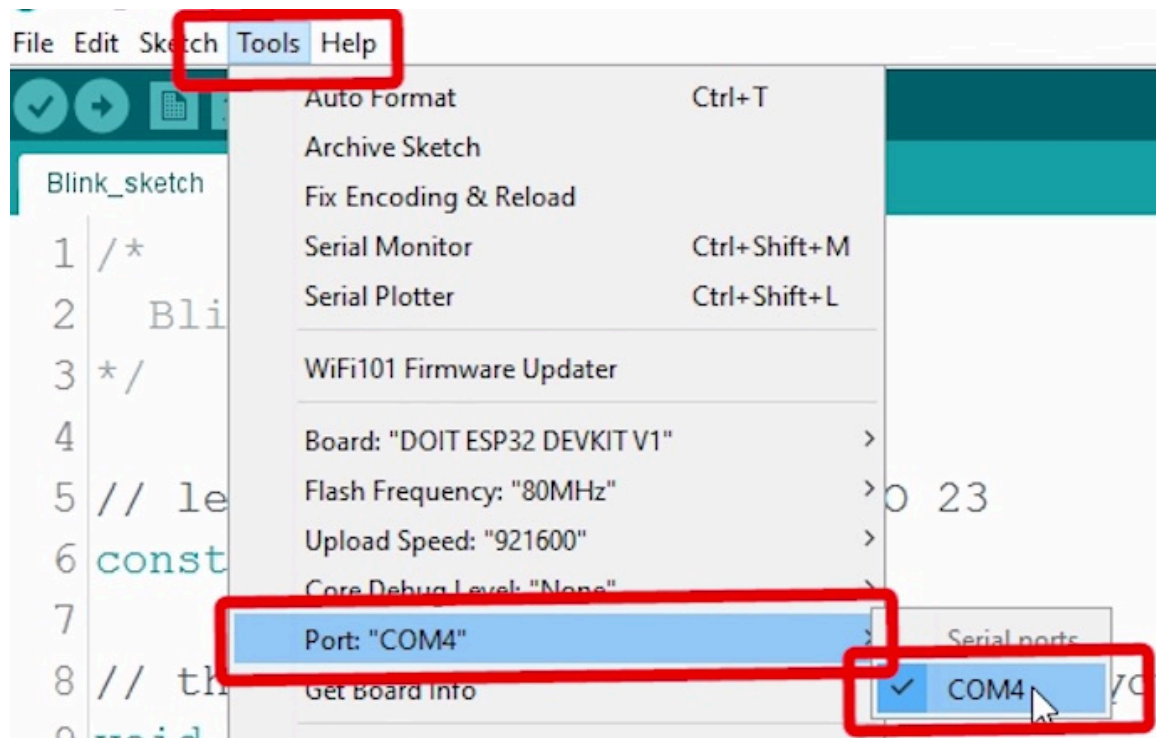
- WiFi: Built into ESP32 core
- WebServer: Built into ESP32 core
- Preferences: Built into ESP32 core
- WiFiClient: TCP Client (built-in ESP32)
- esp\_wpa2: For enterprise WiFi auth (ESP32)
- esp\_wifi: Low-level WiFi functions (ESP32)
- Wire: I2C communication

You can use the Library manager tab to do so.



### 3. Connecting the HubVolt Device

1. Connect your ESP32-WROOM-DA module to your PC using a data-capable USB cable.
2. Open Arduino IDE.
3. Go to Tools > Port and select the correct COM port.



Once the esp32 is connected you will see this tab in BOLD letters.

## 4. Uploading Code to ESP32

### Step 1: Download Firmware

GitHub Repository:

[https://github.com/DankJugal/SRIP\\_FINAL\\_RESOURCES](https://github.com/DankJugal/SRIP_FINAL_RESOURCES)

Open the `.ino` file for the HubVolt firmware and ensure all required libraries are installed.



```
SRIP_FINAL_RESOURCES / HubVolt / HubVolt_v1.ino

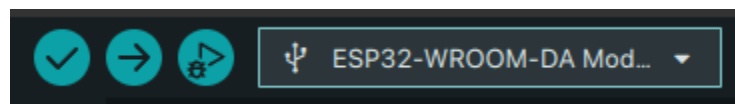
DankJugal first commit 54bc99d · 4 days ago History

Code Blame 340 Lines (290 loc) · 11.1 KB

1 #include <WiFi.h>
2 #include <WebServer.h>
3 #include <Preferences.h>
4 #include <WiFiClient.h>
5 #include <esp_wpa2.h>
6 #include <esp_wifi.h>
7 #include <Wire.h>
8
9 #define bootButtonPin 0 //Flash configuration of server
10 #define POWER_CONTROL_PIN 17 //USB hub power
11
12 typedef String string;
13
14 // Object declarations
15 Preferences powerAccessController;
16 WebServer configServer(80);
17
18 // Access Point credentials
19 const char* ApSSID = "ESP32-Setup";
20 const char* ApPassword = "configureme";
21
22 // Device Name
23 String device_name = "";
24
25 // Web page handler
26 void handleRoot() {
27     String html = R"(<script>function myFunction(){
28         var x = document.getElementById('password');
```

## Step 2: Upload the Code

Click the Upload button in Arduino IDE.



Click on the arrow button for uploading the code to ESP32.

## Step 3: Check Serial Monitor

Open Tools > Serial Monitor and set baud rate to 115200.

## 5. Configuration via Access Point

### Step 1: Connect to the Access Point

Connect to WiFi network: ESP32-Setup

Password: configureme



### Step 2: Open Configuration Portal

Open browser and visit: <http://192.168.4.1>

←

→

↻

🛡️

Not Secure

http://192.168.4.1

## HubVolt (v1.0)

Welcome to HubVolt.

### WiFi & Server Configuration

Device Name (It should not have any space & special characters):

SSID:

Connection-Type: 

Open ▾

Username (Not needed for Open/Personal):

Password (Not needed for Open):

☐ Show Password

Server IP:

Server TCP Port:

### Step 3: Enter Configuration Details

Enter WiFi SSID, Password, server details, and other configuration values.

←

→

↻

🛡️

Not Secure

🔗 http://192.168.4.1

## HubVolt (v1.0)

Welcome to HubVolt.

### WiFi & Server Configuration

Device Name (It should not have any space & special characters):

SSID:

Connection-Type: 

Personal ▾

Username (Not needed for Open/Personal):

☐ Show Password

Server IP:

Server TCP Port:

### Step 4: Save & Reboot

Click Save. The device will reboot and attempt to connect to your network.



**Configuration saved. Device is ready now.**

## 6. Reconfiguration of ESP32

### Method 1 : Using Reset and Boot Buttons

1. **Press the Reset button**, then **immediately press the Boot button**.
2. Wait until the **red LED starts blinking** — this indicates that the ESP32 has entered **AP mode** again.
3. Connect to the **ESP32-Setup** WiFi network and repeat the configuration steps.

### Method 2 : Power-Cycle with Boot Press

1. **Power off** the ESP32 device.
2. While powering **on**, **immediately press and hold the Boot button**.
3. When the **red light blinks**, the device is again in **AP mode** and ready for reconfiguration.

This allows users to re-enter the configuration portal if credentials or settings need to be updated after the initial setup.