

# **RapidFire Mod v0.9.9 — Complete User Guide**

A comprehensive guide to the ESP32-C3 based PS4/PS5 controller rapid-fire mod and its web interface.

## **Supports:**

**PS4 Dual Shock: JDM-001 / 011 / 020 / 030 / 040 / 050 / 055**

**PS5 Dual Sense: BDM-010 / 020 / 030 / 040 / 050**

## **How the Mod Detects & Works on ANY PS4/PS5 R2 Trigger**

The “Universal” compatibility comes from understanding what the R2 trigger actually is.

Inside every PS4 Dual Shock and PS5 Dual Sense controller, the R2 trigger is not a simple digital on/off switch. It is an analog potentiometer connected to a 3.3V reference line.

**The ESP32-C3 reads this signal as an analog voltage, not a digital state.**

**This provides two major advantages:**

- It works with any potentiometer value used across all PS4/PS5 controller revisions.
- The calibration routine automatically learns your controller’s exact idle and fully-pressed voltages, ensuring accurate trigger detection regardless of hardware variation.

## **What the Hardware Mod Actually Does**

This is a hardware mod installed inside your PS4 Dual Shock or PS5 Dual Sense controller.

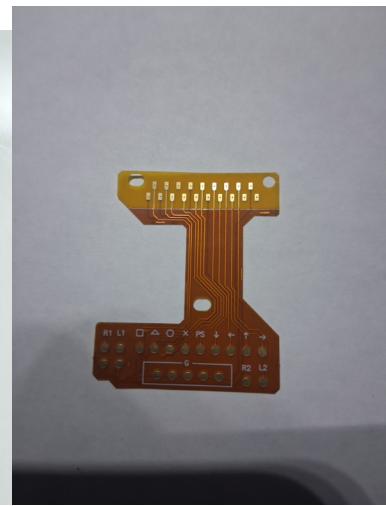
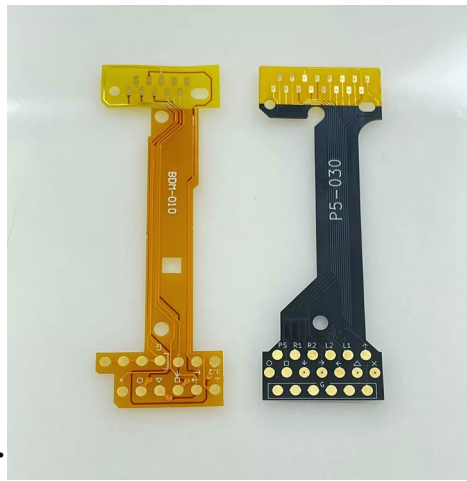
The ESP32-C3 microcontroller sits between the trigger board and the console, intercepting the R2 signal and providing advanced functionality:

- Rapid-Fire (continuous or burst modes)
- Button Remapping (map tactile buttons to Square, Triangle, Circle, or X)
- Tactical Assists (Drop Shot, Jump Shot, Auto Reload)
- Macro Playback (up to 6-step button sequences)
- Randomized Timing to reduce anti-cheat pattern detection
- You configure everything through a web interface hosted on the ESP32's own WiFi access point.

## 🔌 2. Hardware & Connection

**ESP32-C3 Super Mini** ✅ Works (cheaper, smaller)

Need an Easy remap for both controller types.



### GPIO Mapping (ESP32-C3 Super Mini)

**GPIO0**- R2 Trigger sense (ADC + Output)

**GPIO1**- Square button output

**GPIO2**- Triangle button output

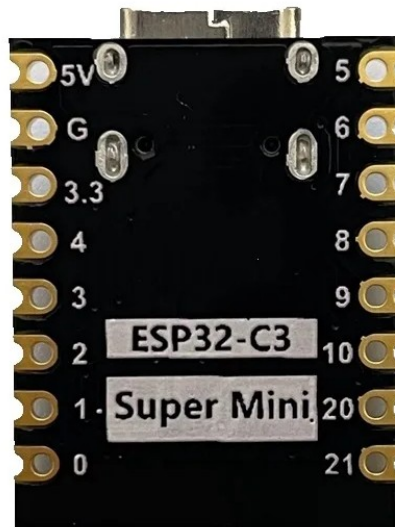
**GPIO3**- Circle button output

**GPIO4**- X button output

**GPIO5**- Tactile Switch A (input)

**GPIO8**- Onboard Status LED

**GPIO10**- Tactile Switch B (input)



### Connecting to the Web UI

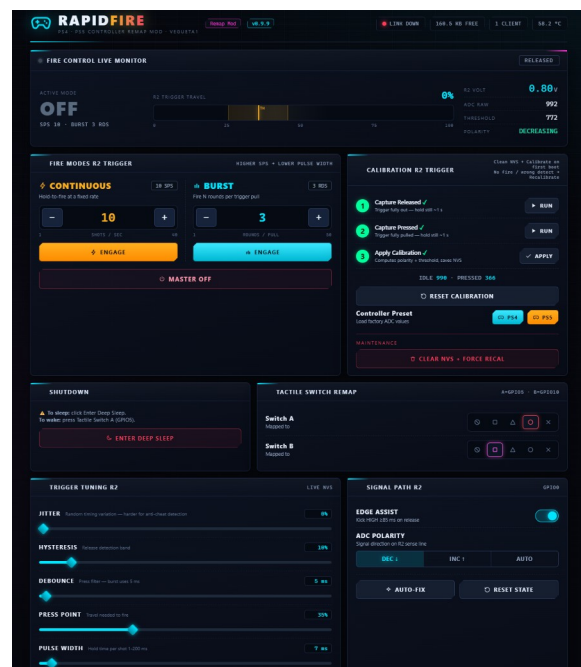
Power on the ESP32 (it creates a WiFi AP)

Connect your phone/PC to WiFi: RapidFireMod\_v0.9.9

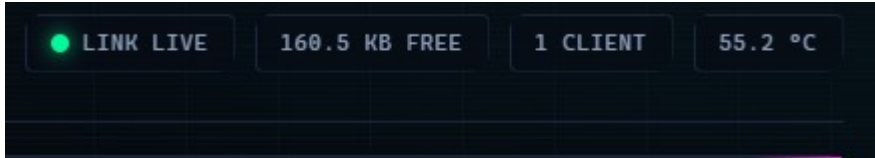
Password: 12345678

Open browser → <http://192.168.4.1>

The themed dashboard loads



### 3. Web App — Section by Section



#### Header Bar

Shows live system status:

**LINK LED** — Green = Web Socket connected, Red = disconnected.

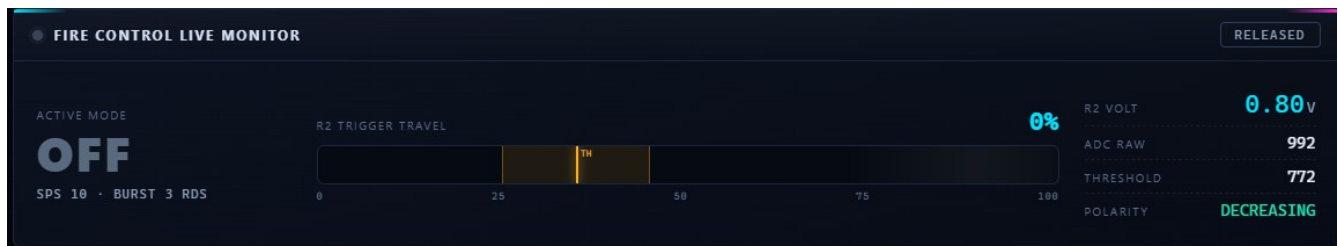
**Free RAM** — Available heap memory.

**Clients** — Number of devices connected to the AP.

**CPU Temp** — ESP32-C3 internal temperature.

### Section 1: Fire Control Live Monitor

The main dashboard showing real-time R2 trigger state.



 Use this to verify your wiring — when you pull R2, the meter should move and voltage should change. If is stuck on 100% check if controller is ON or the R2 wire is connected.

#### What it shows

Active Mode- Current mode: OFF / CONTINUOUS / BURST

R2 Trigger Travel meter- Visual bar showing 0–100% trigger pull

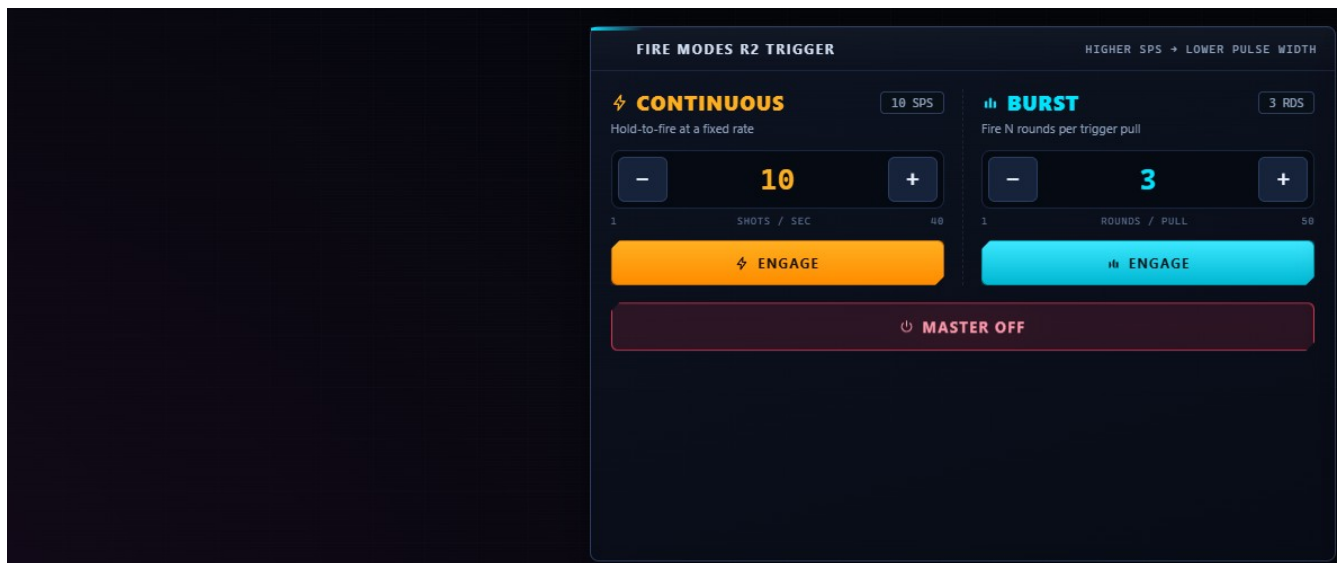
R2 Volt- Real-time voltage on R2 sense line

ADC Raw- Raw 12-bit ADC reading (0–4095)

Threshold- Current trigger-press threshold value

Polarity- Signal direction (Decreasing = PS4/PS5 default)

## ⚡ Section 2: Fire Modes (R2 Trigger)



The core rapid-fire functionality.

### CONTINUOUS Mode

Hold R2 → fires at a fixed rate (Shots Per Second)

SPS Stepper: 1 to 40 shots/second

Default: 10 SPS (safe for most games)

⚠ Higher SPS = lower max pulse width (**slider auto-adjusts**)

### BURST Mode

Pull R2 once → fires N rounds automatically

Burst Stepper: 1 to 50 rounds per pull

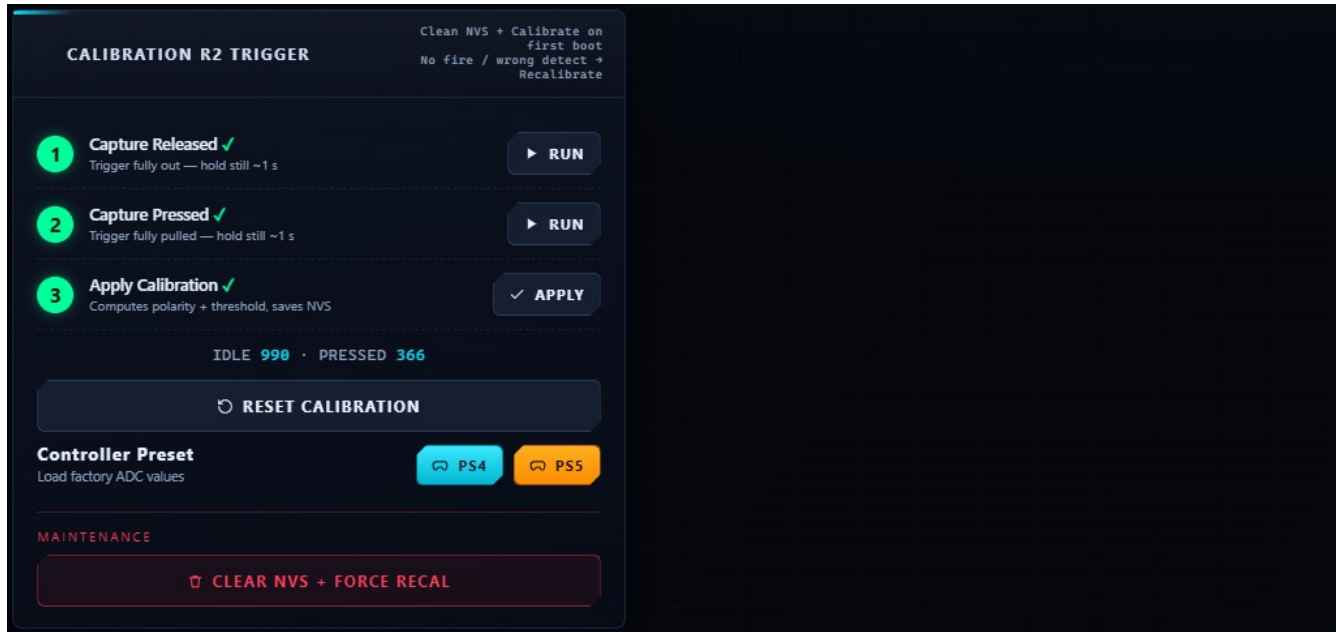
Default: 3 rounds

Great for DMRs, semi-autos, or avoiding full-auto recoil

### Master Off Button

Instantly disables rapid-fire without changing settings.

### Section 3: Calibration (R2 Trigger)



***Critical first-time setup — Run Clear NVS + Force Recal and do 3-Step Calibration Process- teaches the ESP32 what "pressed" and "released" on YOUR specific controller.***

#### 3-Step Calibration Process

**Capture Released** → Leave R2 fully out, press RUN

**Capture Pressed** → Pull R2 fully in, press RUN

**Apply Calibration** → Computes polarity + threshold, saves to NVS

#### Factory Presets (Defaults)

**PS4 button** → Loads Dual Shock 4 factory ADC values (IDLE=2234, PRESSED=667)

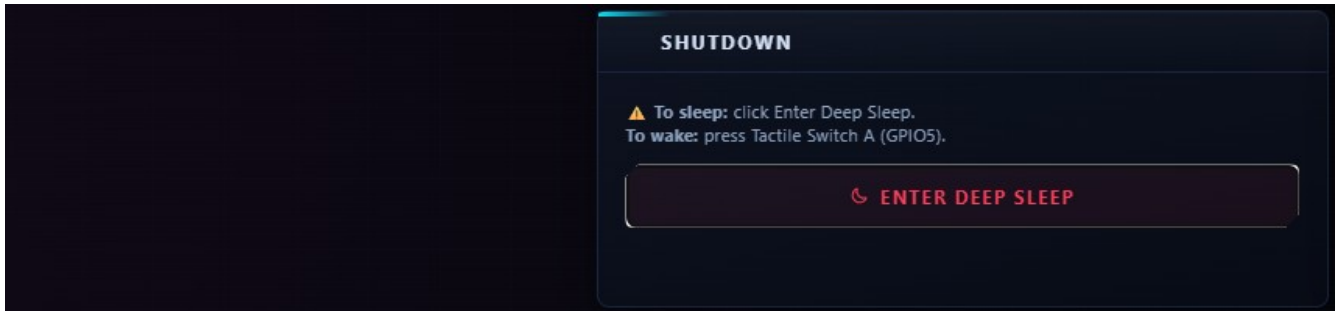
**PS5 button** → Loads Dual Sense factory ADC values (IDLE=1149, PRESSED=555)

#### Maintenance

**Reset Calibration** → Reverts to factory defaults Ps4 factory Presets

**Clear NVS + Force Recal** → Wipes ALL settings (tap twice to confirm)

## zz Section 4: Shutdown



Power management for battery-saving on your controller.

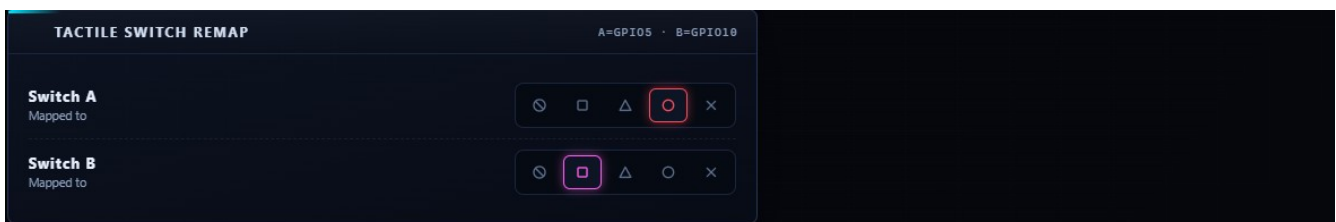
**Enter Deep Sleep** → ESP32 sleeps, draws ~microamps

**Wake up** → Press Tactile Switch A (GPIO5)

**Auto-OFF** after 5 min inactivity (when Rapid fire or Burst mode is ON) other functions still works normal only affect rapid fire or burst modes.

***Deep sleep You have to click the sleep button always at the end of each gaming section to save battery is mandatory if you don't want the esp32 drain your controller battery.***

## 🎮 Section 5: Tactile Switch Remap

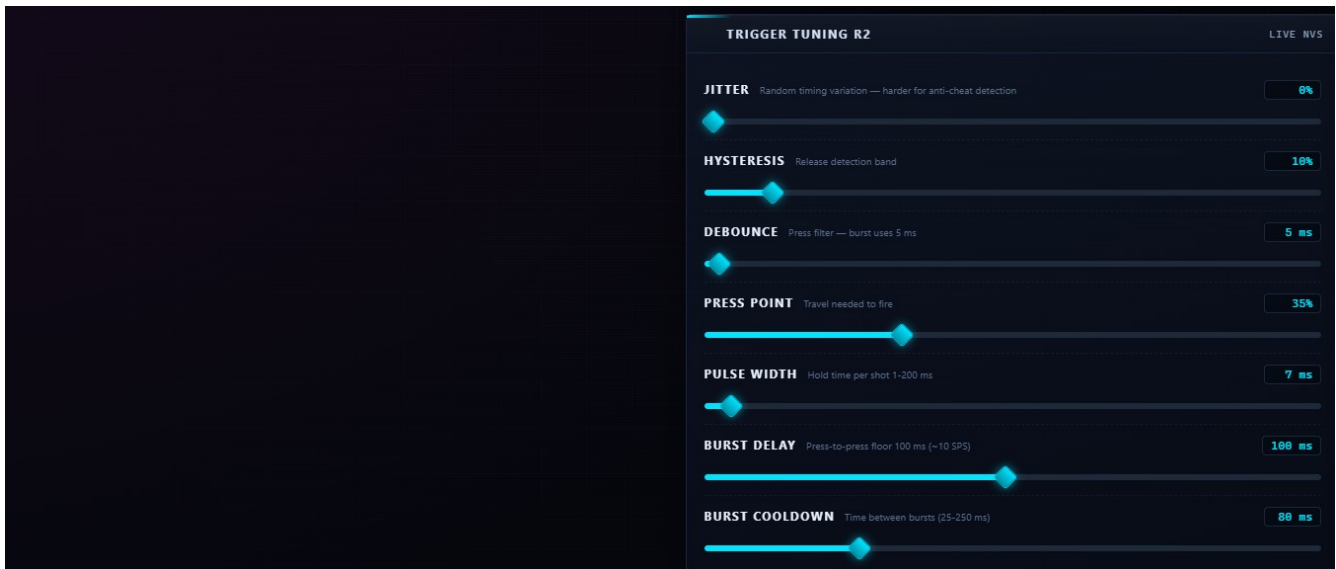


Maps the two physical tactile switches to any face button.

**Switch A-** GPIO5 Map options: None / Square / Triangle / Circle / X

**Switch B-** GPIO10 Map options: None / Square / Triangle / Circle / X

## 🔧 Section 6: Trigger Tuning (R2)



Fine-tune the rapid-fire behavior. All changes save to NVS automatically.

**Jitter- 0–100%** Random timing variation — defeats anti-cheat pattern detection, use around 20%-37% recommended on online play.

**Press Point- 5–100%** How far you pull R2 before it fires. You can change it to the value you want but default works on both controller fine if your ps5 controller is noisy you can increase this.

*This other sliders don't touch them if you have not glue what your doing left them along I save the defaults for both controller that I tested on real online game play and works fine on both controller.*

**Hysteresis- 0–100%** Release detection band — prevents chatter at threshold.

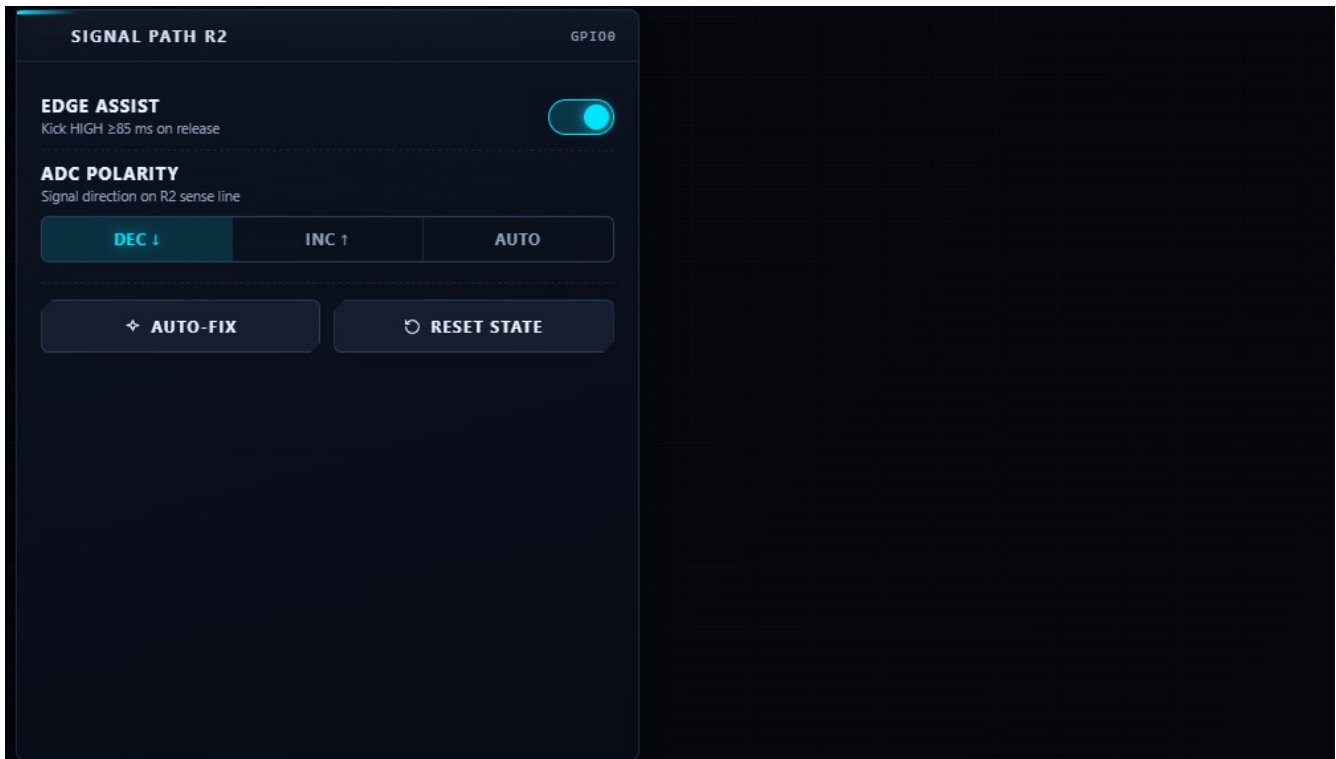
**Debounce- 0–500 ms** Press filter — ignores bouncy signals.

**Pulse Width- 1–200 ms** How long each "shot" press lasts.

**Burst Delay- 5–200 ms** Time between shots in burst mode.

**Burst Cooldown- 25–250 ms** Lockout after a burst ends.

## Section 7: Signal Path (R2)



Low-level electrical settings.

*Don't touch any setting here defaults are set for both controllers.*

### Edge Assist (Pull-up Kick)

**ON (default):** On release, the ESP32 kicks the R2 line HIGH for ~85 ms

This guarantees the game sees a clean release edge

Essential for games with strict input detection (BO2, BOCW etc.)

### ADC Polarity

**DEC ↓ (Decreasing)** — PS4/PS5 default: trigger pull = ADC goes DOWN.

**INC ↑ (Increasing)** — Rare, for custom triggers.

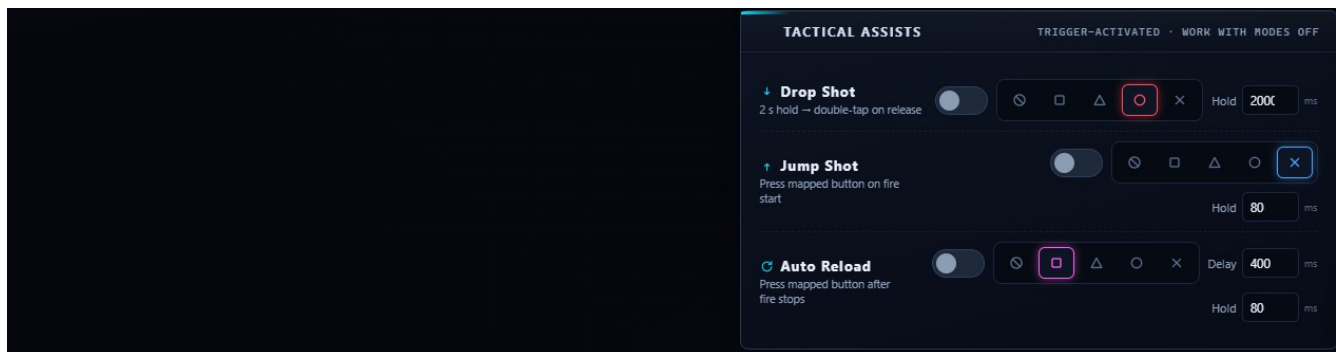
**AUTO** — Detects automatically from calibration.

### Helper Buttons

**Auto-Fix** → Applies safe defaults (debounce, hysteresis, press point, Edge Assist ON).

**Reset State** → Clears firing state without changing settings.

## 🏆 Section 8: Tactical Assists



Three combat features that work even when rapid-fire is OFF.

### Drop Shot 🪂

**On fire start** → holds mapped button for 2 seconds (prone).

**On R2 release** → double-taps the button to stand back up.

**Default map: Circle**

### Jump Shot 🐘

**On fire start** → spams the mapped button while R2 is held.

**Default map: X**

Hold duration: 80 ms per tap, 180 ms between taps.

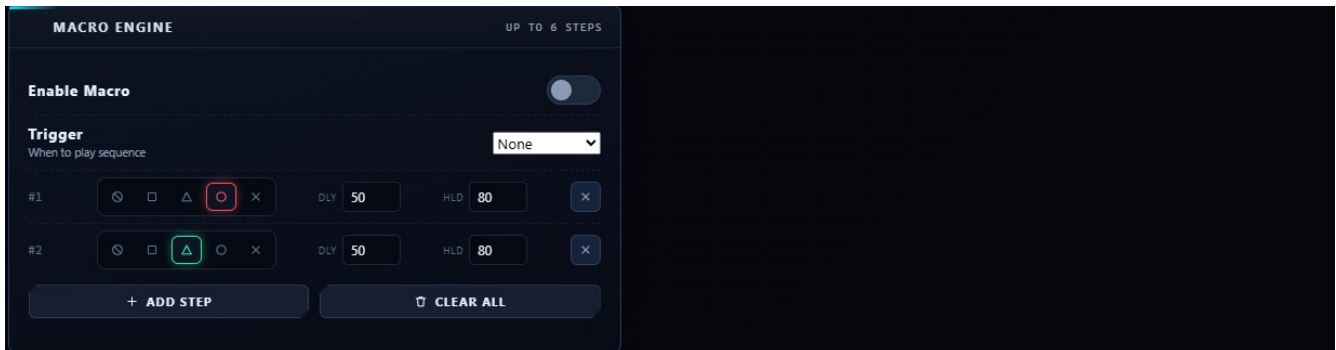
### Auto Reload 🔄

**After firing stops** → waits a delay, then presses mapped button.

**Default map: Square**

Configurable delay (50–3000 ms) and hold (20–500 ms). *Is milliseconds so max is 3000 ms equals to 3 seconds.*

## Section 9: Macro Engine



Record and play up to 6-step button sequences.

### Configuration

**Enable Macro** → Turns the engine on

**Trigger** → When to play:

**None** — Disabled

**Switch A** — Play when tactile A is pressed

**Switch B** — Play when tactile B is pressed

**Fire Start** — Play when R2 starts firing

### Steps

Each step has:

**Button** — Which face button to press (**Square/Triangle/Circle/X**).

**DLY** — Delay before next step (0–2000 ms)

**HLD** — How long to hold the button (20–500 ms)

**Use + Add Step / Clear All to manage the sequence.**

## Section 10: Presets



Save and load 4 full configuration slots.

### Each preset stores:

SPS, burst count, pulse width, delays

All tuning sliders, Tactical assist settings, Macro configuration, Tactile switch mappings.

### Default slots: FPS / Fighting / Sniper / Custom

Rename slots by editing the text field, then Save or Load.

## 6. Technical Notes

Storage: All settings saved to ESP32 NVS (non-volatile) — survives power cycles.

Auto-save delay: Settings save 3 seconds after last change (prevents flash wear).

WiFi: Soft-AP mode, channel 6, 8.5 dBm TX power, max 4 clients.

Web Socket: Live updates every 300 ms (idle) / 800 ms (firing).

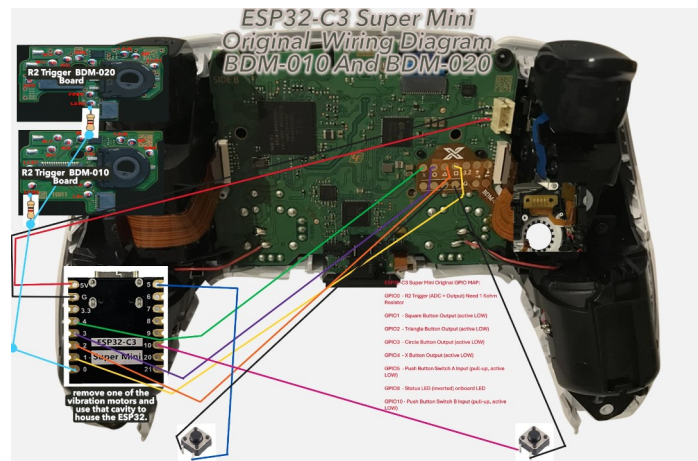
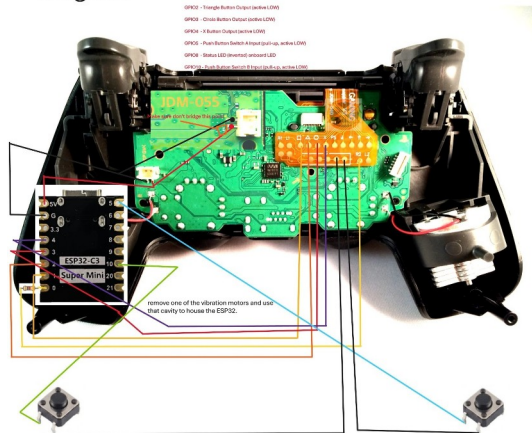
CPU: Single-core RISC-V @ 160 MHz — keep Web Socket load low.

Safety: Auto-disables mode after 5 min inactivity only rapid fire and burst mode; click deep sleep button after you done playing to save your battery is mandatory.

## 🚀 7. Quick Start Checklist

- ☐ Wire ESP32 to controller (R2 sense + 4 face buttons + 2 tactile switches)
- ☐ Add 1 kΩ series resistor on R2 sense line (protects ESP32 ADC)
- ☐ Power on → connect to RapidFireMod\_v0.9.9 WiFi
- ☐ Open <http://192.168.4.1>
- ☐ Run 3-step calibration
- ☐ Check the Live Monitor — pull R2, verify meter moves
- ☐ Set mode to Continuous or Burst
- ☐ Tune SPS / Jitter to taste
- ☐ Enable Tactical Assists if desired
- ☐ Save to a Preset slot for quick swapping

The Photos by The Controller DBM-020 Board Wiring to the same as the DBM-020 Board Wiring to the Battery Connector and the Battery Connector to the USB Cable  
ESP32-C3 Super Mini Original Wiring Diagram



Enjoy your mod! For the latest version and support, Check post or GitHub — made by Vegeta1.

