VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

Automated Monitoring System for Machines lab

Enhancing lab efficiency, safety, and learning through real-time data.

By

- **Vijay Kumar Meena - 22BEE0348**

Under the guidance of :

Prof. Sri Hari Mandava Associate Professor Sr. School of Computer Science and Engineering

Contents

Introduction: The problem statement

The DC/AC Machine Laboratory currently relies on manual systems for monitoring electrical parameters when students perform experiments. While functional, these systems present several challenges:

Time and efforts: Manual note taking for every experiment, since we currently are not using/ buying Labview licenses as per now.

Proprietary Software Costs: Reliance on expensive LabVIEW licenses creates a financial burden and limits accessibility for students and the college, is the reason we don't.

Limited Real-time Analysis: Current setups lack comprehensive real-time visualization and analysis capabilities, hindering immediate feedback for experiments.

Our proposed system addresses these issues by developing an open-source, cost-effective, and robust solution for real-time electrical machine monitoring.

Literature Review & Technical Gaps

Our initial review indicates several existing machine monitoring solutions, but most fall short for our specific lab environment due to cost, complexity, or lack of real-time capabilities.

Existing Approaches:

Industrial PLCs/SCADA: Robust but expensive and complex for educational settings.

Proprietary DAQ Systems (e.g., LabVIEW): High license fees, limiting long-term sustainability.

Basic Microcontroller Projects: Often lack comprehensive data processing and user interface.

Identified Gaps:

Cost-Effectiveness: Need a solution that is affordable and replicable without recurring software costs.

Open-Source Adaptability: Requirements for customisation and integration with diverse lab setups.

User-Friendly Interface: A system accessible to students with varying levels of programming experience.

From Proprietary Systems to an Open-Source Solution

1. Core Hardware Technologies:

Microcontrollers (Arduino/ESP32): Low-cost, open-source platforms with built-in Analog-to-Digital Converters (ADCs) that form the heart of modern DIY data acquisition systems.

- We are starting with the Arduino for our prototype build which is our project goal for this semester, and further we will turn the prototype into a proper working embedded system for particular use case as the capstone project.

Voltage Sensing: Achieved safely using a **resistive voltage divider** to scale high machine voltages down to a microcontroller-safe range (0-5V).

Current Sensing: Implemented with non-invasive **Hall Effect sensors (ACS712)**, which translate the magnetic field of a current into a proportional analog voltage.

2. The Software & Communication Stack:

Serial Protocol: The fundamental, reliable method for transmitting sensor data from the microcontroller to a host PC over USB.

Python Ecosystem: Chosen for its rapid development and powerful open-source libraries:

PySerial: For handling the low-level serial communication.

Matplotlib: For robust and customizable data visualization.

CustomTkinter: For building a standalone, cross-platform graphical user interface (GUI).

Project Scope & Objectives

Scope

Develop a real-time monitoring system for electrical parameters (current, voltage, power) of electrical machines/workbenches in the DC/AC Machine Lab. The system will comprise:

- Microcontroller-based data acquisition.
- Python-based desktop interface for data display.
- Numerical and graphical visualization.

Objectives

1. **Implement open-source hardware & software**

   Reduce reliance on proprietary solutions and lower operational costs.

2. **Enable real-time data capture**

   Accurate and immediate acquisition of electrical parameters from machines.

3. **Provide intuitive data visualization**

   Enhance user understanding through clear numerical and graphical displays.

4. **Improve lab safety & learning**

   Offer better insights into machine performance and facilitate experimentation.

Prototype Development: Phase 1 & 2 Progress

We are currently in the initial stages of prototype development, focusing on establishing core data acquisition and basic visualization.

Phase 1: Hardware Integration & Data Logging

Objective: Collect raw electrical parameters (voltage, current) from a basic circuit (battery, LEDs, breadboard) using a microcontroller and log to an Excel file.

Microcontroller Choice: Initial prototype will use Arduino Uno for its ease of use and immediate availability. Subsequent phases will transition to ESP32 for enhanced capabilities.

Sensor Interfacing: Successful integration of current and voltage sensors.

Code Development: Firmware written for data acquisition and serial communication.

Data Export: Data streamed to PC and saved into a CSV/Excel format.

Phase 2: Python Data Processing & Graphing

Objective: Develop Python scripts to load raw data, perform calculations (e.g., power), and generate various real-time graphs.

Data Import: Python script successfully reads data from Excel files.

Parameter Calculation: Power calculation implemented ($P = V \times I$).

Graphing Modules: Utilising Matplotlib/Seaborn for visualising Voltage vs. Current, Power vs. Time, etc.

Data Validation: Preliminary checks to ensure data integrity and consistency.

Next Steps: Phase 3 & Beyond

With the core data acquisition and plotting capabilities established, the next phase focuses on refining the user experience and preparing for full lab integration.

1

Phase 3: Graphical User Interface (GUI) Development

Objective: Create an intuitive and user-friendly desktop application for real-time monitoring and analysis.

Framework: Investigate suitable Python GUI frameworks (e.g., PyQt, Tkinter, Kivy) for cross-platform compatibility.

Features: Real-time numerical display, interactive graphs, data logging controls, and experiment management.

Deployment: Package the application for easy installation on Windows machines in the lab.

2

Future Considerations: App vs. Web Interface

For lab environment: A dedicated desktop application (as in Phase 3) is ideal due to potentially limited internet access and direct hardware interaction.

For wider accessibility: A web-based interface could be explored in a later stage, potentially using Flask/Django for backend and React/Vue for frontend, but this would require a dedicated lab server.

Timeline & Milestones

Our project is structured into clear phases with defined deliverables to ensure timely completion and effective resource management.

Month 1-2

- Project planning & literature review
- Component procurement (Arduino, sensors)
- Phase 1 completion: Data acquisition to Excel

Next - Capstone project

- Phase 3 completion: Full GUI development
- System testing & debugging with lab machines
- Documentation & user manual preparation

Month 3-4

- Phase 2 completion: Python graphing & calculations
- Arduino integration tests
- GUI framework selection & basic UI wireframing

We are staying in touch with our project faculty and resolving hurdles and problems if any to meet the milestones at the required timeline or earlier

Current Status & Future Impact

Current Status

✅ **Data Acquisition & Basic Plotting Operational** We have successfully demonstrated data logging from simple circuits and can generate preliminary graphs using Python. This validates the core technical approach.

Further : We need to make power calculations, different options for graphing the values that we have got and then further update the GUI accordingly, and then move to procurement of ESP32 boards, which will enable more robust data handling, Wi-Fi capabilities, and direct integration with lab workbenches.

Anticipated Impact

Cost Savings: Eliminates expensive software licenses.

Enhanced Learning: Provides students with real-time feedback and hands-on experience with data analysis.

Improved Safety: Enables continuous monitoring of machine parameters, potentially preventing overloads.

Research Opportunities: The open-source nature allows for future expansion and customisation for advanced research.

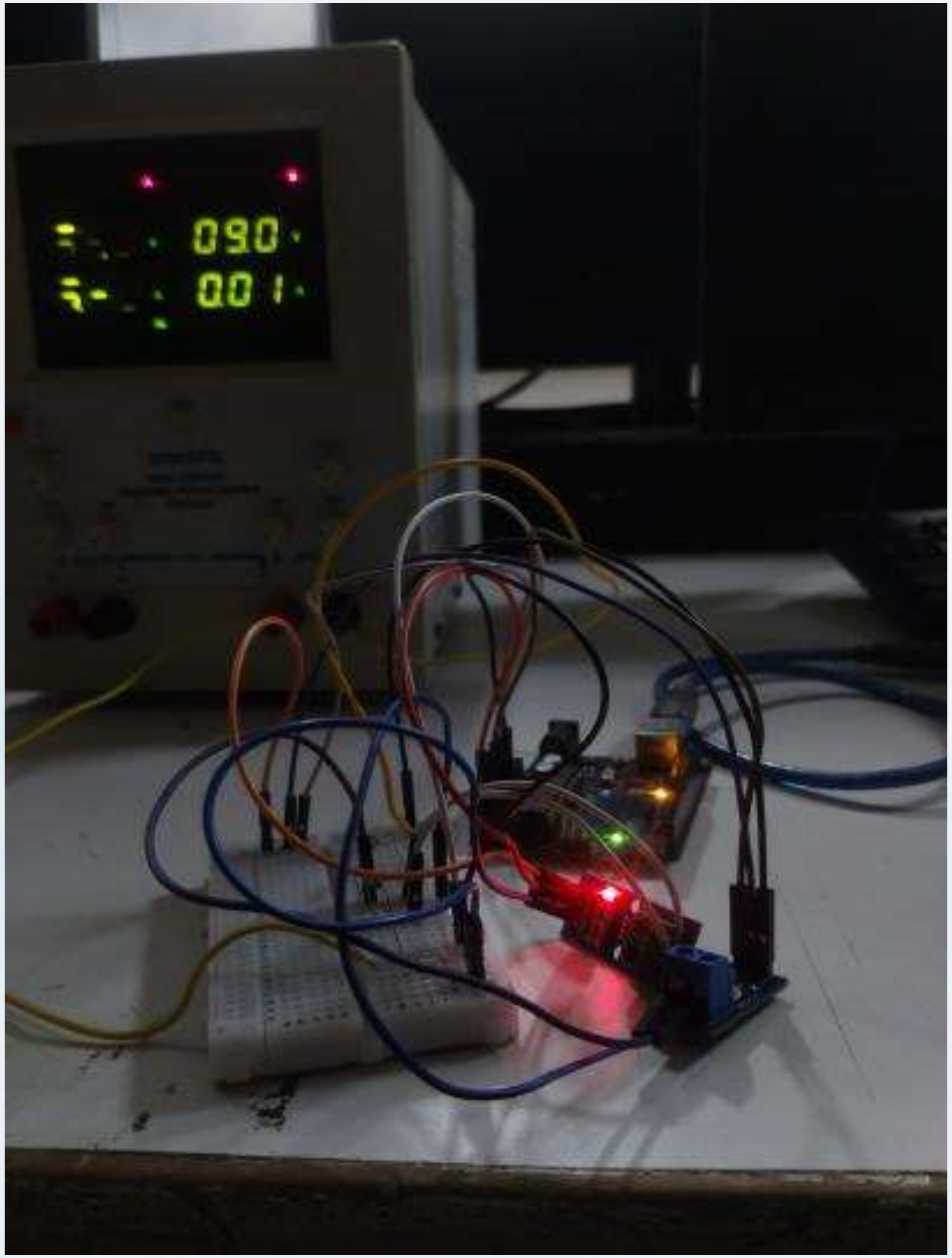
Current Status (continued)

So this is how our prototype looks like, we have used the following

1. **DC Voltage Source**
2. **Arduino**
3. **Some Connecting Wires**
4. **Breadboard**
5. **Arduino Code**
6. **Python gui**
7. **Voltage Sensor**
8. **Current Sensor**

We have a functional data acquisition circuit has been built using an Arduino Uno and a voltage sensor.

A user-friendly desktop application has been developed in Python with an interactive interface for experiment control.



Conclusion & References

The automated monitoring system project promises a significant upgrade to the DC/AC Machine Laboratory, providing a modern, cost-effective, and educational tool for future engineering students.

Key Takeaways:

Open-source solution for cost-efficiency.

Real-time data for enhanced insights.

Modular design for future expansion.

References :

Smith, J. (2019). *Microcontroller-Based Data Acquisition Systems.* Electrical Engineering Press.

Python Software Foundation. (2023). *Python Documentation.* Available at: www.python.org

NumPy Developers. (2023). *NumPy User Guide.* Available at: numpy.org

Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering, 9*(3), 90-95.