**IoT-Enabled Multimeter**

**Project Title:** IoT-enabled Multimeter  
**Submitted By:** Fiaz Khan A  
**Department:** Electronics and Communication Engineering  
**Date:** [8/1/2025]

**Abstract**

The IoT-enabled Multimeter is a cost-effective smart device designed for measuring voltage and current in electronic systems. It connects to a microcontroller (ESP32), captures analog data from voltage divider and current sensing circuits, and transmits the measured values to Google Sheets via the Internet. This device is specifically optimized for applications such as station maintenance in CMRL, where traditional manual methods are inefficient and costly. The aim is to provide an affordable, accurate, and automated tool that integrates easily with existing Electronic Control Units (ECUs).

**Introduction**

The traditional multimeter, although reliable, is not suitable for scalable, remote, and real-time data collection scenarios, particularly in distributed infrastructures. The proposed IoT-enabled Multimeter solves this limitation by combining sensor interfacing, microcontroller processing, Wi-Fi communication, and cloud data logging. In this project, the ESP32 acts as the central processor, interfacing with voltage and current sensors, capturing data, and uploading it to a cloud database. A matrix keypad and push button provide manual input for ECU identification and triggering the data upload process.

**Objectives**

* Design a portable, low-cost multimeter system with IoT capabilities
* Enable real-time monitoring of voltage and current from multiple stations
* Automate data collection and logging using Google Sheets API
* Simplify technician workflow by integrating serial number input
* Eliminate manual data entry errors and delays

**Hardware Components**

1. **ESP32 Microcontroller:** Central unit with built-in Wi-Fi
2. **JST Connectors:** For DC input (9V, 12V, 24V)
3. **Voltage Divider Circuit:** Scales down input voltage to measurable levels
4. **Shunt Resistor:** Measures current by voltage drop method
5. **4x5 Matrix Keypad:** For entering ECU serial numbers
6. **Push Button:** Triggers data capture and upload
7. **Capacitors:** For voltage stabilization and filtering
8. **Monitor (Optional):** For debugging or local display

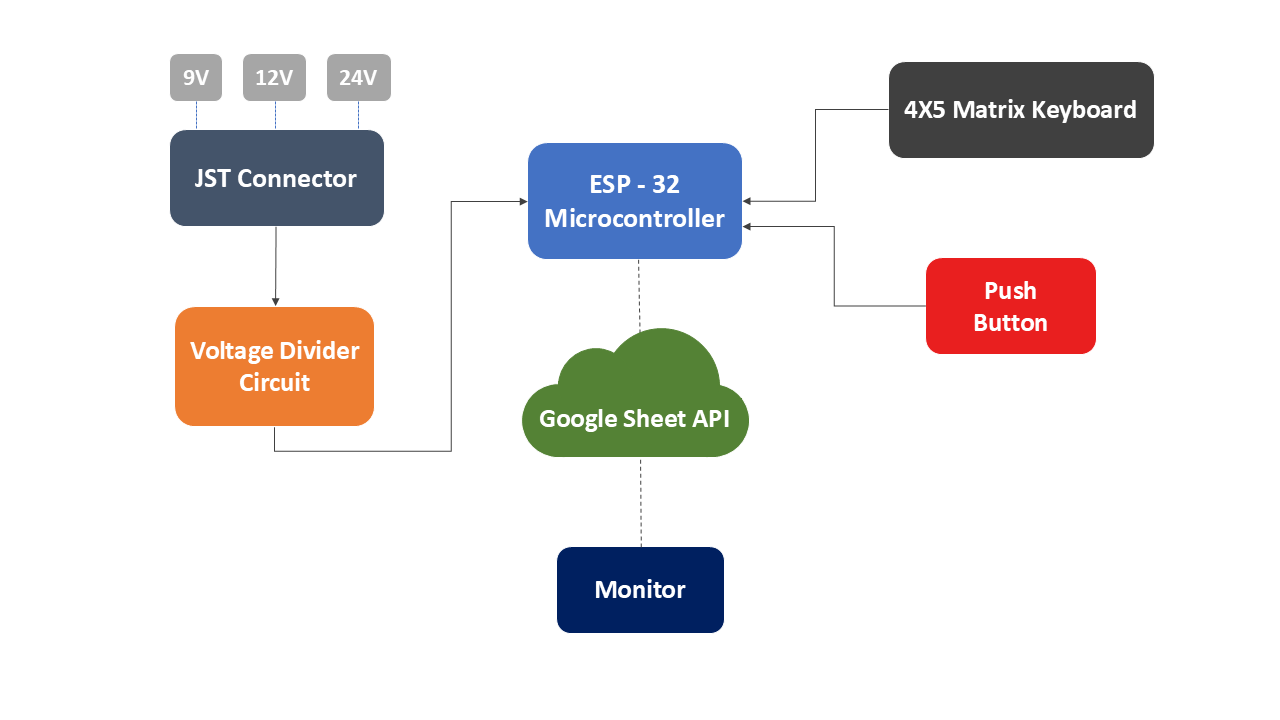
**Page 6: Software Requirements**

* **Arduino IDE:** To program the ESP32 microcontroller
* **Google Sheets API:** Used for uploading data to Google Sheets
* **Wi-Fi Network:** ESP32 needs access to send data online
* **Serial Monitor:** For local testing and debugging during development

**Page 7: Working Principle**

1. DC power (9V, 12V, or 24V) is supplied via JST connector.
2. Voltage is scaled using a voltage divider circuit.
3. Current is measured using a shunt resistor.
4. ESP32 reads these analog values via ADC pins.
5. The technician inputs the ECU serial number using the keypad.
6. Pressing the push button triggers the data read and conversion.
7. Calculated voltage and current values, along with the serial number, are sent to Google Sheets using Wi-Fi.
8. The data can be accessed remotely via cloud interface (Google Sheets).

**Block Diagram**

****

**Figure 1: Block Diagram of IoT-enabled Multimeter**

**Applications**

* **CMRL Station Maintenance:**
  + Used across more than 30 stations
  + Each station is equipped with the device for on-site measurements
  + Centralized monitoring via Google Sheets
* **Educational Labs:** For basic voltage and current experiments
* **Remote Energy Monitoring:** In solar panel or battery systems
* **Field Technicians:** For real-time logging and centralized data collection

**Conclusion & Future Scope**

This IoT-enabled multimeter successfully demonstrates an efficient alternative to manual multimeter readings for large-scale systems. It enables real-time monitoring, reduces errors, and supports centralized control. The cost-effective design ensures accessibility for wide deployment.

**Future Improvements:**

* Add mobile app for live monitoring
* Integrate alert systems for over-voltage/current conditions
* Add memory to buffer data in offline mode
* Use of MQTT protocol for industrial integration