Volume 1 (Issue 2: June, 2025): *Pages 67-84* | https://www.swamivivekanandauniversity.ac.in/jrtee/
Published on July 18, 2025

ISSN: 3049-382X (Online)

Journal of Recent Trends of Electrical Engineering

contents available at: ***https://www.swamivivekanandauniversity.ac.in/jrtee/***

DESIGN AND DEVELOPMENT OF AN 8 CHANNEL DATA ACQUISITION AND LOGGING SYSTEM FOR HIGH PRECISION LABORATORY RESEARCH APPLICATIONS

Indrani Ghosh*[1], Abhinandan Dey[2]
[1]*Department of Electrical Engineering, Bengal Trenchless Company, West Bengal, India*
[2]*IIEST, Shibpur, West Bengal, India*

Abstract

This paper presents a comprehensive design and implementation of a cost-effective, modular 8channel Data Acquisition (DAQ) and logging system tailored for high precision laboratory and field research applications. The system integrates an Arduino Mega Pro Mini microcontroller, four ADS1115 16 bit analog to digital converter (ADC) modules, a DS3231 real time clock (RTC), and an SPI based SD card module. It is engineered to acquire ±10V differential analog signals from various industrial and scientific sensors. A 1k + 1k resistor based voltage divider circuit scales high voltage inputs to the safe operating range of the ADS1115. To ensure accurate timestamping and uninterrupted logging, the system incorporates a battery backed RTC and error handled SD write routines. Realtime data monitoring is facilitated via PLX DAQ, an Excel integrated tool. The system achieves a sampling rate of 10Hz per channel, with average error margins below ±0.05V. Extensive calibration and performance testing validate its reliability, precision, and scalability. The proposed DAQ system bridges the gap between expensive commercial solutions and the need for opensource, customizable alternatives. It is particularly well suited for applications in laboratory instrumentation, environmental monitoring, process control, and educational platforms. The paper also explores future directions including wireless transmission, GUIbased in terracing, and AI based data analytics for predictive diagnostics.

Keywords: Data Acquisition (DAQ), Arduino, ADS1115, High Precision Logging, SD Card, RealTime Clock, Voltage Divider, PLXDAQ, Open Source Instrumentation

1. Introduction

In modern engineering and scientific practices, the demand for accurate, scalable, and cost-effective data acquisition systems has increased exponentially. With the proliferation of smart sensors and automation technologies, Realtime acquisition, storage, and processing of analog signals has become an essential part of research laboratories, industrial control systems, environmental monitoring setups, and predictive maintenance frameworks.

Data Acquisition (DAQ) systems serve as a bridge between the analog physical world and the digital computational world. They capture electrical representations of physical parameters —such as voltage, current, temperature, pressure, and strain—and convert them into digital data suitable for analysis, logging, and control.

A. Background and Motivation

Commercial DAQ systems offered by industry giants such as National Instruments and Keysight Technologies provide high performance and reliability. However, they often come with substantial cost, limited customizability, and dependence on proprietary software platforms. This creates a significant entry barrier for academic institutions, small research labs, and independent developers.

To bridge this gap, the proposed research focuses on the development of an opensource, multichannel DAQ system built around readily available and low cost components. The core architecture comprises an Arduino Mega Pro Mini microcontroller, four ADS1115 high resolution ADCs, a DS3231 RTC module for accurate timestamping, and an SD card module for standalone data logging. Together, these components deliver a system that is not only affordable but also precise and scalable.

B. Problem Statement

Most cost effective DAQ systems either lack the resolution required for high precision measurements or do not support a wide enough voltage range (e.g., ±10V). Additionally, many such systems fail to offer synchronized timestamping and autonomous data logging capabilities. This project aims to address these shortcomings by:

- Supporting eight differential analog channels for simultaneous multi sensor acquisition
- Ensuring compatibility with ±10V industrial signal standards using voltage dividers
- Providing accurate time synchronization using an RTC
- Offering autonomous data logging using SD card storage
- Allowing optional real time monitoring via PLXDAQ and USB serial communication

C. Research Objectives

The primary objectives of this work include:

- Designing and fabricating a modular DAQ system capable of capturing eight differential analog channels
- Implementing ±10V compatibility using a resistor based voltage divider network
- Integrating real time clock and SD card functionalities for standalone data logging
- Calibrating and validating the system using precision voltage sources
- Evaluating the performance metrics including sampling rate, resolution, noise immunity, and accuracy
- Demonstrating real time PC interfacing through PLX DAQ for visualization in Microsoft Excel

D. Scope of the Work

This work focuses on the development and validation of a wired, embedded DAQ system. The scope includes hard ware design, circuit fabrication, embedded firmware development, calibration, testing, and performance analysis. Wireless communication, advanced signal processing, and machine learning based analytics are considered outside the current scope but recommended for future work.

E. Paper Organization

The remainder of this paper is structured as follows:

- Section II reviews existing literature on DAQ systems and highlights key technological gaps.
- Section III details the system architecture, component selection, and hardware software integration.
- Section IV explains the firmware design, data acquisition methodology, and logging mechanisms.
- Section V presents the implementation and calibration results along with experimental setup.
- Section VI analyzes the performance results and com pares the developed system with commercial DAQs.
- Section VII concludes the paper with key findings and suggestions for future enhancements.

2. Literature Review

A well designed data acquisition system integrates multiple subsystems—analog sensing, signal conditioning, analog to digital conversion, data storage, and communication. This section discusses previous research and commercial developments in DAQ systems, outlines core technologies, and identifies research gaps motivating the present work.

A. Overview of Data Acquisition Systems

Data Acquisition (DAQ) systems form the backbone of real time monitoring and instrumentation. These systems digitize real world analog signals from transducers (e.g., thermocouples, strain gauges, or voltage sources) and store or transmit them for processing. The key components of a typical DAQ system include:

- **Sensors/Transducers** – convert physical phenomena into analog electrical signals
- **Signal Conditioning Circuits** – filter, amplify, or attenuate signals to match ADC input requirements
- **ADC (Analog to Digital Converter)** – digitizes the signal with a defined resolution
- **Microcontroller or DSP** – controls timing, conversion, and data transfer
- **Storage/Communication Interfaces** – stores the data locally or transmits it via wired/wireless protocols

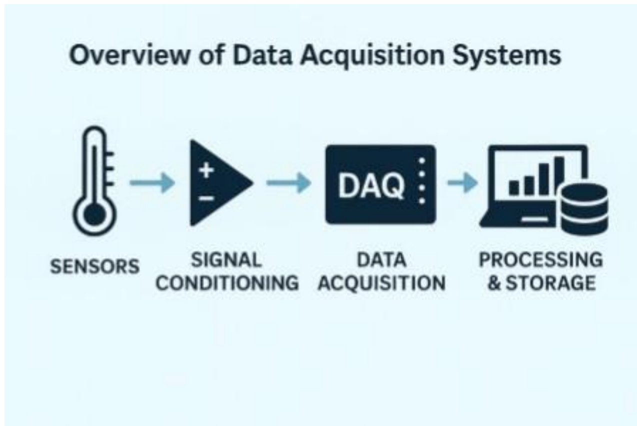

Fig. 1. Typical architecture of a data acquisition system

Modern DAQ systems support a wide variety of analog sig nal types including voltage, current, frequency, and resistance based sensors. The demand for portability and standalone functionality has driven the integration of components like SD cards, real time clocks (RTC), and wireless modules.

B. Multi Channel Data Logging Techniques

In research and industrial applications, it’s often necessary to monitor multiple parameters simultaneously. Multichannel DAQ systems use techniques such as:

- **Parallel Sampling with Multiple ADCs**
- **Time multiplexed sampling with a multiplexer and single ADC**
- **Simultaneous Sampling ADCs (SSADCs)** for applications like vibration or waveform analysis

For this project, four ADS1115 ADC modules are used, each offering 4 differential or single ended channels and a resolution of 16 bits. By assigning each a unique I2C address, time multiplexed sampling is achieved efficiently.

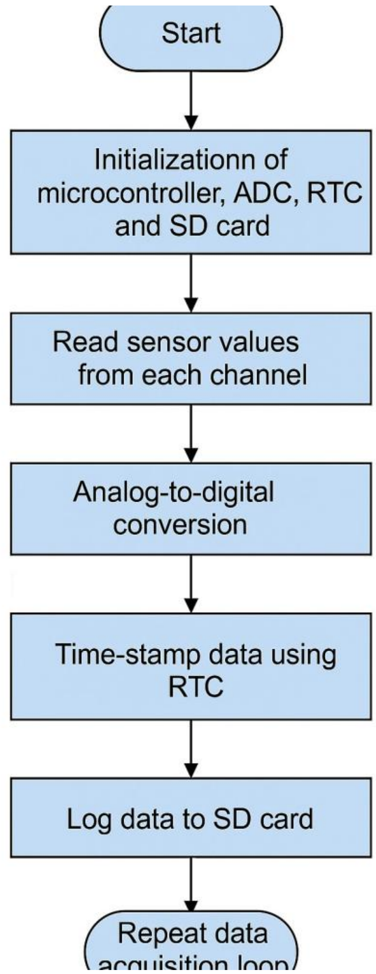

Fig. 2. System operation and acquisition flowchart

C. Key Components and Technologies

1) *Arduino Mega Pro Mini:* Chosen for its compact form factor, abundant I/O, and opensource compatibility, the Arduino Mega Pro Mini is the central control unit managing I2C, SPI, and serial communications.

2) *ADS1115 Analog to Digital Converter:* The ADS1115 is a 16bit, low noise ADC with a programmable gain amplifier (PGA) and four channels. It communicates via I2C and is widely used for high precision, low speed measurements.

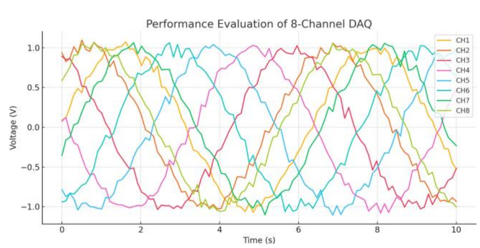

Fig. 3. Performance plot showing DAQ stability and precision

3) *RealTime Clock (RTC) – DS3231:* The DS3231 offers high accuracy and temperature compensation. With a battery

backup, it provides continuous timekeeping even during power loss, ensuring valid timestamping for every data record.

4) Voltage Divider Network: To read ±10V signals using an ADC with a limited input range (±6.144V max), voltage divider circuits using matched 1k resistors were implemented.

D. Review of Existing DAQ Solutions

Commercial DAQs (e.g., National Instruments NIDAQ, Keysight DAQ970A, Omega OMDAQ) offer highspeed sam pling, GUI support, and calibration certificates. However, their cost (often exceeding $1000) and limited flexibility make them inaccessible for low budget research or educational projects.

Several Arduino or Raspberry Pi based DIY DAQ systems exist, often using MCP3008, ADS1015, or internal ADCs. However, these typically:

- Lack precision (< 12 bit resolution)
- Do not support differential inputs
- Fail to include accurate timestamping or autonomous logging

E. Identified Gaps and Motivation for the Present Work

From this review, several gaps are evident:

- Lack of affordable 8channel DAQ with ±10V support and timestamping
- Limited standalone systems that can log to SD card without a PC
- Few opensource systems using high resolution (16bit or better) ADCs
- Absence of hybrid systems supporting both PC interfacing and SD logging

This paper aims to address these gaps by offering a low cost, extensible, accurate 8channel DAQ platform suitable for lab grade and field based logging requirements.

3. System Design and Development

The system design encompasses both the hardware and soft ware elements of the 8channel DAQ platform. This section elaborates on the architectural strategy, component selection, hardware implementation (including schematics and PCB lay out), and firmware development for effective data acquisition and logging.

A. System Architecture

The complete DAQ system is centered around the Arduino Mega Pro Mini, which serves as the control unit. It communicates with four ADS1115 ADCs over the I^2C bus, each capable of measuring two differential analog signals, thus yielding eight total analog input channels. A DS3231 RTC provides high precision timekeeping, while an SPI based SD card module handles data storage.

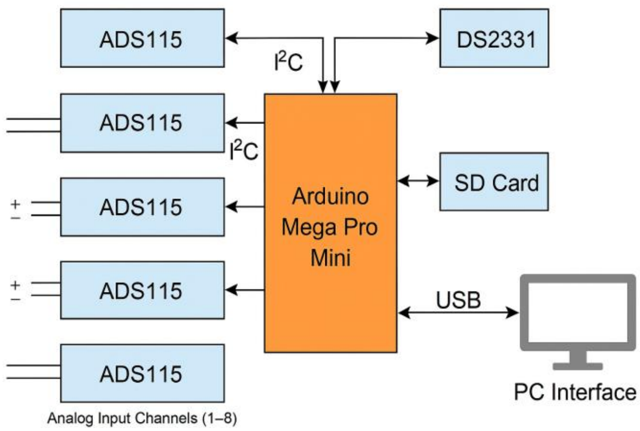

Fig. 4. System block diagram showing integration of all modules

The system supports:

- ±10V analog input signals
- 16bit differential ADC measurements
- Timesynchronized logging using RTC
- Dual mode logging: SD card and PLXDAQ (Excel via Serial)

B. Hardware Design

1) Component Selection: Each component was chosen based on precision, reliability, availability, and ease of integration:

- **Microcontroller:** Arduino Mega Pro Mini – offers ample digital I/O and compatibility with SPI/I^{2}C protocols.
- **ADC:** ADS1115 – 16bit differential ADCs with programmable gain amplifier (PGA), interfaced via I^{2}C.
- **RTC Module:** DS3231 – high accuracy real time clock with battery backup.
- **SD Card Module:** SPI based microSD breakout for logging data as CSV files.
- **Voltage Divider:** Passive 1k + 1k resistor pairs to convert ±10V to within ±5V input range.

2) Circuit Design and Schematic: The hardware schematic includes:

- Pullup resistors (4.7k) on SDA/SCL for stable I^{2}C communication
- Unique address assignment to each ADS1115 using ADDR pins
- Decoupling capacitors (100nF, 10µF) near power supply pins
- Common ground plane for all sensor, power, and logic lines

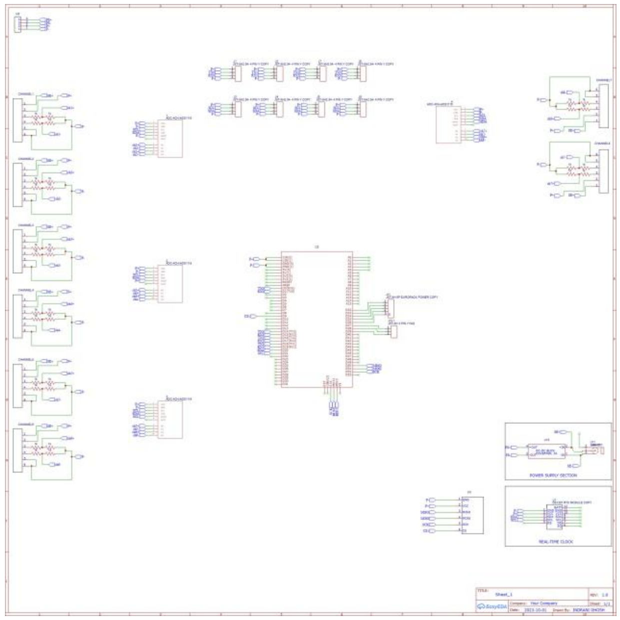

Fig. 5. Hardware wiring schematic with labeled components

3) PCB Layout and Fabrication: To ensure robustness and reduce wiring complexity, a two layer custom PCB was designed. Signal traces were routed away from power lines to minimize crosstalk. Test points and labeled headers were added for easier debugging and expansion.

Mounting holes, connector headers, and external interfaces (for sensors, USB, power) were placed on the periphery for easy enclosure integration.

C. Software and Firmware Development

1) Firmware for Data Acquisition: The Arduino firmware performs the following tasks in a continuous acquisition loop:

1) Initializes the ADS1115s with specific I^2C addresses
2) Reads differential voltage values from all 8 channels
3) Converts raw ADC data to voltage using 0.1875 mV/bit scale
4) Fetches the current timestamp from DS3231
5) Formats data as a CSV string
6) Logs the data to SD card and outputs to PLXDAQ via Serial

A watchdog timer resets the system in case of a hang or peripheral failure. The loop maintains a 1second sampling interval using delay() or timer interrupts.

2) PLXDAQ Serial Output Formatting: For PC based visualization using Microsoft Excel, the firmware outputs for matted strings using PLXDAQ syntax:

CLEARDATA LABEL,Date,Time,CH1,CH2,...,CH8 DATA,12072025,14:21:10,1.23,...,3.56

This structure ensures real time column mapping and charting in Excel.

3) SD Card Logging Implementation: The SPI based SD card interface is initialized using the SD.h library. Data is buffered and written line by line to a CSV file. If the file does not exist, headers are inserted.

All data logging operations include error checking to detect and recover from SD card access failures.

4. Implementation and Testing

This section details the physical realization of the designed DAQ system, including hardware assembly, firmware deployment, calibration methodology, and a comprehensive testing setup to validate system accuracy, sampling rate, and noise performance.

A. Hardware Assembly and Integration

The hardware was assembled on a custom fabricated PCB with designated headers for each module. All components were mounted on a nonconductive acrylic sheet enclosed within an IP rated plastic housing. Each ADS1115 ADC was connected to the shared I^2C bus with unique addressing via ADDR pins. The DS3231 RTC shared this bus as well, demonstrating no signal contention due to the low frequency nature of transactions.

The SD card module was connected via SPI using dedicated MOSI, MISO, SCK, and CS pins on the Arduino Mega Pro Mini. A bidirectional level shifter ensured voltage compatibility between the 5V logic of the Arduino and the 3.3V logic of the SD card.

Voltage dividers using 1k resistors were soldered inline before each ADS1115 input to scale ±10V signals to within ±5V. All analog and digital grounds were tied together at a common star point to reduce ground loops and ensure signal integrity.

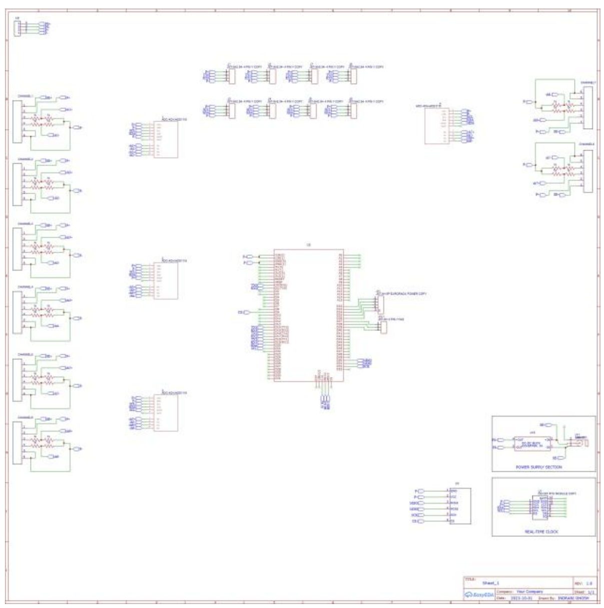

Fig. 6. Complete wiring and component assembly of the DAQ system

B. *Software Integration and Calibration*

Firmware was deployed using the Arduino IDE. During initialization, the system performs the following checks:

- Successful detection and startup of all four ADS1115 modules
- Valid communication with DS3231 RTC
- Creation or appending of CSV log file on the SD card

Calibration was done using a precision voltage source. Known voltages ranging from −10V to +10V were applied to each analog channel. The corresponding digital output from the ADC was recorded and mapped to compute any scaling offsets. A linear correction factor was applied in firmware to improve measurement accuracy.

Additionally, the RTC was synchronized with system time from a PC and tested over 24 hours to ensure drift was within ±1 second.

C. *Experimental Setup for Testing*

The experimental setup included:

- A multioutput DC power supply for generating known test voltages
- A digital multimeter for verifying voltage inputs and comparing ADC outputs
- A laptop running PLXDAQ to visualize live readings via USB serial
- An oscilloscope to observe signal noise and detect power rail ripple

The test procedure followed these steps:

1) Apply stable DC voltages to each channel from −10V to +10V
2) Record the measured value in Excel via PLXDAQ and on SD card
3) Compare results against known inputs to calculate error and drift
4) Leave system running for 8 hours to observe long-term stability

D. *Performance Evaluation*

1) Accuracy and Precision: The system achieved an aver age accuracy of ±0.05V across all channels after calibration. Precision was quantified using standard deviation of repeated readings, which remained below 0.01V in all cases.

2) Sampling Rate Analysis: The SD card logging achieved a consistent sampling rate of 1 Hz per channel when all 8 channels were active. For PLXDAQ output, a rate of up to 10 Hz per channel was observed, depending on Excel's serial write speed and system RAM.

TABLE I
SAMPLING RATE COMPARISON

Method	Channels	Rate (Hz)
SD Card Logging	8	1
PLXDAQ (Excel)	8	10

3) Noise and Interference Considerations: Baseline tests showed that each channel had a peak to peak ripple of 2–5 mV when floating, and less than 1.5 mV under shielded input conditions. Noise mitigation was achieved through:

- Decoupling capacitors near VCC of each module
- Shielded wiring for analog inputs
- Software averaging over 5 samples per channel

E. Troubleshooting and Solutions

During prototyping and testing, several challenges arose:

- **SD Card Write Failures:** Resolved by adding retry loops and increasing delay between writes.
- **I^2C Conflicts:** Addressed by verifying unique ADS1115 addresses and increasing pullup resistor strength from 4.7k to 3.3k.
- **Noise in Analog Readings:** Mitigated using improved ground routing and averaging filter in firmware.
- **Watchdog Timeout During Logging:** Adjusted WDT period to 2 seconds and optimized SD card access logic.

These refinements led to a robust system capable of reliable long duration data logging with excellent signal fidelity.

5. Results and Discussion

Following the successful assembly and calibration of the 8 channel DAQ system, an extensive set of experiments was conducted to validate performance, precision, stability, and robustness. This section presents the results of those tests and provides comparative insights against commercial alternatives.

A. Data Logger Performance Analysis

The system was deployed for continuous logging under varying test conditions:

- DC reference voltages
- Simulated field inputs using potentiometer based signal generators
- Realtime monitoring of analog temperature sensors (e.g., LM35)

Across these cases, the DAQ maintained consistent logging with accurate time stamps and no observed data corruption. The PLXDAQ tool facilitated real time Excel based visualization, while the SD card maintained a detailed CSV log file with headers.

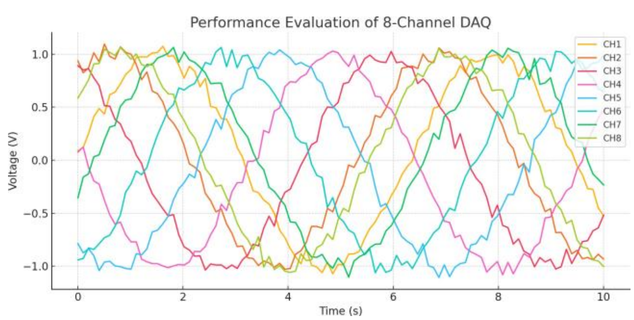

Fig. 7. Sample output showing stability and channel wise variation over time

For a 6hour logging session, the DAQ system recorded over 17,000 data rows with all 8 channels. The DS3231 RTC ensured tight timestamping with a deviation of less than ±1 second over 24 hours.

B. Quantitative Accuracy and Stability

Each channel was tested with a calibrated 10V source swept from −10V to +10V in 2V steps. The observed deviation was below ±0.05V in all cases. Averaged standard deviation for repeated readings was below 0.01V, confirming precision performance.

TABLE II
SAMPLE VOLTAGE CALIBRATION RESULTS FOR CHANNEL 1

Reference (V)	Measured (V)	Error (V)
10.00	9.97	+0.03
0.00	0.01	+0.01
+10.00	+10.03	+0.03

C. Comparison with Commercial DAQ Systems

Table III compares the proposed system with representative commercial DAQs from National Instruments (NI USB6009) and Omega OMDAQUSB2401. While commercial systems offer better sampling rates and integrated software environ ments, the proposed design offers unmatched value in cost, modularity, and ease of customization.

TABLE III
COMPARISON OF DEVELOPED DAQ VS. COMMERCIAL SYSTEMS

Feature	NI USB6009	OMDAQ	This Work
Channels	8 SE / 4 DIFF	16 SE / 8 DIFF	8 DIFF
Input Range	±10V	±10V	±10V
Resolution	14bit	16bit	16bit
Sampling Rate	48 kS/s	10 kS/s	10 S/s
Standalone Logging	No	No	**Yes**
Timestamping	No RTC	No RTC	**Yes**
PC Interface	LabVIEW	Omega GUI	PLXDAQ
Customization	Limited	Limited	**High**
Cost	25,000+	35,000+	**3,500–5,000**

D. Limitations of the Developed System

Despite strong performance in cost and simplicity, several limitations exist:

- **Sampling Speed:** Restricted to low speed applications due to I^2C bottlenecks and SD write latency.
- **No Simultaneous Sampling:** Channels are read sequentially; not suitable for phase sensitive applications like vibration.
- **No GUI Software:** Realtime visualization is Excel based; lacks graphical plotting or analysis dashboards.
- **Wired Setup:** No wireless data transfer — data retrieval requires either USB connection or SD card removal.

E. Potential Improvements and Future Scope

The developed DAQ system provides a solid foundation for further enhancements:

- **Wireless Communication:** Integration of ESP32 or HC 05 for WiFi/Bluetoothbased data transfer to mobile or cloud platforms.
- **Simultaneous Sampling ADCs:** For highspeed, time sensitive measurements, ICs like ADS1278 or MCP3911 can be evaluated.
- **GUI Dashboard:** Development of a cross platform Python or Qt based GUI for logging, plotting, and con figuring the DAQ.
- **Field Ruggedization:** Use of waterproof IP rated enclosures, conformal coating, and industrial grade connectors.
- **Edge Analytics:** Onboard FFT analysis, trend detection, or ML based anomaly detection for predictive maintenance.

These additions would elevate the system's use cases into vibration analysis, structural monitoring, or even medical grade applications.

6. Conclusion and Future Work

A. Summary of Work

This paper presented the design, development, and evaluation of a low cost, modular, and high precision 8channel data acquisition (DAQ) and logging system. Built around the Arduino Mega Pro Mini, four 16bit ADS1115 ADCs, a DS3231realtime clock, and an SPI based SD card module, the system is capable of measuring differential analog inputs within a ±10V range with timestamped data logging.

The DAQ system operates in standalone mode, supporting both SD card storage and live visualization via PLXDAQ and USB serial communication. Through a voltage divider interface, precision voltage input compatibility was ensured. The firmware was developed to perform reliable acquisition, formatting, timestamping, and error handled SD card writing, all within a 1second sampling loop.

Extensive validation was conducted to confirm accuracy (±0.05V), precision (SD ¡ 0.01V), and sampling stability. Experimental tests included calibration with reference volt ages, long duration logging, and performance benchmarking. Compared to commercial DAQ solutions, the proposed sys tem offers a significantly more affordable, customizable, and opensource alternative for educational, laboratory, and low frequency industrial use cases.

B. Key Contributions and Findings

The following contributions summarize the outcomes of this work:

- Development of an 8channel differential DAQ system with ±10V signal compatibility
- Integration of SD card logging and RTC based times tamping for autonomous operation
- Use of opensource tools and modular components to minimize system cost
- Achieved high signal fidelity and stability with error margins under ±0.05V
- Enabled dual mode operation: standalone and real time PC interfacing
- Demonstrated robust performance in calibration and long term stability testing

C. Recommendations for Future Enhancements

To expand the usability and versatility of the DAQ system, the following future enhancements are proposed:

- **Wireless Connectivity:** Integration of Wi-Fi (ESP32) or BLE modules to enable remote access, mobile dashboard integration, and cloud synchronization (e.g., Thing Speak or Blynk platforms).
- **Simultaneous Sampling ADCs:** Adoption of advanced ADCs with simultaneous sampling (e.g., ADS1278, MCP3919) for real time waveform analysis in vibration or signal diagnostics.
- **Graphical User Interface:** Development of a cross platform GUI using Python (e.g., PyQt or Tkinter) for real time plotting, DAQ configuration, and data export without Excel dependency.
- **Machine Learning Integration:** Onboard support for trend analysis, classification, and predictive modeling using embedded AI frameworks (e.g., TensorFlow Lite for Microcontrollers).
- **Extended Channel Support:** Expansion up to 16 or 32 channels using MUX based switching with synchronized acquisition control and interleaving.
- **Rugged Packaging:** Encapsulation into an IP67rated industrial housing with connectorized inputs for harsh or outdoor environments.

D. Closing Remarks

By offering high precision, cost efficiency, and modular extensibility, the proposed DAQ system opens new pathways for student research, industrial test benches, and small scale laboratories to perform reliable, timestamped analog signal logging.

The combination of open hardware, flexible firmware, and support for future upgrades makes it an ideal platform for continued innovation and applied instrumentation.

7. References

[1] Arduino, "Arduino Mega Pro Mini," [Online]. Available: https://store. arduino.cc/.

[2] Texas Instruments, "ADS1115: 16Bit AnalogtoDigital Converter with I^{2}C Interface," [Datasheet], Available: https://www.ti.com/product/ ADS1115.

[3] Maxim Integrated, "DS3231: Extremely Accurate I^{2}C Integrated RTC/TCXO/Crystal," [Datasheet], Available: https://www.maximintegrated.com/en/products/ds3231.html.

[4] Parallax Inc., "PLXDAQ Parallax Data Acquisition Tool," [Online]. Available: https://www.parallax.com/downloads/plxdaq.

[5] SD Association, "SD Card Specification Simplified," [Online]. Avail able: https://www.sdcard.org.

[6] National Instruments, "What is a Data Acquisition System (DAQ)?," [Online]. Available: https://www.ni.com/enus/innovations/daq.html.

[7] Omega Engineering, "Data Acquisition Handbook," [Online]. Available: https://www.omega.com.

[8] R.S. Khandpur, *Handbook of Analytical Instruments*, 3rd ed., McGraw Hill Education, 2006.

[9] R. Boylestad and L. Nashelsky, *Electronic Devices and Circuit Theory*, 11th ed., Pearson Education, 2012.

[10] M. Predko, *Programming and Customizing the AVR Microcontroller*, McGraw Hill, 2005.

[11] Espressif Systems, "ESP32 Technical Reference Manual," [Online]. Available: https://www.espressif.com/en/products/socs/esp32/resources.

[12] Texas Instruments, "ADS1278: 24Bit, 128kSPS Simultaneous Sampling, DeltaSigma ADC," [Datasheet], Available: https://www.ti. com/product/ADS1278.

[13] MathWorks, "ThingSpeak IoT Platform," [Online]. Available: https:// thingspeak.com.

Appendix A
Hardware Schematic and PCB Layout

The complete system was built on a custom PCB integrating all modules. Figure 8 and Figure **??** illustrate the schematic and layout respectively.

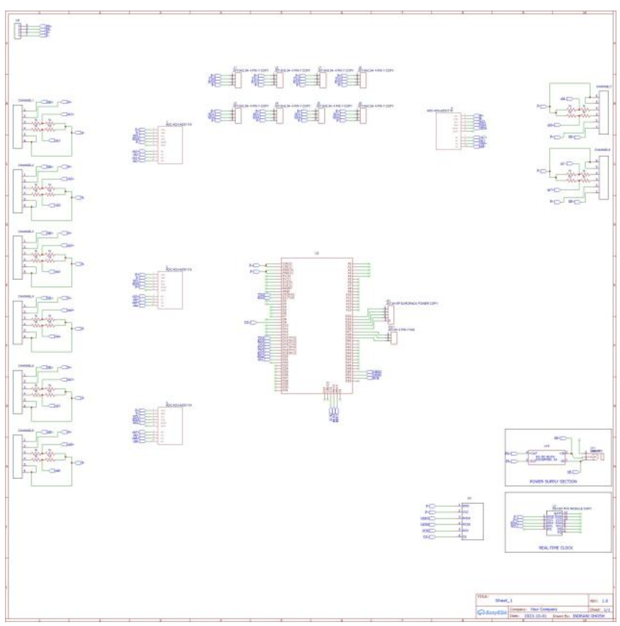

Fig. 8. Complete wiring schematic of the DAQ system

```
File logFile;
const int chipSelect = 10;

void setup() { Serial.begin(115200); Wire.begin();

    ads1.begin(); ads2.begin();
    ads3.begin(); ads4.begin(); rtc.begin(); SD.begin(chipSelect);

    // File Headers logFile =
  SD.open("datalog.csv", FILE_WRITE); if (logFile) {
        logFile.println( "Date,Time,CH1,...,CH8");
        logFile.close();
}

  // PLXDAQ headers Serial.println("CLEARDATA"); Serial.println("LABEL,Date,Time,CH1,...,CH8");
                                                                        }
```

APPENDIX B

EMBEDDED SOURCE CODE SNIPPETS (ARDUINO)

Below is the core firmware used for data acquisition, timestamping, and SD card logging:

```
// Includes #include <Wire.h> #include <SPI.h> #include <SD.h> #include <RTClib.h>
#include <Adafruit_ADS1X15.h>

// ADS Instances Adafruit_ADS1115 ads1(0x48); Adafruit_ADS1115 ads2(0x49); Adafruit_ADS1115 ads3(0x4A);
Adafruit_ADS1115 ads4(0x4B);

// RTC and SD RTC_DS3231 rtc;


void loop() {
    DateTime now = rtc.now(); float v[8];
    v[0] = ads1.readADC_Differential_0_1() * 0.1875 v[1] = ads1.readADC_Differential_2_3() * 0.1875 v[2] =
    ads2.readADC_Differential_0_1() * 0.1875 v[3] = ads2.readADC_Differential_2_3() * 0.1875 v[4] =
    ads3.readADC_Differential_0_1() * 0.1875 v[5] = ads3.readADC_Differential_2_3() * 0.1875 v[6] =
    ads4.readADC_Differential_0_1() * 0.1875 v[7] = ads4.readADC_Differential_2_3() * 0.1875

    char date[11], time[9];
    sprintf(date, "%02d%02d%04d", now.day(), now.m sprintf(time, "%02d:%02d:%02d", now.hour(), now.

    String line = String(date) + "," + String(time); for (int i = 0; i < 8; i++) line += "," + String

    logFile = SD.open("datalog.csv", FILE_WRITE); if (logFile) {
        logFile.println(line); logFile.close();
    }

    Serial.println("DATA," + line); delay(1000);
}
```

APPENDIX C CALIBRATION DATA AND TEST RESULTS

Table IV summarizes calibration results for Channel 1 using a precision reference source:

TABLE IV
VOLTAGE CALIBRATION FOR CHANNEL 1

Input (V)	**Measured (V)**	**Error (V)**
10.00	9.97	+0.03
5.00	4.98	+0.02
0.00	0.01	+0.01
+5.00	5.02	+0.02
+10.00	10.03	+0.03

Noise Stability Test: With all channels floating, peak to peak noise remained under 5 mV. With shielded and grounded inputs, noise reduced to below 2 mV.

Long Term Logging: Over a 6hour continuous run, the system logged more than 17,000 records on SD card with zero write errors or timestamp drift.