BedJetWebSchedule

Download, Flash, and First-Time Setup Guide (ESP32-S3 N16R8 + Arduino IDE)

Last updated: January 17, 2026

This document walks a new user through: (1) downloading the BedJetWebSchedule source from GitHub, (2) purchasing a compatible ESP32-S3 N16R8 dev board (16MB flash + 8MB PSRAM), (3) installing Arduino IDE and the Espressif ESP32 board package, (4) selecting the correct board/flash/partition settings, (5) uploading the firmware, and (6) completing the first-boot Wi-Fi + BedJet configuration portal.

Important: In some listings the board is described as “N16R8”. That means **16MB Flash** + **8MB PSRAM** (megabytes, not gigabytes).

At-a-glance checklist

- Buy an ESP32-S3 N16R8 board (USB-C or micro-USB) + a data-capable USB cable.
- Install Arduino IDE 2.x (Windows / macOS / Linux).
- Add Espressif board manager URL and install the *esp32* platform package.
- Install required libraries (at minimum: NimBLE-Arduino).
- Open **BedJetWebSchedule.ino** in Arduino IDE and select the correct board + flash/PSRAM + partition scheme.
- Upload (flash) the firmware to the ESP32-S3.
- Join the setup Wi-Fi AP **BedJetSetup-XXXX** and configure Wi-Fi + BedJet MAC + hostname.
- Open the web UI using **http://bedjetweb.local/** or the device IP.

1) Download the project from GitHub

Repository: **digitalrcs/BedJetWebSchedule**. The README describes features, first-boot provisioning, and how to access the UI after setup.

Option A - Download ZIP (no Git needed)

- Open the repository page in a browser.
- Click **Code** → **Download ZIP**.
- Extract the ZIP to a folder on your computer (for example: *C:\Projects\BedJetWebSchedule*).
- In the extracted folder, locate **BedJetWebSchedule.ino** (root of the repo).

Option B - Git clone (recommended for updates)

If Git is installed:

```
git clone https://github.com/digitalrcs/BedJetWebSchedule.git
```

Then open the cloned folder and proceed to the Arduino IDE steps.

2) Buy the ESP32-S3 N16R8 development board

Search Amazon (or similar) for one of these phrases. The exact brand varies, but specs should match:

- **ESP32-S3 N16R8** development board
- **ESP32-S3-DevKitC-1 N16R8**
- “**16MB Flash**” and “**8MB PSRAM**” in the listing
- USB-C (preferred) or micro-USB; ensure it supports data, not charge-only

Typical listings explicitly call out “N16R8 configuration: 16MB flash and 8MB PSRAM”.

What you need on your desk

- ESP32-S3 N16R8 dev board
- USB data cable (USB-C or micro-USB depending on the board)
- A BedJet with BLE (BedJet 3 is commonly used)
- 2.4 GHz Wi-Fi network credentials (SSID + password)

3) Install Arduino IDE

Install Arduino IDE 2.x for your operating system. After installation, launch Arduino IDE once so it can finish initial setup.

4) Install Espressif ESP32 board support

Espressif documents the supported install flows. The most common Arduino IDE approach uses the Boards Manager.

Step-by-step (Arduino IDE Boards Manager)

- Arduino IDE → **File** → **Preferences**.
- In **Additional Boards Manager URLs**, add:

```
https://espressif.github.io/arduino-esp32/package_esp32_index.json
```

If that URL is blocked by your network, an alternate mirror commonly used is:

```
https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index
.json
```

- Click **OK** to save Preferences.
- Arduino IDE → **Tools** → **Board** → **Boards Manager…**
- Search for **esp32** and install **“esp32 by Espressif Systems”**.

If installation fails due to network restrictions, try again on a different network or review Espressif’s official install guide for alternate methods.

5) Install required Arduino libraries

The project is tested primarily with NimBLE-Arduino (for BLE) plus Arduino WebServer (included in the ESP32 core).

- Arduino IDE → **Sketch** → **Include Library** → **Manage Libraries…**
- Search for **NimBLE-Arduino** (author: h2zero) and install it.
- If compilation later complains about a missing header/library, install the named library via Library Manager and re-compile.

6) Select the correct ESP32-S3 board and build settings

Plug the ESP32-S3 into your computer via USB. If Windows asks for drivers, many boards use CP210x or CH340 USB-to-UART chips; install the driver for your board if needed.

Recommended Arduino IDE settings (ESP32-S3 N16R8)

Setting	Value / Guidance
Tools → Board	Select an ESP32-S3 board definition matching your hardware (commonly “ESP32S3 Dev Module”).
Tools → Port	Select the COM/serial port that appears when you plug in the board.
Tools → USB CDC On Boot	Enabled (common default for S3 boards).
Tools → Flash Size	16MB (because N16R8 has 16MB flash).
Tools → PSRAM	Enabled (8MB PSRAM; often “OPI PSRAM” on S3 modules).
Tools → Partition Scheme	Choose a 16MB partition with a large APP region (often labeled “Huge APP” or similar).
Tools → Upload Speed	Start with 460800 or 921600; if uploads fail, lower to 115200.

Common pitfall: If you accidentally select a 4MB flash setting or a partition meant for smaller flash, the bootloader may report partition table errors. Fix by switching Flash Size to 16MB and selecting a 16MB partition scheme.

7) Compile and upload the firmware

- In Arduino IDE, click **File** → **Open…** and select **BedJetWebSchedule.ino**.
- Confirm the board + port + flash/PSRAM/partition settings from the prior section.
- Click **Verify** (checkmark) to compile.
- If compile succeeds, click **Upload** (right arrow).

If upload fails (S3 boot mode tips)

- Try a different USB cable (many charge-only cables fail).
- Lower upload speed to 115200.
- Some boards require holding **BOOT** while pressing **RESET** to enter download mode, then release BOOT after upload starts.
- Confirm you selected the correct COM/serial port.

Optional - watch serial logs

Open **Tools** → **Serial Monitor** at 115200 baud to view boot logs and connection status. This is very useful for troubleshooting Wi-Fi or BLE.

8) First-boot setup (AP provisioning portal)

On first boot (no saved config), the firmware starts a temporary Wi-Fi access point and a local web portal for configuration:

- Setup SSID: **BedJetSetup-XXXX**
- Setup Portal URL: **http://192.168.4.1/**

Provisioning steps

- Power on / reset the ESP32-S3.
- On your phone or laptop, join Wi-Fi network **BedJetSetup-XXXX**.
- Open a browser and go to **http://192.168.4.1/**.
- Enter your home Wi-Fi SSID + password.
- Enter your BedJet Bluetooth MAC address in format **AA:BB:CC:DD:EE:FF**.
- Choose DHCP or set a static IP (if static: IP, subnet mask, gateway, DNS).
- Optionally set a hostname (default: **BEDJETWEB**).
- Click **Save & Reboot**.
- Reconnect your phone/laptop back to your normal home Wi-Fi after the setup AP disappears.

How to find the BedJet Bluetooth MAC address

The easiest approach is using a BLE scanning app (Android recommended). Example: install **nRF Connect**, scan for nearby devices, and identify the BedJet entry (often named with “BEDJET”); copy its MAC address.

> Note: iOS typically does not expose BLE MAC addresses in the same way as Android. If you only have an iPhone, use a Windows PC BLE scanner, an Android device, or another method recommended by your BLE tooling.

Accessing the UI after reboot

- Try mDNS hostname: **http://bedjetweb.local/** (works on many networks/OSes; not guaranteed on all Android setups).
- Or use IP: **http://<device-ip>/** (find the device in your router DHCP client list; hostname often shows as **BEDJETWEB**).

9) Normal mode usage

- Open the main web UI.
- Click **Connect** to connect the ESP32 to the BedJet over BLE.

- Use quick controls (OFF/HEAT/TURBO/COOL/DRY/EXT-HEAT) and optional Fan/Temp/Runtime.
- Create or edit schedules; they are stored on-device and survive reboot.
- Use Import/Export to save schedules as JSON.

10) Security / network hygiene

This project is intended for trusted home networks and does not include authentication by default. Treat the device like any other IoT appliance:

- Do not port-forward the web UI to the public internet.
- If desired, place the ESP32 on a trusted VLAN/SSID.
- Keep your home Wi-Fi credentials protected; change them if you suspect compromise.

Troubleshooting

Web UI does not load

- Confirm your phone/PC is on the same LAN/VLAN as the ESP32.
- If you just used the setup AP, explicitly re-join your home Wi-Fi (phones sometimes stay “half-attached” to the setup AP).
- If bedjetweb.local does not resolve, use the IP address.

BLE connect is flaky / scan returns 0 devices

- Move the ESP32 physically closer to the BedJet for initial testing.
- Avoid USB 3.0 hubs/cables right next to the ESP32 (2.4 GHz interference).
- BedJet may advertise infrequently when idle; try pressing a button on the BedJet remote/device and retry.

Boot error: partition table exceeds flash size

- You selected a partition scheme for larger flash than your board, or selected 4MB flash settings.
- Fix: set **Flash Size** to 16MB and pick a 16MB partition scheme (often “Huge APP”). Then re-upload.

References

Primary references used to build these instructions:

- **BedJetWebSchedule** repository + README: https://github.com/digitalrcs/BedJetWebSchedule
- **Espressif Arduino-ESP32** official installation guide: https://docs.espressif.com/projects/arduino-esp32/en/latest/installing.html
- **Espressif Boards Manager** package index (stable): https://espressif.github.io/arduino-esp32/package_esp32_index.json
- **Arduino IDE** downloads: https://www.arduino.cc/en/software
- **ESPHome BedJet** docs (notes on discovering BedJet BLE MAC via BLE client tooling): https://esphome.io/components/climate/bedjet.html

If this guide is being shared with someone else, include the repository URL and ask them to follow the README first if anything differs from this document.