Hardware Approach Document

EV Battery Intelligence Challenge - Hardware Layer

1. System Context & Hardware Role

```
Challenge: Unlock "black-box" BMS data → Indigenous RISC-V analytics → Transparent RUL
Target: Simple E-Pluto 7G 2W Test Rig (16S6P, 60V, 1.2kWh Li-ion)
BMS: DALY 16S60V60A (CAN 250kbps, DB9 test point)
Edge: VSDSquadron ULTRA + THEJAS32 RISC-V (100MHz)
```

Hardware Mission: Passive CAN observer → Raw signal decode → Feature extraction → MQTT telemetry

2. Complete Hardware Stack

```
┌──────────────────────────────────────────────────────────┐
│ 2. Scooter Rig: Simple E-Pluto 7G Test Rig               │
│  ├── Battery: 16S6P Li-ion (60V, 1.2kWh)                 │
│  ├── BMS: DALY 16S60V60A                                 │
│  │   ├── Signals: V_pack, I_pack, T_NTCs, V_cell[1-16]   │
│  │   └── CAN: 250kbps, DB9 Test Point (CANH/CANL/GND)    │
│  └── LV Power: 12V Auxiliary Bus                         │
└──────────────────────────────────────────────────────────┘
                            ↓
┌──────────────────────────────────────────────────────────┐
│ 3. CAN Physical Layer                                    │
│  ├── Connector: DB9 → Screw Terminal Adapter             │
│  ├── Transceiver: SN65HVD230 (3.3V, 2.5kV Isolation)     │
│  └── Termination: 120Ω (matches BMS bus impedance)       │
└──────────────────────────────────────────────────────────┘
                            ↓ SPI (10MHz)
┌──────────────────────────────────────────────────────────┐
│ 4. CAN Controller                                        │
│  ├── Chip: MCP2515 (SPI Slave, 16MHz Crystal)            │
```

```
|   ├── Mode: Listen-Only (CANCTRL=0x14, No Tx/No ACK)          |
|   ├── Buffer: 32 Rx Buffers (Circular)                        |
|   └── Interrupt: Pin 1 → THEJAS32 GPIO17                      |
└──────────────────────────────────────────────────────────────┘
                              ↓ GPIO/SPI Pins
┌──────────────────────────────────────────────────────────────┐
| 5. Edge Compute: VSDSquadron ULTRA                            |
|   ├── Processor: THEJAS32 RISC-V 64-bit (100MHz)              |
|   ├── Memory: 512KB SRAM + 16MB Flash                         |
|   ├── Peripherals: SPI2, GPIO17-20, UART2, 32-bit Timer       |
|   └── Power: 12V→5V DC-DC → 3.3V LDO (500mA margin)           |
└──────────────────────────────────────────────────────────────┘
                              ↓ UART2 (115200)
┌──────────────────────────────────────────────────────────────┐
| 6. Telemetry Output                                           |
|   ├── MQTT Client: Lightweight JSON Serializer                |
|   ├── Topics: ev/scooter1/telemetry, ev/scooter1/health       |
|   └── Fallback: UART CSV for PC dashboard                      |
└──────────────────────────────────────────────────────────────┘
```

3. Pinout & Connections

```
Scooter Harness DB9 → CAN Adapter:
Pin 2 → CANL (Orange)
Pin 7 → CANH (Yellow)
Pin 3 → GND (Black)
Pin 9 → 12V LV (Red)

VSDSquadron GPIO Mapping:
GPIO17 ← MCP2515 INT (Rx Ready)
GPIO18 → MCP2515 CS
```

```
GPIO19 → MCP2515 SCK
GPIO20 → MCP2515 MOSI
GPIO21 ← MCP2515 MISO

Status LEDs (GPIO22-24):
GPIO22 → Green (Healthy)
GPIO23 → Yellow (Warning)
GPIO24 → Red (Critical)
```

4. Power Architecture

```
Scooter LV Bus (12V, 79W total load capacity)
    ↓ [LM2596 DC-DC: 12V→5V, 3A]
USB-C Input (VSDSquadron: 5V/2A)
    ↓ [Local LDO: 5V→3.3V, 1A]
MCP2515 (3.3V, 50mA) + SN65HVD230 (3.3V, 20mA)
    ↓ THEJAS32 Core (3.3V, 300mA peak)
```

Total Consumption: ~400mA @ 3.3V = 1.3W (fits LV budget)

5. Firmware-Hardware Interface

MCP2515 Register Configuration

```
CANCTRL = 0x14        // Listen-Only Mode
CANINTE = 0x03        // Rx0 + Error Interrupt
CNF1 = 0x03           // 250kbps @ 16MHz
CNF2 = 0xB1           // Sample Point 75%
CNF3 = 0x02           // Normal/Sync Jump
```

Interrupt Flow

```
500ms Timer IRQ → Poll MCP2515 → Decode Frame →
Timestamp → Feature Calc → MQTT Publish → LED Update
```

6. BOM (Bill of Materials)

Component	Qty	Part Number	Source	Cost
VSDSquadron ULTRA	1	VSD Kit	Hackathon	Free
MCP2515 Module	1	Waveshare	Amazon	₹150
SN65HVD230 Module	1	Generic	Local	₹100
DB9 Breakout	1	Generic	Local	₹50
Status LEDs + Resistors	3	Generic	Local	₹20
DC-DC 12→5V	1	LM2596	Local	₹80
Total				₹400

7. PCB / Assembly Notes

```
Form Factor: Perfboard (10x10cm) - Hackathon Ready
Connectors: DB9 Female + USB-C + UART Header
Mounting: DIN Rail Clips (optional)
Enclosure: IP54 Plastic Box (future)
Debug: SWD Header for THEJAS32
```

8. Test & Validation Plan

```
Phase 1: CAN Signal Verification
├── Multimeter: V/I/T static check
├── Oscilloscope: CANH/CANL waveform
└── Logic Analyzer: MCP2515 SPI traffic

Phase 2: Edge Processing
├── Serial Monitor: Raw decode verification
├── LED Test: Health zones trigger
└── MQTT Inspector: Payload validation

Phase 3: Scooter Integration
├── Live discharge: 5km simulated ride
├── Charge cycle: Full SOC sweep
└── Stress test: High current acceleration
```

9. Edge-to-Cloud Data Contract

```
Every 500ms → MQTT Payload (180 bytes avg):
├── Raw: V,I,T,ΔV,SOC_ref (40 bytes)
├── Features: C_rate,Energy,Cycles,R_est (60 bytes)
└── Scores: HealthIndex,StressScore (20 bytes)

Full 16-cell snapshot: Every 30s or on ΔV>20mV event
```