**IoT Node with Voltage Management System  
Content Type: Project Report**

**INDEX**

| **Section** | **Title** | **Page No.** |
| --- | --- | --- |
| **1** | **Introduction** | **1** |
| **2** | **List of Material** | **2** |
| **3** | **Power Management** | **3** |
| **4** | **Communication Interfaces** | **4** |
| **5** | **Sensors – Environmental** | **5** |
| **6** | **Sensors – Motion & Position** | **6** |
| **7** | **User Interface Components** | **7** |
| **8** | **Display & Output** | **8** |
| **9** | **Power Control & Actuators** | **9** |
| **10** | **Passive Components – Resistors** | **10** |
| **11** | **Passive Components – Capacitors** | **11** |
| **12** | **Inductors & Crystals** | **12** |
| **13** | **Connectors & Headers** | **13** |
| **14** | **Essential Core Software** | **14** |
| **15** | **Flowchart Explanation** | **15** |
| **16** | **Project Utility and Application** | **16** |
| **17** | **Step-by-Step Design Process** | **17** |
| **18** | **Limitations vs. Benefits** | **18** |
| **19** | **Why Schematic, PCB & 3D View are Crucial** | **19** |
| **20** | **Cost Comparison** | **20** |
| **21** | **Conclusion** | **21** |

**1. Introduction**

This project involves the design and development of a dedicated **IoT Node** featuring a core microcontroller unit and a robust, integrated voltage management system. The primary objective is to create a reliable, standalone PCB that can serve as the central processing unit for IoT applications, capable of being powered by a variety of sources (such as a 12V DC input or USB) and providing stable, clean power to the microcontroller, sensors, and communication modules.

**2.List of Material**

1. **Microcontroller & Core Processing**

| **Component Type** | | **Part Number** | **Package** | **Quantity** | **Value/Specs** | **Purpose** |
| --- | --- | --- | --- | --- | --- | --- |
| Main MCU | ESP32-WROOM-32 | QFN48 | 1 | Wi-Fi + BT, 4MB Flash | Main processor |
| Backup MCU | STM32F103C8T6 | LQFP48 | 1 | ARM Cortex-M3 | Alternative processor |
| RTC | DS3231 | SOIC-16 | 1 | I2C, Battery backup | Real-time clock |
| EEPROM | 24C32 | SOIC-8 | 1 | 32Kbit I2C | Data storage |

**2. Power Managemen**

| **Component Type** | **Part Number** | **Package** | **Quantity** | **Value/Specs** | **Purpose** |
| --- | --- | --- | --- | --- | --- |
| Buck Converter | MP1584EN | SOP-8 | 1 | 12V→5V, 3A | Main power step-down |
| LDO Regulator | AMS1117-3.3 | SOT-223 | 2 | 5V→3.3V, 1A | 3.3V regulation |
| LDO Regulator | MCP1700-3302 | SOT-23 | 1 | 3.3V, 250mA | Low-power rail |
| PTC Fuse | 1812L | 1812 | 2 | 1A, 6V | Overcurrent protection |
| TVS Diode | SMAJ12A | SMA | 2 | 12V protection | Voltage spikes |
| Schottky Diode | SS34 | SMA | 2 | 3A, 40V | Reverse polarity |

**3. Communication Interfaces**

| **Component Type** | **Part Number** | **Package** | **Quantity** | **Value/Specs** | **Purpose** |
| --- | --- | --- | --- | --- | --- |
| USB-UART | CH340C | SOP-16 | 1 | USB 2.0 | Programming interface |
| Ethernet | W5500 | LQFP-48 | 1 | SPI Ethernet | Wired networking |
| RS485 | MAX485 | SOIC-8 | 1 | RS485 transceiver | Industrial comms |
| CAN Bus | MCP2551 | DIP-8 | 1 | CAN transceiver | Automotive comms |
| LoRa | SX1276 | LoRa-16 | 1 | 433/868/915MHz | Long-range wireless |

**4. Sensors – Environmental**

| **Component Type** | **Part Number** | **Package** | **Quantity** | **Value/Specs** | **Purpose** |
| --- | --- | --- | --- | --- | --- |
| Temp/Humidity | DHT22 | Through-hole | 1 | -40~80°C, 0-100% | Environmental sensing |
| Temp/Humidity/Pressure | BME280 | LGA-8 | 1 | I2C/SPI | All-in-one environmental |
| Air Quality | MQ-135 | Module | 1 | CO2, NH3, smoke | Air quality detection |
| Gas Sensor | CCS811 | LGA-10 | 1 | TVOC, eCO2 | Indoor air quality |
| Light Sensor | BH1750 | SOP-5 | 1 | 0-65535 lux | Ambient light |

**5. Sensors - Motion & Position**

| **Component Type** | **Part Number** | **Package** | **Quantity** | **Value/Specs** | **Purpose** |
| --- | --- | --- | --- | --- | --- |
| Accelerometer/Gyro | MPU-6050 | QFN-24 | 1 | 6-axis IMU | Motion tracking |
| Magnetometer | HMC5883L | DFN-16 | 1 | 3-axis compass | Direction sensing |
| GPS | NEO-6M | Module | 1 | GPS receiver | Location tracking |
| PIR Motion | HC-SR501 | Module | 1 | Human detection | Motion sensing |
| Ultrasonic | HC-SR04 | Module | 1 | 2cm-400cm | Distance measurement |

**6. User Interface Components**

| **Component Type** | **Part Number** | **Package** | **Quantity** | **Value/Specs** | **Purpose** |
| --- | --- | --- | --- | --- | --- |
| Tactile Switch | TS-1187A | 6x6mm | 4 | SPST, through-hole | User input |
| Rotary Encoder | EC11 | Through-hole | 1 | 15 pulses/360° | Rotary input |
| Touch Sensor | TTP223 | SOT-23-6 | 2 | Capacitive touch | Touch input |
| Potentiometer | 3362P | Through-hole | 2 | 10K linear | Analog input |
| Joystick | PS2-Joy | Module | 1 | 2-axis analog | Direction control |

**7. Display & Output**

| **Component Type** | **Part Number** | **Package** | **Quantity** | **Value/Specs** | **Purpose** |
| --- | --- | --- | --- | --- | --- |
| OLED Display | SSD1306 | Module | 1 | 128x64, I2C | Visual output |
| Character LCD | 1602A | Module | 1 | 16x2, I2C | Text display |
| RGB LED | WS2812B | 5050 | 4 | Addressable RGB | Status indication |
| LED | Various | 0805 | 10 | Red/Green/Blue | Status indicators |
| Buzzer | EM-27314 | Through-hole | 1 | 5V, passive | Audio feedback |

**8. Power Control & Actuators**

| **Component Type** | **Part Number** | **Package** | **Quantity** | **Value/Specs** | **Purpose** |
| --- | --- | --- | --- | --- | --- |
| Relay | SRD-05VDC-SL-C | Module | 2 | 5V, 10A/250VAC | High power switching |
| MOSFET | IRLB8743 | TO-220 | 4 | N-channel, 30V | Power switching |
| Motor Driver | DRV8833 | WSON-10 | 1 | Dual H-bridge | DC motor control |
| SSR | CPC1017N | DIP-4 | 1 | 60V, 100mA | AC power control |
| Servo Connector | CONN-3 | Header | 2 | 3-pin, 2.54mm | Servo motor interface |

**9. Passive Components - Resistors**

| **Component Type** | **Part Number** | **Package** | **Quantity** | **Value/Specs** | **Purpose** |
| --- | --- | --- | --- | --- | --- |
| Resistor | RC0805 | 0805 | 20 | 10Ω | Current limiting |
| Resistor | RC0805 | 0805 | 20 | 100Ω | Pull-up/down |
| Resistor | RC0805 | 0805 | 20 | 1KΩ | General purpose |
| Resistor | RC0805 | 0805 | 20 | 10KΩ | Pull-up/down |
| Resistor | RC0805 | 0805 | 10 | 100KΩ | Sensor dividers |
| Resistor | RC0805 | 0805 | 10 | 1MΩ | High impedance |

**10. Passive Components – Capacitors**

| **Component Type** | **Part Number** | **Package** | **Quantity** | **Value/Specs** | **Purpose** |
| --- | --- | --- | --- | --- | --- |
| Ceramic Cap | CC0805 | 0805 | 30 | 100nF (0.1μF) | Decoupling |
| Ceramic Cap | CC0805 | 0805 | 10 | 10μF | Power filtering |
| Ceramic Cap | CC0805 | 0805 | 10 | 22μF | Bulk capacitance |
| Electrolytic | ECE-1 | Radial | 10 | 100μF/16V | Power supply |
| Electrolytic | ECE-2 | Radial | 5 | 470μF/25V | Input filtering |
| Tantalum | TC-1 | 7343 | 5 | 47μF/16V | Stable capacitance |

**11. Inductors & Crystals**

| **Component Type** | **Part Number** | **Package** | **Quantity** | **Value/Specs** | **Purpose** |
| --- | --- | --- | --- | --- | --- |
| Power Inductor | LPS4012 | 4x4mm | 3 | 10μH | Buck converter |
| Power Inductor | LPS4012 | 4x4mm | 3 | 22μH | Buck converter |
| Crystal | HC-49S | Through-hole | 2 | 16MHz | MCU clock |
| Crystal | MC-306 | 3.2x2.5mm | 2 | 32.768kHz | RTC clock |
| Ferrite Bead | BLM18 | 0603 | 5 | 600Ω @100MHz | Noise suppression |

**12. Connectors & Headers**

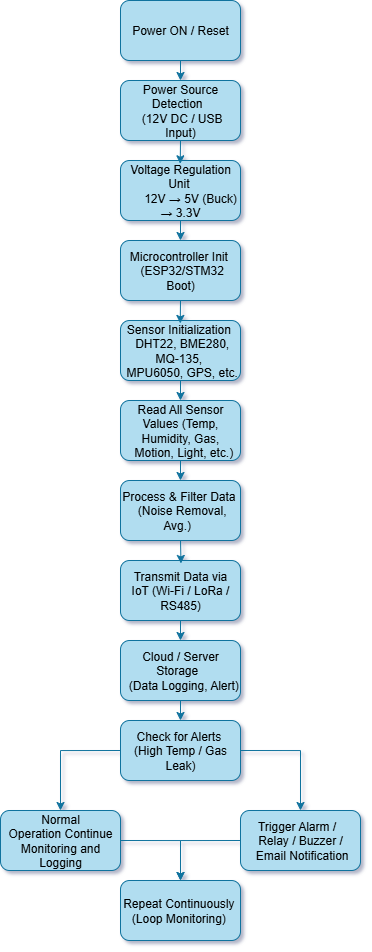
| **Component Type** | **Part Number** | **Package** | **Quantity** | **Value/Specs** | **Purpose** |
| --- | --- | --- | --- | --- | --- |
| GPIO Header | CONN-20 | 2.54mm | 2 | 20-pin, female | Expansion ports |
| Power Jack | DC-005 | Panel | 1 | 5.5x2.1mm | DC power input |
| USB-C | USB-C-31 | Through-hole | 1 | USB 2.0 | Power/programming |
| Terminal Block | TB-2 | 5.08mm | 2 | 2-pin | Power/sensor I/O |
| JST Connector | XH-2 | JST-XH | 4 | 2-pin | Battery/sensor |
| MicroSD Slot | TF-CARD | Push-push | 1 | MicroSD | Data storage |

**Essential Core Software**

* **KiCad Official Library** (built-in)
* **SnapEDA** (online component search)
* **Ultra Librarian** (component generator)
* **Component Search Engine** (for finding parts)

**Flowchart Explanation**

The IoT Node with Voltage Management System powers on, regulates the input voltage, and provides stable 5V and 3.3V outputs for all components. The microcontroller then initializes, collects sensor data, processes it, and transmits the information to the cloud or display. If any abnormal conditions are detected, the system triggers alerts through buzzer or relay while continuing real-time monitoring.



**Why the Focus on an IoT Node with Voltage Management?**

The title was chosen to highlight the two most critical aspects of a successful IoT device:

1. **The "Brain" (Microcontroller):** The IoT node requires a capable microcontroller to collect sensor data, process information, and handle communication protocols (like Wi-Fi, Bluetooth, or LoRa).
2. **The "Heart" (Voltage Management):** IoT devices are often deployed in the field and powered by unstable sources (batteries, solar, noisy DC adapters). A robust voltage management system is not an accessory; it is a **fundamental requirement**. It ensures:
   * **Stable Operation:** Prevents crashes and resets caused by voltage fluctuations.
   * **Component Protection:** Shields sensitive ICs from voltage spikes and incorrect polarity.
   * **Efficiency:** Extends battery life through efficient power conversion.
   * **Versatility:** Allows the same node to be used in different power environments.

This project focuses on integrating these two systems seamlessly onto a single, reliable PCB.

**Why KiCad was the Ideal Choice**

KiCad was selected as the EDA tool for this project for several key reasons:

* **Professional-Grade Capabilities for Complex Boards:** Designing a mixed-signal board (with digital MCU and analog power management) requires a tool that can handle different design rules. KiCad's powerful Design Rule Checker (DRC) and custom net classes are essential for this.
* **Integrated Workflow:** The seamless transition from schematic to PCB layout ensured that the complex connections between the power management ICs (like the U2 voltage regulator) and the microcontroller were perfectly synchronized.
* **Library Flexibility:** The project required specific components (e.g., specific switching regulators, MCU packages, USB connectors). KiCad's robust library system, allowing for easy creation and modification of symbols and footprints, was invaluable.
* **Cost-Effectiveness:** As a free and open-source tool, KiCad makes professional IoT hardware development accessible, aligning with the open nature of many IoT projects.

**Project Utility and Application**

This IoT Node PCB is designed to be the foundation for a wide array of applications:

* **Smart Agriculture:** Monitoring soil moisture, temperature, and humidity.
* **Industrial Monitoring:** Tracking vibration, temperature, and operational status of machinery.
* **Smart Home Devices:** Serving as a hub for sensors and actuators.
* **Environmental Sensing:** Air quality and weather monitoring stations.

Its core utility lies in providing a **pre-validated, ready-to-manufacture** design that handles the critical and often tricky aspects of power and processing, allowing developers to focus on their specific application code and sensor integration. **Step-by-Step Design Process**

The design followed a structured approach to ensure reliability, particularly for the power system.

**Schematic Capture Process**

1. **Architecture Definition:** The system was broken down into blocks: **Power Input/Protection**, **Voltage Regulation (12V->5V->3.3V)**, **Microcontroller Core**, **Programming Interface**, and **Sensor/Communication Headers**.
2. **Component Selection:**
   * **Voltage Management:** Components like switching regulators for high-efficiency step-down (e.g., from 12V to 5V), LDOs for clean 3V3 for the MCU, and protection elements like fuses (F1, F2) and TVS diodes were carefully chosen.
   * **Microcontroller:** A capable MCU (like an ESP32 or STM32) was selected with sufficient I/O pins, as evidenced by the numerous GPIO nets (/21, /22, /26, etc.).
3. **Schematic Design:**
   * Each block was drawn on the schematic sheet.
   * **Power Paths were clearly defined:** Thick lines and clear net labels (/12V, /5V, /5V\_P, /3V3, GND) were used to trace the power flow from input to every IC.
   * **Decoupling Capacitors** were placed close to every power pin on the MCU and other ICs to ensure stable operation.
   * The **USB-to-Serial interface (CH340C)** was included for both programming and power.

**PCB Design Process**

This phase was critical for realizing a stable IoT node.

1. **Board Outline and Layer Stack-up:** A suitable size was defined on the Edge.Cuts layer. A 2-layer design was likely used, as per the file's layer setup (F.Cu & B.Cu).
2. **Strategic Component Placement:**
   * **Power Section First:** The input connector, fuses, and voltage regulators (U2) were placed first. Their input and output capacitors were positioned *immediately adjacent* to the regulator pins to minimize loop areas and ensure stability.
   * **MCU Placement:** The microcontroller was placed centrally to simplify routing to all GPIO headers.
   * **Noise Separation:** The analog and power sections were kept away from the high-speed digital crystal oscillator circuit.
3. **Critical Routing:**
   * **Power Planes:** A solid **Ground Pour** on both layers was created to provide a low-impedance return path and act as a shield. A pour was also used for the main 5V or 3V3 rail where possible.
   * **Wide Traces for High Current:** Traces carrying higher current (e.g., from the 12V input to the regulator) were drawn with wider widths than signal traces.
   * **Sensitive Signal Routing:** Signals like the crystal oscillator were kept short and close to the MCU, with guarding ground traces to prevent noise coupling.
4. **Design Rule Check (DRC):** A strict DRC was run with appropriate clearances for power and voltage settings to ensure manufacturability and reliability.

**3D View and Final Review**

1. **3D Model Integration:** 3D models were assigned to key components (connectors, MCU, USB port) to visualize the final assembled board.
2. **Mechanical Verification:** The 3D viewer was used to check for any component collisions, especially for taller parts like electrolytic capacitors or connectors.
3. **Assembly Review:** The 3D view provided a clear picture for the assembly house, confirming component placement and orientation. This step was crucial to verify that the board was not only electrically sound but also physically practical.

**Limitations of the Custom PCB vs. a Breadboard**

While a custom PCB is the final, professional solution, it has specific limitations during the development phase compared to a breadboard.

1. **Permanence and Inflexibility:**
   * **PCB:** Once manufactured, the circuit is fixed. Correcting a design error (like a swapped pin or missing connection) requires cutting traces and adding "bodge wires," or worse, a completely new board revision (revising the PCB), which is time-consuming and costly.
   * **Breadboard:** The ultimate flexibility. You can change components, connections, and the entire circuit layout in minutes.
2. **High Initial Cost and Time:**
   * **PCB:** There is a significant upfront cost in time (design, layout, waiting for shipping) and money (manufacturing fees). You cannot test a single idea instantly.
   * **Breadboard:** Virtually zero cost and time to start prototyping. You can validate a circuit concept in an afternoon with components you have on hand.
3. **Component Accessibility:**
   * **PCB:** Uses Surface-Mount Devices (SMDs) like the 0805 resistors/capacitors and other ICs. These are difficult or impossible to hand-solder without practice and the right tools.
   * **Breadboard:** Uses Through-Hole components which are easy to plug in and remove by hand.
4. **Debugging Visibility:**
   * **PCB:** All wires are hidden inside layers of the board. Probing signals requires carefully finding test points. Debugging is more analytical and less visual.
   * **Breadboard:** All connections are completely visible and easy to probe with a multimeter or oscilloscope.

**Benefits of the Custom PCB vs. a Breadboard**

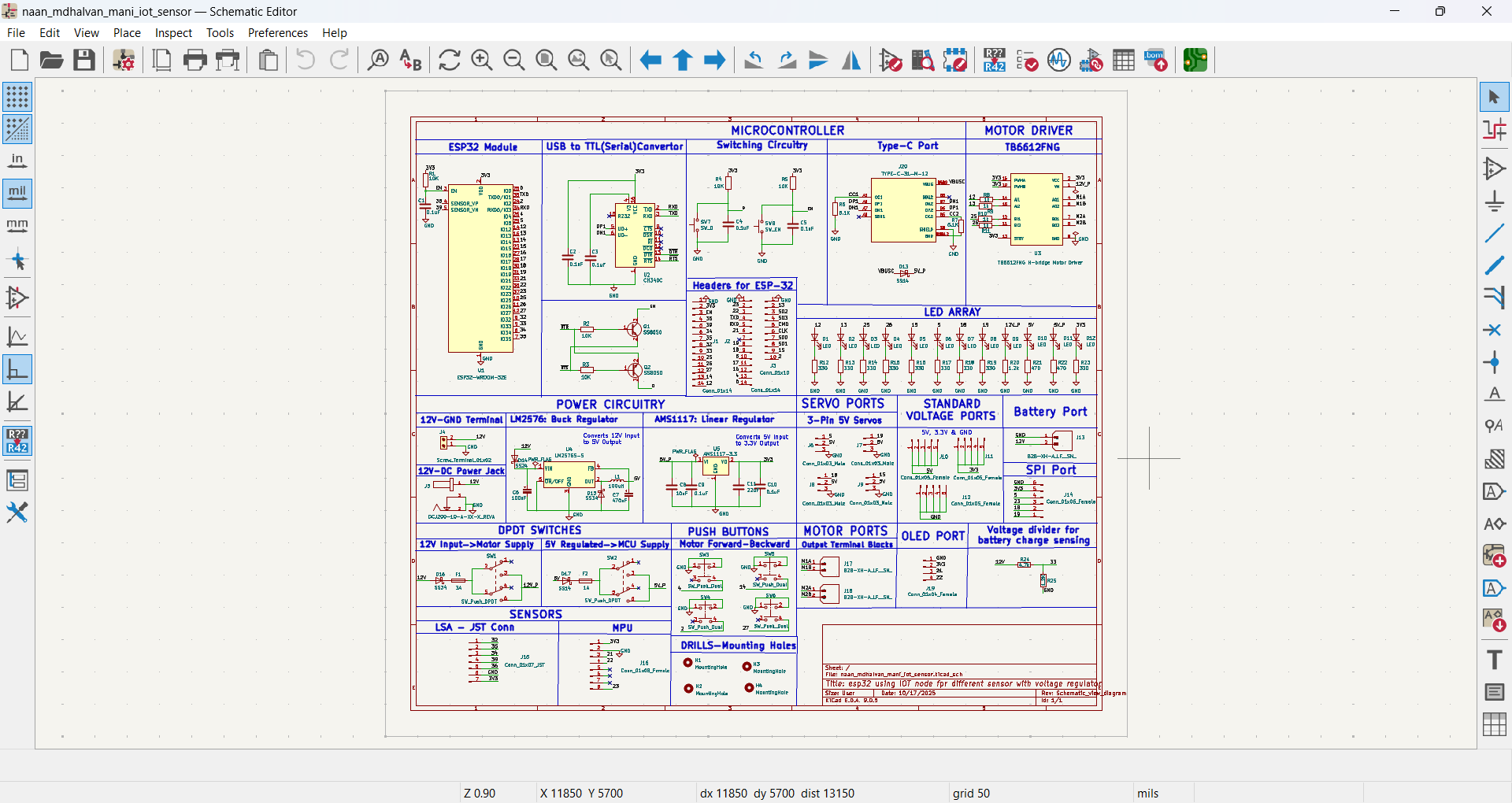
The limitations of the breadboard are precisely why the custom PCB is necessary for a final product.

1. **Reliability and Robustness:**
   * **PCB:** Solid, soldered connections do not come loose due to vibration or movement. This is non-negotiable for any device deployed in the field (like an IoT node).
   * **Breadboard:** Connections are prone to becoming loose, intermittent, or falling out, leading to erratic and unreliable behavior.
2. **Performance (Critical for IoT):**
   * **Signal Integrity:** PCB allows for controlled impedance, proper grounding planes, and short traces. This is vital for stable power delivery and high-speed communication (USB, crystal oscillators). Breadboards have long, looping wires with high inductance and capacitance, which can cause noise, signal degradation, and circuit instability.
   * **Power Distribution:** The PCB uses **power planes** and **wide traces** to deliver clean, stable power with low impedance. A breadboard's power rails are inadequate for anything but the simplest, low-power circuits.
3. **Size and Form Factor:**
   * **PCB:** Can be made compact and into any custom shape to fit inside an enclosure. Your IoT node is a single, integrated unit.
   * **Breadboard:** Bulky, fragile, and not suitable for any product that isn't permanently sitting on a lab bench.
4. **Scalability and Cost at Scale:**
   * **PCB:** While the initial cost is high, the cost per unit drops dramatically when manufacturing in volume. It is the only viable option for producing more than a handful of devices.
   * **Breadboard:** Impractical and prohibitively expensive to scale beyond a single prototype.

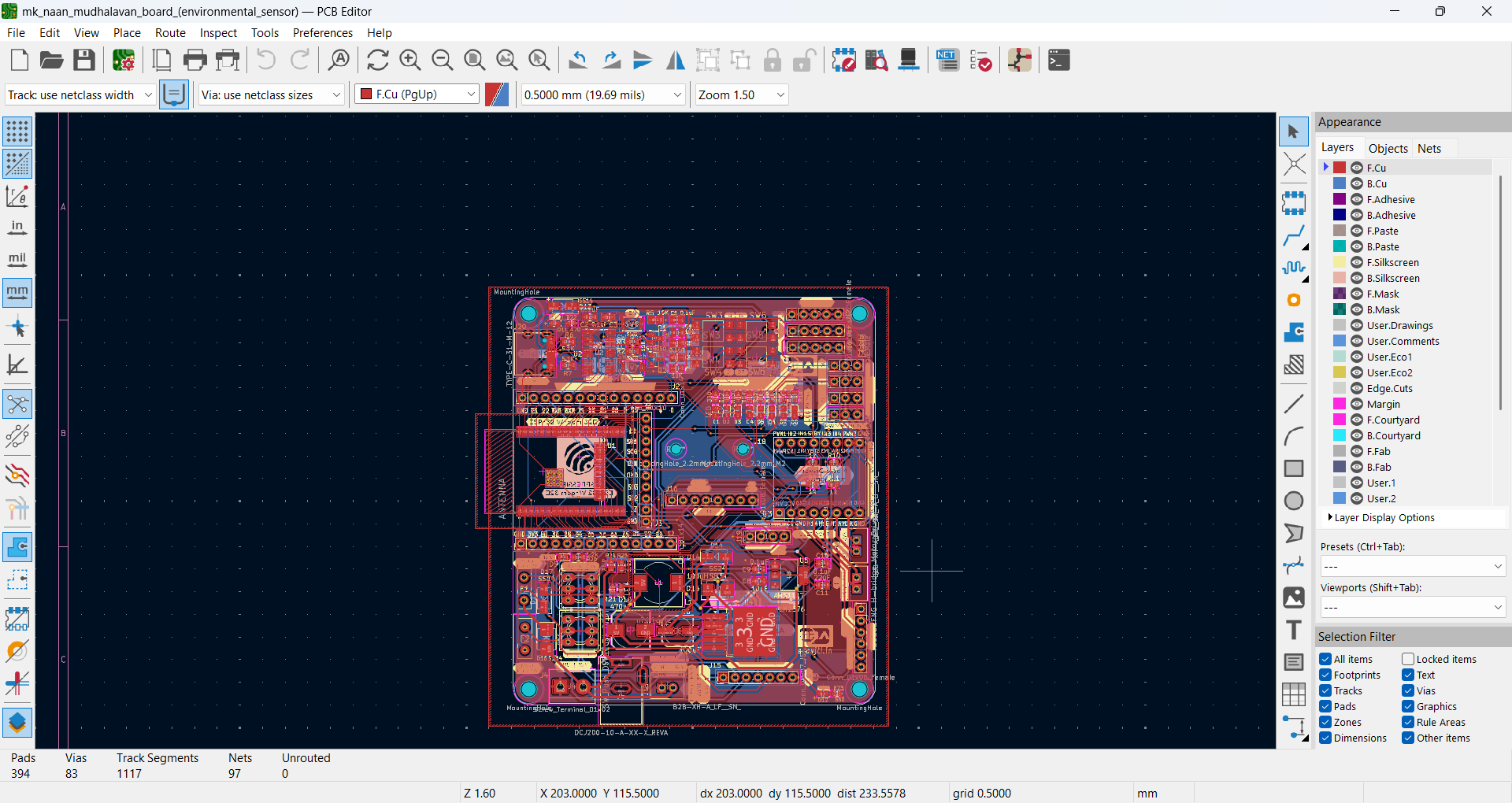
**Why the Subtitle of Schematic, PCB Design, and 3D View is Crucial**

This subtitle describes the **product development lifecycle**. Each stage has a major, non-negotiable role.

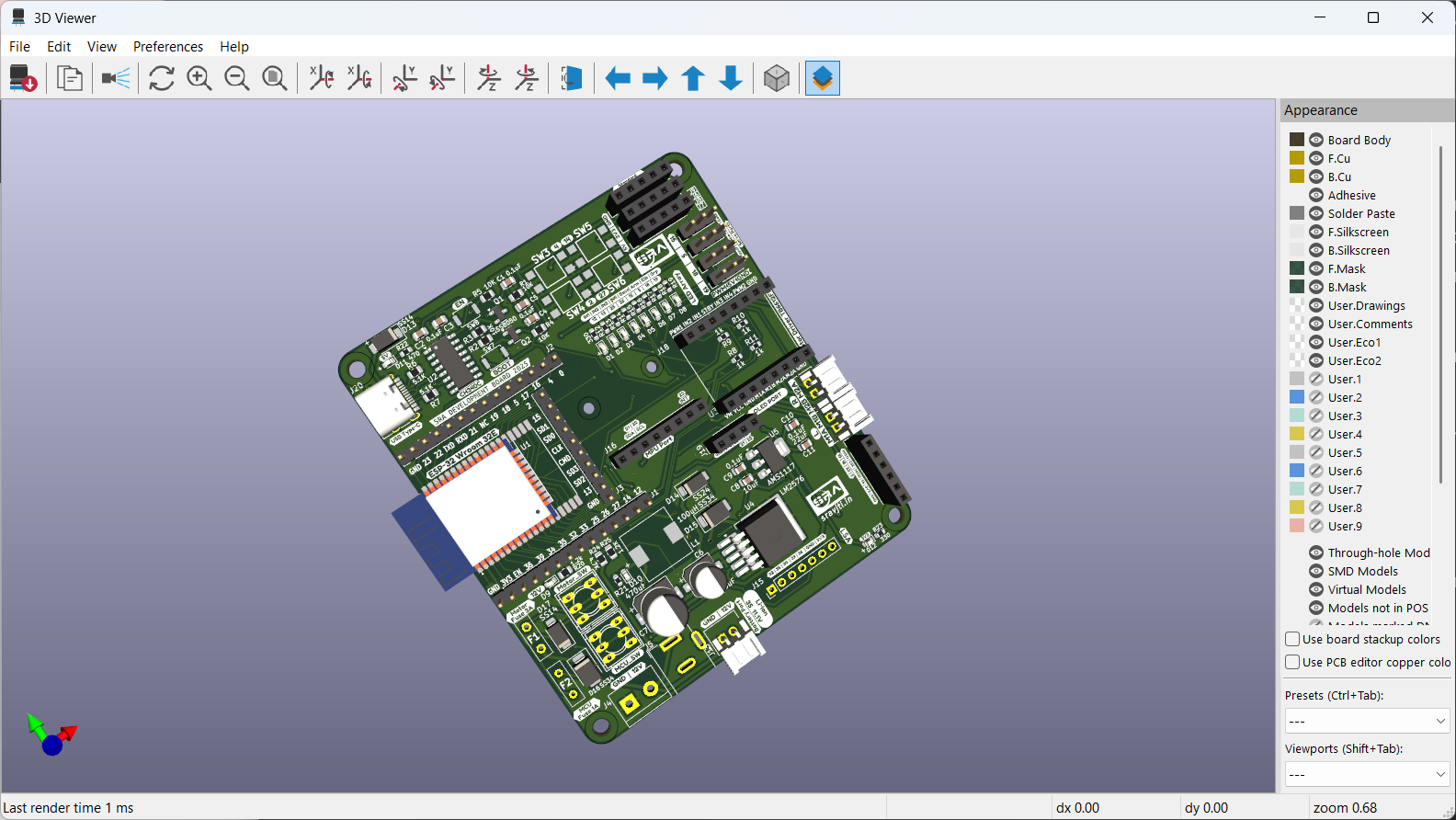
* **Schematic (The "Logic & Legal Contract"):**
  + **Role:** This is the *circuit's blueprint*. It defines *what* components are used and *how* they are connected logically.
  + **Major Role:** It is the source of truth for the BOM (Bill of Materials) and the netlist. Any error here will be replicated and harder to fix later. It's where you ensure the circuit theory is correct



* **PCB Design (The "Physical Reality"):**
  + **Role:** This is the *physical implementation* of the schematic. It defines *where* components are placed and *how* the copper traces are routed to connect them.
  + **Major Role:** This stage determines the **performance, reliability, and manufacturability** of the board. Poor layout can make a perfectly logical schematic fail completely due to noise, crosstalk, or power issues. This is where engineering meets artistry.



* **3D View (The "Virtual Prototype"):**
  + **Role:** This is a three-dimensional visualization of the assembled PCB.
  + **Major Role:** It is critical for **Mechanical Verification**. You can check for:
    - **Component Collisions:** Does a tall capacitor hit the USB connector when soldered?
    - **Enclosure Fit:** Will the board fit inside its intended case? Are the connectors aligned with the panel cutouts?
    - **Assembly Guidance:** It provides a realistic view for the assembly house, reducing erro



**In summary:** You **think** with the Schematic, you **build** with the PCB Layout, and you **verify** with the 3D View. Skipping or rushing any step leads to a non-functional or unreliable product.

**Cost Comparison: Custom PCB vs. Other Methods**

| **Feature** | **Custom PCB (Your IoT Node)** | **Breadboard** | **Perfboard/Stripboard** | **Off-the-Shelf Dev Board (e.g., Arduino, ESP32 DevKit)** |
| --- | --- | --- | --- | --- |
| Initial Unit Cost | High (Design time + PCB Fab + Components) | Very Low | Low | Very Low |
| Cost at Scale (10+ units) | Very Low (per unit) | Impractical | High (labor) | High (you're buying retail) |
| Reliability Cost | Low (Highly reliable) | Very High (Prone to failure) | Medium (Depends on skill) | Low (Generally reliable) |
| Performance Cost | Low (Optimized design) | High (Noise, instability) | Medium | Low/Medium (May have features you don't need) |
| Time & Labor Cost | High upfront, then low | Low upfront | Very High (manual labor) | Lowest (Ready to use) |
| Best For | Final product, scalable deployment | Prototyping & learning | One-off, permanent hobby projects | Prototyping, proof-of-concept, low-volume projects |

**Conclusion on Cost:** For a one-off project, an off-the-shelf development board is the cheapest and fastest option. However, the moment you plan to deploy multiple reliable, optimized IoT nodes into the field, the custom PCB becomes the most cost-effective and professional solution in the long run. Your custom PCB integrates only the necessary components, potentially lowering the unit cost below that of a generic dev board while offering superior reliability and a tailored form factor.

Github link: <https://github.com/mani-2403/naan_mudhalavan>