# KARL'S ULTIMATE DIY DYNO

Functional Specification Document (FSD) V3.58

Brake & Inertia Modes | EGT Monitoring | Full Run Storage | GPIO Trigger | Gear Ratio Tracking

| Document Field | Value |
|---|---|
| Date | May 06, 2026 |
| Firmware Version | 3.58 |
| Build Date | 06.05.26 |
| Document Version | 3.58 |
| Improved By | AI-Assisted Development |

---

## 1. Project Overview

### 1.1 Purpose

The Karl's Ultimate DIY Dyno is a low-cost, ESP32/ESP32-S3-based dynamometer system for measuring engine performance (RPM, horsepower, torque) on motorcycles, scooters, small engines, or similar applications. This document defines the functional requirements for Firmware V3.5.

### 1.2 Supported Dynamometer Modes

| Mode | Description |
|---|---|
| Brake Mode | Uses load cell force on a lever arm + drum RPM to calculate torque and power |
| Inertia Mode | Uses drum acceleration (inertia of flywheel/roller) to calculate power |

### 1.3 RPM Sources

| Source | Description |
|---|---|
| Spark | Engine RPM from ignition trigger (primary source) |
| Brake | Drum RPM × brake-to-engine ratio |
| Drum | Direct drum RPM display |
| CVT | Virtual gear ratio for continuously variable transmissions |

**1.4 Key Features (V3.5)**

- Self-contained WiFi access point or client mode
- Real-time web dashboard with analog-style gauges
- Full run data storage and retrieval (CSV format)
- Historical run comparison with overlay charts
- Live trend monitoring (RPM, drum RPM, HP, Torque)
- PDF report generation with embedded graphs
- EGT (Exhaust Gas Temperature) monitoring via MCP9600
- OTA firmware and filesystem updates
- GPIO trigger button support (NEW in V3.5)
- Automatic gear ratio tracking (NEW in V3.5)
- Atmospheric (air density) correction (NEW in V3.5)
- ESP32 and ESP32-S3 board support (NEW in V3.5)

**1.5 Key Hardware**

| Component | Model | Purpose |
|---|---|---|
| Microcontroller | ESP32 or ESP32-S3 | WiFi + processing, dual-core |
| Load Cell Amplifier | Adafruit NAU7802 | 24-bit ADC for force measuremen |

| Component | Model | Purpose |
|---|---|---|
| Thermocouple Amplifier | MCP9600 (I2C, 0x67) | K-type thermocouple (EGT) |
| Barometric Sensor | BMP280 (I2C, 0x76/0x77) | Pressure & temperature for air den |
| Hall Sensors (2x) | - | Engine RPM + Drum RPM measure |
| Lever Arm | Adjustable (default 25 cm) | Torque calculation in brake mode |
| Drum/Roller | Configurable mass/diameter | Inertia calculation |
| External Button | Momentary switch | GPIO trigger for run start/stop |

**1.6 Hardware Connections / Pinout (Board Specific)**

ESP32 Dev Board

| Peripheral | GPIO Pin | $I^2C$ Address | Notes |
|---|---|---|---|
| NAU7802 (Load Cell) | SDA=21, SCL=22 | 0x2A | Gain 128, 320 SPS |
| MCP9600 (EGT) | SDA=21, SCL=22 | 0x67 | K-type thermocouple |
| BMP280 | SDA=21, SCL=22 | 0x76 (or 0x77) | Pressure/temp |
| Hall Sensor (Engine RPM) | GPIO 26 | - | INPUT_PULLUP, debounce 200µs |
| Hall Sensor (Drum RPM) | GPIO 25 | - | INPUT_PULLUP, debounce 30µs |

| Peripheral | GPIO Pin | I²C Address | Notes |
|---|---|---|---|
| GPIO Trigger Button | GPIO 18 (default) | - | INPUT_PULLUP, FALLING edge |

ESP32-S3 Board

| Peripheral | GPIO Pin | I²C Address | Notes |
|---|---|---|---|
| NAU7802 (Load Cell) | SDA=1, SCL=2 | 0x2A | Gain 128, 320 SPS |
| MCP9600 (EGT) | SDA=1, SCL=2 | 0x67 | K-type thermocouple |
| BMP280 | SDA=1, SCL=2 | 0x76 (or 0x77) | Pressure/temp |
| Hall Sensor (Engine RPM) | GPIO 4 | - | INPUT_PULLUP, debounce 200µs |
| Hall Sensor (Drum RPM) | GPIO 5 | - | INPUT_PULLUP, debounce 30µs |
| GPIO Trigger Button | GPIO 17 (default) | - | INPUT_PULLUP, FALLING edge |

Board Selection: #define IS_ESP32S3 1 in firmware before compilation

1.7 Required Arduino Libraries

| Library | Version | Purpose | Required |
|---|---|---|---|
| ESPAsyncWebServer | 1.2.3+ | Async web server + WebSocket | Yes |

| Library | Version | Purpose | Required |
|---|---|---|---|
| AsyncTCP | 1.1.1+ | Dependency for async server | Yes |
| Adafruit NAU7802 | Latest | 24-bit ADC driver | Yes |
| Adafruit MCP9600 | Latest | Thermocouple amplifier | Yes |
| Adafruit BMP280 | Latest | Pressure/temperature sensor | Yes |
| DNSServer | Built-in | Captive portal (AP mode) | Yes |
| Preferences | Built-in | NVS settings storage | Yes |
| LittleFS | Built-in | Filesystem for web files | Yes |
| Update | Built-in | OTA firmware updates | Yes |

---

**2. Functional Requirements**

**2.1 User Roles & Access**

- Single user (owner/operator)
- No authentication — open WiFi AP or client mode
- Captive portal redirects to dashboard in AP mode

**2.2 Main Dashboard Layout**

Three-Column Layout:

- Left Column: HP gauge + Drum RPM display (stacked vertically)
- Center Column: Large RPM gauge with EGT display inside
- Right Column: Torque gauge + Loadcell kg display (stacked vertically)

Live Displays:

- RPM gauge with colored zones (Green 0-7k, Yellow 7k-12k, Red 12k+)
- Digital RPM readout (center)

- HP gauge (0 to MAX_HP)
- Torque gauge (0 to MAX_TORQUE)
- Loadcell reading (kg) below torque gauge
- Drum RPM display below HP gauge
- EGT reading (°C) inside RPM gauge box
- Ambient temperature (°C) in bottom-right corner
- RPM source indicator (Spark/Brake/Drum/CVT) in bottom-left
- Gear ratio display (NEW - shows current engine-to-drum ratio)
- Air density / correction factor (NEW - shows atmospheric correction)

Status Indicators:

- Recording status (red pulse animation during active recording)
- Peak HP and Peak Torque with RPM values
- Connection status (green when connected)

### 2.3 GPIO Trigger Button (NEW in V3.5)

#### 2.3.1 Overview

External momentary push button connected between a configurable GPIO pin and GND. Pressing the button opens the run dialog (same as keyboard F12), allowing the user to start a manual run with customer/vehicle information.

#### 2.3.2 Technical Specifications

| Parameter | Value |
|---|---|
| Default pin (ESP32) | GPIO 18 |
| Default pin (ESP32-S3) | GPIO 17 |
| Interrupt type | FALLING edge |
| Debounce time | 500 ms |
| Pull-up | Internal INPUT_PULLUP |

| Parameter | Value |
| --- | --- |
| Action | Same as F12 (show run dialog) |

### 2.3.3 WebSocket Commands

| Command | Purpose |
| --- | --- |
| SET_GPIO_PIN:<pin> | Configure GPIO trigger pin (0-39, avoids used pins) |
| GET_GPIO_STATUS | Returns current pin and enabled status |
| TEST_GPIO_TRIGGER | Simulate button press for testing |

### 2.3.4 Pin Conflict Avoidance

The firmware prevents assigning pins already used for:

- HALL_PIN (engine RPM)
- DRUM_PIN (drum RPM)
- I2C_SDA / I2C_SCL

## 2.4 Gear Ratio Tracker (NEW in V3.5)

### 2.4.1 Overview

Automatically calculates and tracks the ratio between engine RPM and drum RPM in real-time, enabling accurate torque calculations without manual ratio entry.

### 2.4.2 Calculation Method

text

instant_ratio = engine_rpm / drum_rpm (when engine_rpm > 800 AND drum_rpm > 20)

### 2.4.3 Signal Processing

| Parameter | Value |
| --- | --- |
| History buffer | 10 samples |

| Parameter | Value |
|---|---|
| Valid range | 0.5 - 50.0 |
| Stability threshold | >98% for high confidence |
| Filter strength (high confidence) | 0.01 |
| Filter strength (low confidence) | 0.10 |
| Confidence decay | 0.002 per cycle when inactive |
| Timeout fallback | 5 seconds → last valid ratio |

**2.4.4 Confidence Levels**

| Confidence | Meaning | Action |
|---|---|---|
| > 80% | Very stable | Use tracked ratio, minimal filtering |
| 60-80% | Moderately stable | Use tracked ratio with caution |
| 30-60% | Low confidence | Blend with default ratio |
| < 30% | Unstable | Fallback to BRAKE_TO_ENGINE_RATIO |

**2.4.5 Commands**

- RESET_RATIO - Reset gear ratio tracking to default

**2.5 Atmospheric Correction (NEW in V3.5)**

**2.5.1 Overview**

Real-time air density calculation using BMP280 sensor data, with SAE-style correction applied to HP and torque readings.

**2.5.2 Air Density Formula**

text

air_density = pressure_pa / (287.05 × temperature_kelvin)

where:

- pressure_pa = atmospheric_pressure × 100

- temperature_kelvin = ambient_temp_c + 273.15

**2.5.3 Correction Factor**

text

correction_factor = STD_AIR_DENSITY (1.204) / air_density

clamped to range: 0.85 - 1.15

**2.5.4 Application**

Both HP and torque values are multiplied by the correction factor before display and recording.

**2.5.5 Fallback Behavior**

If BMP280 is not detected:

- Air density defaults to 1.204 kg/m$^3$
- No correction applied (factor = 1.0)
- System continues to operate normally

**2.6 Data Acquisition**

**2.6.1 Engine RPM Measurement**

- Hall effect sensor on ignition/crank trigger
- Period measurement (microsecond intervals)
- Median filtering (5-point) + exponential smoothing
- Formula: RPM = 60,000,000 / (interval_µs × pulses_per_revolution)
- Debounce: 200µs
- Timeout: 500ms (zero reading if no pulse)
- Max RPM: 50,000

**2.6.2 Drum RPM Measurement**

- Hall effect sensor on drum/roller
- Period measurement with 10 pulses per revolution default

- Median filtering (5-point) + exponential smoothing (α = 0.15)
- Debounce: 30µs
- Safety cap: 25,000 RPM (brake mode) / 12,000 RPM (inertia mode)

**2.6.3 Load Cell Measurement (NAU7802)**

- 24-bit ADC, Gain 128, 320 SPS
- Zero offset compensation after tare
- Calibration factor stored in Preferences
- Conversion: kg = (raw - zeroOffset) × CALIBRATION_FACTOR / 1000

**2.6.4 EGT Measurement (MCP9600)**

- K-type thermocouple at I2C address 0x67
- 5-point moving average filter
- Peak EGT tracking during runs
- Ambient compensation
- Filter coefficient: 3, ADC resolution: 14-bit
- Update rate: 15 Hz

**2.6.5 BMP280 Measurement**

- Pressure and temperature for air density correction
- Update rate: 2 Hz
- Sampling: X2 pressure, X16 temperature, X16 filter

**2.7 Power Calculations**

**2.7.1 Brake Mode**

text

Force (N) = kg × 9.81

Brake Torque (Nm) = Force × lever_arm_meters

With gear ratio tracking:

Engine Torque (Nm) = Brake Torque / gear_ratio

Engine Torque = Engine Torque / drivetrain_efficiency

HP = (Engine Torque × engine_rpm) / 7121

**2.7.2 Inertia Mode**

Inertia Calculation:

| Type | Formula |
|---|---|
| Solid cylinder | $I = 0.5 \times mass \times radius^2$ |
| Hollow cylinder | $I = 0.5 \times mass \times (radius^2 + inner\_radius^2)$ |
| Custom | I = user_defined |

Power Calculation:

text

Angular acceleration (rad/s$^2$) = $\Delta\omega / \Delta t$

Torque (Nm) = I × angular_acceleration

HP = (Torque × drum_rpm) / 7121

2.8 Signal Processing

| Parameter | Filter Type | Value |
|---|---|---|
| RPM | Median (5-point) + Exponential | α = 0.50 |
| Drum RPM | Median (5-point) + Exponential | α = 0.15 |
| HP | Moving average | 9 samples |
| Torque | Moving average | 9 samples |
| EGT | Moving average | 5 samples |
| Gear Ratio | Adaptive exponential | 0.01-0.10 based on confiden |

| Parameter | Filter Type | Value |
|---|---|---|
| Angular Acceleration | Exponential | α = 0.30 |

### 2.9 Recording System

#### 2.9.1 Manual Mode

- Start via "Manual Run" button, F12 key, or GPIO trigger button
- Customer/vehicle/comments input modal
- Manual stop or F12/GPIO button to finish
- Maximum run duration: 45 seconds (safety timeout)

#### 2.9.2 Auto-Run Mode

- Start RPM threshold + hysteresis (default 180 RPM)
- End RPM threshold - hysteresis
- Automatic start when RPM crosses threshold
- Automatic stop when RPM reaches end threshold
- Configurable via modal (F8)

#### 2.9.3 Recording Data Structure

cpp

```
struct RunPoint {
 float rpm;     // Display RPM
 float hp;      // Horsepower (corrected)
 float torque;  // Torque (Nm, corrected)
};
```

- Maximum points: 1500 per run
- Sample rate: 10 Hz (100ms interval)

### 2.10 File System Management

#### 2.10.1 Run Storage (LittleFS)

text

```
/run_<timestamp>.csv    # Run data (RPM,HP,Torque)
```

/runs.idx # Index file (metadata)

### 2.10.2 CSV Format

csv

RPM,HP,Torque

1200,5.23,22.8

1250,5.67,23.1

...

### 2.10.3 Index Format

text

<timestamp>,<filename>,<customer>,<vehicle>,<peakHP>,<peakTorque>,<peakEGT>

### 2.10.4 Run Management

- Maximum saved runs: 10 (FIFO)
- Delete individual runs
- Delete multiple selected runs
- Compare up to 6 runs simultaneously

## 2.11 WebSocket Protocol

### 2.11.1 Message Types

| Type | Direction | Purpose |
|---|---|---|
| settings | Server → Client | Configuration values, version info |
| data | Server → Client | Live RPM, HP, Torque, drum RPM, EGT, gear ratio |
| runs | Server → Client | List of saved runs |
| run_data | Server → Client | Full run curve data |
| run_complete | Server → Client | Run completion with peaks |

| Type | Direction | Purpose |
|---|---|---|
| info | Server → Client | Toast messages |
| sensor_status | Server → Client | NAU7802/MCP9600/BMP280 status |
| gpio_status | Server → Client | GPIO trigger pin configuration |

**2.11.2 Client Commands**

| Command | Purpose |
|---|---|
| TARE | Zero load cell |
| CALIBRATE:<kg> | Calibrate with known weight |
| GET_SETTINGS | Request configuration |
| GET_RUNS | Request run list |
| GET_RUN_DATA:<filename> | Request full run data |
| GET_SENSOR_STATUS | Request sensor health status |
| DELETE_RUN:<index> | Delete run by index |
| AUTO_RUN:<start>:<end> | Start auto-run |
| MANUAL_START | Start manual recording |
| MANUAL_STOP | Stop manual recording |
| CANCEL_AUTO_RUN | Cancel auto-run |

| Command | Purpose |
|---|---|
| SET_SINGLE:<key>=<value> | Update single setting |
| SET_CUSTOMER:<name>\|<unit>\|<comments> | Set run metadata |
| WIFI_CLIENT:<ssid>:<pass> | Configure WiFi client |
| RESET_DEFAULT | Reset to AP mode |
| RESET_ALL_SETTINGS | Factory reset |
| RESET_RATIO | Reset gear ratio tracking |
| SET_GPIO_PIN:<pin> | Configure GPIO trigger pin |
| GET_GPIO_STATUS | Get GPIO trigger status |
| TEST_GPIO_TRIGGER | Simulate GPIO trigger |
| PAGE_MAIN, PAGE_SETTINGS, PAGE_TRENDS | Page state |

**2.12 Configuration Parameters**

| Parameter | Range | Default | Description |
|---|---|---|---|
| MAX_RPM | 5000-50000 | 16000 | Gauge maximum RPM |
| MAX_HP | 5-100 | 25 | Gauge maximum HP |

| Parameter | Range | Default | Description |
|---|---|---|---|
| MAX_TORQUE | 5-100 | 25 | Gauge maximum torque (Nm) |
| LEVER_ARM_METERS | 0.05-1.00 | 0.25 | Lever arm length (m) |
| CALIBRATION_FACTOR | 0.0001-0.1 | 0.00678 | Load cell scaling |
| PULSES_PER_REVOLUTION | 0.5-10 | 2.0 | Engine pulses per revolution |
| PULSES_PER_REV_DRUM | 0.5-20 | 10.0 | Drum pulses per revolution |
| BRAKE_TO_ENGINE_RATIO | 0.1-50 | 10.0 | Brake RPM to engine RPM ratio (fallback) |
| RPM_FILTER_ALPHA | 0.1-0.9 | 0.50 | Exponential smoothing factor |
| AUTO_HYSTERESIS | 50-500 | 180 | Auto-run hysteresis (RPM) |
| DYNO_MODE | brake/inertia | brake | Operating mode |

| Parameter | Range | Default | Description |
|---|---|---|---|
| DRUM_MASS | 1-300 kg | 10.0 | Drum mass |
| DRUM_DIAM_METERS | 0.1-3.0 m | 0.3 | Drum diameter |
| DRUM_INERTIA_TYPE | solid/hollow/custom | solid | Inertia calculation type |
| DRUM_WALL_THICKNESS_CM | 0.5-20 cm | 2.0 | Hollow drum wall thickness |
| DRUM_INERTIA_CUSTOM | 0.01-100 | 0.5 | Custom inertia ($kg \cdot m^2$) |
| RPM_SOURCE | spark/brake/drum/cvt | spark | RPM display source |
| VIRTUAL_GEAR_RATIO | 0.1-200 | 1.0 | CVT virtual ratio |
| DRIVETRAIN_EFFICIENCY | 70-99% | 93% | Drivetrain loss compensation |
| GPIO_TRIGGER_PIN | 0-39 | 18 (ESP32) / 17 (S3) | External trigger button pin |

**2.13 Web Pages & API Endpoints**

| Page/Endpoint | URL | Purpose |
|---|---|---|
| Dashboard | / | Main dyno display with gauges |
| Settings | /settings | Configuration and calibration |
| Trends | /trend.html | Live trends + run history |
| Update | /update | OTA firmware update |
| Filesystem Update | /update_fs | OTA filesystem update |
| API: Sensors | /api/sensors | JSON sensor status |
| API: Version | /version | Firmware version string |

**2.14 Trend Page Features**

- Live Trend Charts: RPM, Drum RPM, HP, Torque (4 separate charts)
- Auto-scrolling: Last 150 samples displayed
- Run History: List of saved runs with peaks
- Run Comparison: Overlay multiple runs on same chart
- Run Management: Delete individual or multiple runs
- Run Details: Customer, vehicle, HP, torque, EGT

**2.15 PDF Report Generation**

- jsPDF library embedded in browser
- Includes:
    - Customer/vehicle information
    - Peak HP and torque values
    - Peak and final EGT
    - Ambient temperature and air density correction
    - Graph with HP and torque curves

- Timestamp and version info

- File naming: karls_dyno_<customer>_<timestamp>.pdf

**2.16 Diagnostics & Status Monitoring**

**2.16.1 Sensor Status JSON**

json

```
{
  "nau7802": true,
  "mcp9600": true,
  "bmp280": false
}
```

**2.16.2 Serial Console Output**

Every 65 seconds, the firmware prints:

text

Heap: <bytes> | Runs: <count> | EGT: <temp>°C | GPIO Pin: <pin> | Sensors: NAU=<0/1> MCP=<0/1> BMP=<0/1>

---

**3. System Architecture**

**3.1 Task Structure (FreeRTOS)**

| Task | Core | Stack | Priority | Purpose |
|---|---|---|---|---|
| sensorTask | Core 0 | 8192 | 2 | Data acquisition, calculations |
| calibrationTask | Core 1 | 4096 | 1 | Weight calibration process |
| Loop (main) | Core 0 | - | - | DNS, WebSocket cleanup |

**3.2 Update Rates**

| Component | Rate |
|---|---|
| Sensor reading | 50 Hz (20ms) |
| WebSocket broadcast | 20 Hz (50ms throttled) |
| EGT reading | 15 Hz (66ms) |
| Gear ratio calculation | 10 Hz (100ms) |
| BMP280 reading | 2 Hz (500ms) |
| Run data recording | 10 Hz (100ms) |
| Serial status print | ~0.015 Hz (65 sec) |

**3.3 Data Flow**

text

Hall Sensors → ISR → Period Measurement → Median Filter → RPM

Load Cell → NAU7802 → Zero Offset → Calibration Factor → kg

BMP280 → Pressure/Temp → Air Density → Correction Factor

MCP9600 → EGT → Moving Average → Peak Tracking

Force (N) = kg × 9.81

Torque = Force × lever_arm (Brake) OR I × α (Inertia)

Torque = Torque × correction_factor

HP = (Torque × RPM) / 7121 × correction_factor

WebSocket → Client Dashboard

---

**4. Safety Features**

| Feature | Limit |
|---|---|
| Max drum RPM (brake) | 25,000 RPM |
| Max drum RPM (inertia) | 12,000 RPM |
| Recording timeout | 45 seconds |
| RPM stale timeout (engine) | 500ms |
| RPM stale timeout (drum) | 1000ms |
| HP/Torque cap | 2× MAX value (display) |
| Debounce (engine) | 200µs |
| Debounce (drum) | 30µs |
| GPIO debounce | 500ms |
| Correction factor clamp | 0.85 - 1.15 |

---

**5. Non-Functional Requirements**

| Requirement | Specification |
|---|---|
| Update rate | ~50 Hz live data |
| Calibration duration | 5 seconds |
| Recording timeout | 45 seconds |
| Internet requirement | None (local WiFi only) |

| Requirement | Specification |
|---|---|
| Accuracy | ±1-4% after calibration |
| Maximum runs stored | 10 (automatic FIFO) |
| Maximum points per run | 1500 |
| WiFi mode | AP (default) or STA |
| AP SSID | karlsdyno |
| AP Password | karlsdyno |
| mDNS | karlsdyno.local |
| Startup time | < 10 seconds |

---

**6. Assumptions & Constraints**

- Load cell range: ~1-50 kg typical
- NAU7802 gain: 128, channel A
- MCP9600 fixed address: 0x67
- EGT thermocouple type: K-type
- BMP280 addresses: 0x76 (primary) or 0x77 (secondary)
- No temperature compensation for load cell
- Dashboard remains responsive during calibration
- Drum RPM sensor: 10 pulses/revolution typical
- GPIO trigger button: Normally open, momentary switch

---

**7. Revision History**

| Version | Date | Author | Changes |
|---|---|---|---|
| 1.0 | Feb 17, 2026 | Karl | Initial FSD |
| 3.1 | Mar 28, 2026 | Karl | Added drum RPM, trend page, run comparison, dual RPM sources, OTA updates |
| 3.5 | Apr 14, 2026 | Karl + AI | Added GPIO trigger support, gear ratio tracking, atmospheric correction, ESP32-S3 support, updated debounce values, new WebSocket commands, API endpoints, sensor status monitoring |

---

**8. AI Contribution Acknowledgement**

This firmware and documentation were improved with assistance from AI including:

- GPIO trigger button implementation and documentation
- Gear ratio tracking algorithm design
- Atmospheric correction formulas and integration
- ESP32 vs ESP32-S3 pin configuration management
- WebSocket protocol extensions
- Code optimization suggestions
- FSD update and synchronization with actual firmware