Low-Cost, Robust Data Acquisition System for Automotive Testing and Validation Using LabVIEW

P Kiran Kumar Reddy [1]
R V College of Engineering
Bangalore, India
pkiran1508@gmail.com

Rashmi A [2]
PES University
Bangalore, India
rashraju23@gmail.com

Adarsha B V [3]
Ather Energy
Bangalore, India
adarsha.bv@atherenergy.com

Govinda Raju M [4]
R V College of Engineering
Bangalore, India
govindarajum@rvce.edu.in

***Abstract*—Data Acquisition (DAQ) is a vital process used in various fields including engineering, medicine, and scientific research. In the automotive industry, DAQ systems are essential for vehicle testing and validation. This paper addresses the limitations of traditional DAQs by developing a low-cost, robust DAQ system using LabVIEW, a powerful graphical programming language. The objective of this project is to design and implement a cost-effective and flexible DAQ system that meets specific application needs. The system focuses on voltage probing, Controller Area Network(CAN) communication architecture, load bank configurations, and a LabVIEW interface. The methodology involves selecting appropriate peripherals, establishing Microcontroller Unit(MCU)-LabVIEW communication, designing the voltage probe mechanism, setting up CAN communication, incorporating load bank control, and integrating all features into a LabVIEW dashboard. The developed custom DAQ system demonstrated satisfactory performance, with minor voltage reading errors (0.22%) and a small percentage of frame loss (1.95%) in CAN communication. The system's usefulness has been proved, highlighting its potential as a valuable tool for testing and validating applications in the automobile industry.**

***Index Terms*—Data Acquisition, LabVIEW, Arduino, CAN communication, Automotive Industry**

I. Introduction

Data acquisition refers to the process of collecting and recording data from various sources such as sensors, instruments, and devices. It involves capturing and converting analog or digital signals into a format that can be stored and analyzed by a computer or data acquisition system. Data acquisition systems are commonly used in engineering, scientific research, and industrial applications to monitor and analyze the behavior of systems and processes. They enable the measurement and recording of various parameters, such as temperature, pressure, voltage, current, and other physical or electrical quantities. Data Acquisition Systems using LabVIEW is a powerful tool that enables engineers to design and implement data acquisition systems quickly and efficiently. LabVIEW is a graphical programming language that allows engineers to create custom data acquisition applications without writing complex code. LabVIEW offers a wide range of built-in libraries, functions, and tools that facilitate interfacing with hardware devices, such as data acquisition systems, sensors, cameras, motors, and other instruments. With LabVIEW, engineers can build custom graphical user interfaces (GUIs) to control and monitor their data acquisition systems easily.

The data acquisition system for a test bench of centrifugal pumps using Arduino microcontroller and LabVIEW software is developed in [1], to display centrifugal pump discharge pressure values, voltage, current, and volumetric flow of the test bench in real-time. A novel multichannel data acquisition system(NMDAS) that integrates additional sensor nodes is designed in [2], triggering mechanisms, and NI LabVIEW user interface with PLC data for device actuation in automobile air conditioning. The proposed model in [3] utilizes LabVIEW software to develop a virtual multimeter for diode characterization, integrating an Arduino, and sensors, and displaying accurate voltage and current measurements

In [4], a low-cost current and voltage measurement system for Non-Intrusive Load Monitoring (NILM) of Home Electrical Appliances is presented, enabling electricity consumption monitoring and empowering consumers with insights into their energy usage. The technique developed in [5] offers a low-cost and efficient solution for real-time monitoring of sensor data, using an Arduino UNO board and Python programming for data processing and visualization, demonstrating linearity and stability. Paper [6] presents a cost-effective data acquisition system with automated control capabilities, integrating sensors and internet connectivity for various applications like detecting unauthorized entry, automating garden or street lights, and temperature/fire control inside a house. In [7], a low-cost control system for Active Magnetic Bearings (AMB) is designed using LabVIEW Interface for Arduino (LIFA).

The study in [8] utilizes LabVIEW software to enhance the data acquisition interface for a rocket, demonstrating improved reliability and stability, with potential applications in aerospace for real-time monitoring and analysis. A low-cost multi-sensor data acquisition system for fault detection in 3D printed products is developed in [9], using Fused Deposition Modelling (FDM). In [10], a data acquisition system for the XENON1T experiment is described, achieving low signal thresholds, high event identification efficiency, and a readout bandwidth exceeding 300 MB/s for detecting dark matter interactions.

This paper presents the development of an innovative data acquisition (DAQ) system that aims to overcome the limitations commonly associated with traditional DAQs. Traditional DAQs are often characterized by their inflexibility, very high cost, and limited customization options for specific applications. Moreover, the procurement of traditional DAQs has become increasingly challenging due to persistent supply chain issues. To address these challenges, we propose a custom DAQ system that offers a more cost-effective and flexible solution for data acquisition. The primary objective of this project is to design and implement a modular and low-cost DAQ system that seamlessly integrates with NI LabVIEW and establishes communication with a vehicle's core through CAN communication.

The key focus of this project revolves around providing a robust DAQ system developed for the testing and validation of vehicles in the automotive industry. By leveraging the capabilities of LabVIEW, our system can be easily customized and adapted to meet the diverse requirements of various testing scenarios. Our contributions are as follows:

- Development of a novel, low-cost DAQ system that addresses the limitations of traditional solutions, enabling enhanced flexibility and cost-effectiveness in data acquisition processes.
- Integration of the DAQ system with NI LabVIEW, empowering users with a user-friendly interface and streamlined customization capabilities.
- Implementation of CAN communication to establish seamless connectivity with the vehicle's core, facilitating efficient data acquisition and analysis during testing and validation procedures.
- Successful application of the developed DAQ system for the testing and validation of automobiles, offering valuable insights and contributions to the automotive industry.

Functionalities being:

- Implementation of a data acquisition system to accurately measure the behavior of electrical systems and circuits by acquiring voltage values from a probe at a desired sampling rate.
- Integration of Controller Area Network (CAN) communication with a testing device, enabling communication with the vehicle's onboard Electronic Control Units (ECUs) to acquire and analyze data during the testing process.
- Configuration of a load bank to facilitate comprehensive analysis of the vehicle's behavior in various driving scenarios by subjecting it to different load conditions.
- Development of a Graphical User Interface (GUI) using LabVIEW to provide engineers with a user-friendly interface for efficient control and monitoring of the data acquisition system.

The rest of this paper is organized as follows. Section II outlines the design aspects of the DAQ. The design encompasses hardware and software components, sensor integration, data transmission protocols, and synchronization mechanisms. Section III presents an in-depth discussion on the main implementation strategies employed in the DAQ. Section IV discusses the performance of the DAQ. Concluding remarks and the future scope are provided in Sections V and VI respectively.

II. Design/System Model

To comprehend the system model, it is crucial to understand the system requirements. The voltage probe needs to measure voltages ranging from 0 V to 60 V at a desired sampling frequency. CAN communication is expected to transmit and receive CAN messages at a standard rate of 500 kbps, commonly used in automotive applications. Lastly, for load bank control, only 7 switches are necessary, as specified by the experimental setup.

Considering the design requirements, the system model can be divided into two parts: hardware design and software design.

A. Hardware design

The primary hardware component for this project is the Arduino Nano MCU. It was selected due to its compact size, affordability, and similar specifications to the Arduino Uno and Arduino Mega (offering more GPIOs, PWM, and analog pins). For the voltage probe, a scaling circuit is designed to enable voltage measurements up to 60 V. This is necessary because the Arduino's analog-to-digital converter (ADC) has a maximum reference voltage of 5 V. After calculations, it was determined that a scaling factor of 12 is required, where the upper voltage divider resistor is 11 times the lower one. This design allows for measurements within the full voltage range required, with which a measurement precision of 58.6 mV. To establish CAN communication, the Arduino itself does not have a built-in CAN port. Therefore, an additional component called the MCP2515 module is used. The MCP2515 serves as an interface between the Arduino and the CAN bus. It communicates with the Arduino using the SPI and converts the data into CAN frames for communication with the CAN bus and vice versa. As for the load bank control, no additional hardware design is needed. The digital output pins of the Arduino are directly connected to electronic switches (relays) that control the load switches.

B. Software design

The software design follows a layered architecture, as depicted in Figure 1.

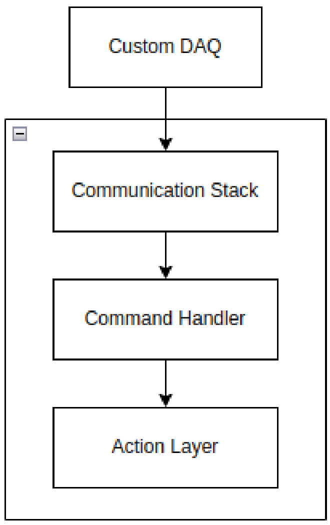

Fig. 1. System Flow

The communication stack plays a vital role in facilitating data transfer between the MCU and LabVIEW. Serial com-

979-8-3503-0219-6/23/$31.00 ©2023 IEEE

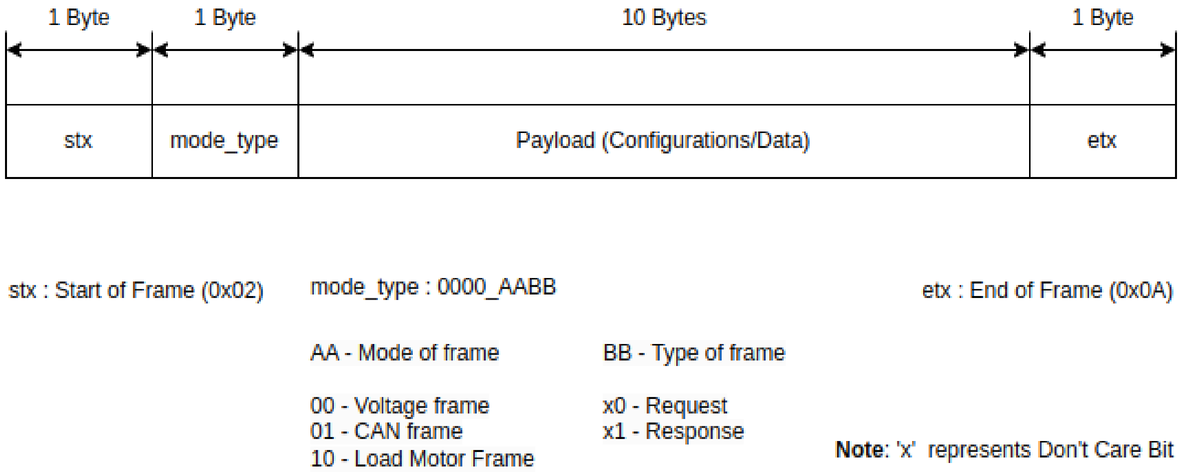

Fig. 2. Custom Frame Format

on the I2C bus. Serial communication is also preferred due to its simplicity, flexibility, and asynchronous nature. On the LabVIEW side, National Instruments VISA is utilized to read and write serial data. VISA is selected for its reliability, security, and extensive support, making it a superior choice compared to alternatives such as LINX and LIFA.

The Command handler has two main functions. Firstly, it validates the received data to ensure it is error-free and uncorrupted. Then, it parses the data to extract only the essential information. This information is subsequently utilized by the action layer. As the name suggests, the action layer acts upon the received information. On the MCU side, the action layer updates the necessary data structures to retrieve the required data. On the LabVIEW side, the data is parsed and displayed on the respective indicators. Additionally, the action layer on LabVIEW prepares request messages based on user input from the dashboard.

III. Implementation

The implementation of the system proceeded with the initial focus on establishing communication between the MCU and LabVIEW. To transmit data between the two components, serial communication in string format was considered. However, using strings would require higher bandwidth as the attributes increase and introduce additional computational costs for parsing. To address these concerns and ensure robustness and scalability, a frame format was implemented. This frame format allows for easy incorporation of additional attributes in the future.

Figure 2 illustrates the implemented frame format. The frame includes the Start of Frame (STX) and End of Frame (ETX) fields, which are necessary for synchronization between the sender and receiver. The Mode_Type field indicates the attribute being carried by the payload (mode) and whether it is a request or a response (type). Finally, the payload field carries the relevant information for the specific attribute, such as voltage probe data, CAN messages, or load bank control. Each frame has a total size of 13 bytes.

The acquisition of voltage probe data occurs within the Interrupt Service Routine (ISR) of the MCU Timer peripheral. This timer generates an interrupt every 5ms, and the sampling rate can be chosen in multiples of 5. This constraint was implemented based on user input, as data is typically requested at multiples of 5 (e.g., 10 ms, 50 ms, 75 ms), and it helps reduce the computational overhead of the overall functional loop when data is not being acquired. Based on the acquired data, the corresponding frames are prepared and stored in a buffer to be sent in every iteration.

In CAN communication, for CAN receive requests, the CAN ID from the request is stored and updated on the MCU. The MCU filters out irrelevant CAN messages and only responds with the requested CAN messages. For CAN send requests, the messages are transmitted directly to the CAN bus through the MCU without any modification.

Regarding load bank control, for each byte of the payload value, the corresponding load bank switches are turned on or off.

The LabVIEW Graphical User Interface (GUI) that integrates all the aforementioned features is depicted in Figure 3.

The left side of the panel provides input fields and a push button to send requests. On the right side, the received data is displayed. The voltage probe data can be plotted by selecting the desired probe. Currently, the CAN messages are not parsed, and all 8 bytes are displayed as is.

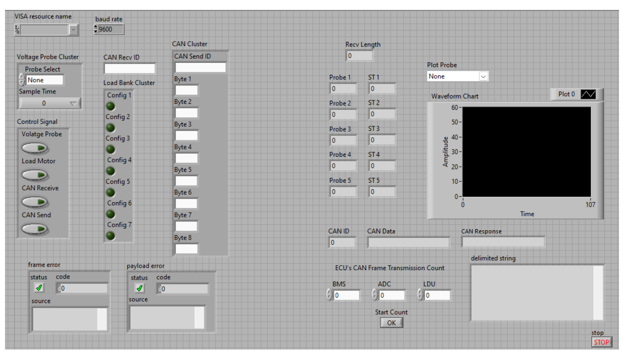

Fig. 3. LabVIEW dashboard

IV. Results Analysis and Discussion

This chapter provides a comprehensive examination and discussion of the results obtained from the evaluation of various components within the data acquisition system, including the voltage probe's performance, CAN connections, load bank control, and LabVIEW GUI.

A. Voltage Probing

The evaluation of the voltage probe demonstrates its potential by showcasing promising traits in accurately measuring voltage across the battery voltage line. The hardware of the voltage probe shows promise, particularly in terms of the required scaling and current requirements. To assess its performance, a comparison was conducted with the NI-9229 module. The NI-9229 is chosen for its error-free performance, thanks to its self-calibration routine, high input impedance, and low noise floor. These features ensure that the module is always accurate. The evaluation involved connecting both probes to the battery voltage line [11] and measuring at 100ms intervals. Table I presents a sample of readings obtained at various time instances. The observations reveal that there is no correlation between the measured voltage and the observed error at that point. The analysis reveals an average error of 0.22% that falls within an acceptable range given the context and requirements of the system.

B. CAN Communication

An efficient implementation of CAN communication was achieved using the MCP2515 module. The CAN Send and CAN Receive commands demonstrated flawless functionality, as visually observed. This was further validated by establishing a successful connection between the MCP2515 module's CANH and CANL pins and the CAN connector of the electric automobile. However, to evaluate the system's efficiency, a predetermined number of CAN messages were transmitted through the CAN bus, with an equal number of messages expected to be received and logged in LabVIEW. The results are summarized in Table II.

Upon analyzing Table II, it is evident that there is a loss of messages, indicating an average loss rate of 1.95%. However, when considering a large number of messages, the loss remains below 1%, except in cases involving a low number of messages, where an approximate loss of 3% is observed. One potential explanation for this loss could be a synchronization issue between LabVIEW and the microcontroller unit (MCU) during the initial stages of execution, leading to consistent message loss. Despite this observed loss, it does not appear to pose a significant threat to the functionality of the data acquisition system (DAQ). While a loss of messages was observed during the evaluation, with an average loss rate of 1.95%, the impact on the overall functionality of the DAQ seems negligible.

C. Load Bank Control

Testing the digital outputs of the pins connected to the switches proved to be a straightforward objective among all the objectives. The monitoring process entailed sending all 8 possible configuration commands from LabVIEW, and it was observed that all the switches responded seamlessly to the configurations without any glitches. These findings indicate the successful implementation of load bank control.

TABLE I
CAN THROUGHPUT TEST

Voltage Readings (V)		
NI-9229	**Proposed DAQ**	**Error Percentage(%)**
5.02	4.99	0.05
10.00	9.96	0.40
15.03	15.05	0.19
20.01	19.98	0.15
25.03	25.14	0.43
30.02	30.05	0.10
35.01	35.09	0.22
40.03	40.13	0.25
45.02	45.16	0.28
50.07	50.21	0.28
55.02	55.16	0.25
60.00	59.94	0.10

D. LabVIEW GUI

Figure 3 illustrates the graphical user interface (GUI) of the complete system, providing an overview of its key components and functionalities. In the top-left section, a drop-down list allows users to select the desired communication port from the available options for project execution. Various text boxes enable user input to request specific commands, and corresponding push buttons facilitate their execution.

On the right side of the GUI, the outputs are displayed. Separate indicators present the voltage probe data along with the sampling rate. Users have the option to plot the selected probe data on a graph, enhancing data visualization. Towards the bottom-right, indicators are available to display the received CAN frames, including CAN ID and corresponding CAN message content. Additionally, an indicator shows the response of the CAN send command, providing real-time feedback. Lastly, an indicator is provided to display the entire received serial data, serving as a debugging tool for troubleshooting purposes. The voltage probe and CAN data were logged into separate CSV files for further analysis and processing.

In summary, this chapter extensively covers the performance evaluation of the voltage probe, CAN connections, load bank control, and the LabVIEW GUI. The voltage probe demonstrates promising characteristics with an average error of 0.22%. The MCP2515 module enables efficient CAN communication, although some frame loss is observed, potentially attributed to synchronization issues between LabVIEW and the microcontroller unit (MCU) during the initial stages of execution. The implementation of load bank control proves successful, as all switches respond seamlessly to configurations. The LabVIEW GUI provides a user-friendly interface for communication and effectively displays the voltage probe data, received CAN frames, and received serial data for debugging purposes.

V. CONCLUSION

The project aimed to develop an affordable robust DAQ System for automotive testing, integrating voltage measurement, CAN communication, load bank configurations, and a LabVIEW dashboard. The implementation followed a layered architecture approach with a Communication Stack, Command Handler, and Action Layer. A 13-byte frame format, including STX, ETX, mode type field, and payload,

TABLE II
CAN THROUGHPUT TEST

No. of Frames Sent	No. of Frames Received	Frames Lost
100	97	3
500	484	16
1000	966	34
3000	2973	27
5000	4969	31
10000	9941	59

was adopted. A voltage divider circuit enabled measurement range accommodation, and voltage measurements were acquired through a 5ms timer ISR. CAN communication utilized an Arduino Nano with the MCP2515 module. The load bank configuration controlled seven switches based on the payload frame. Overall, the project successfully delivered an affordable and robust DAQ System meeting automotive testing requirements.

VI. Future Scope

The current DAQ system design provides a solid foundation for implementing data analytics and integrating new features seamlessly. Enhancements can include CAN Bus Load Analysis, enabling the analysis of ECU occupancy through message counting and ratio calculation. Incorporating a current sensor would enable power consumption calculation, offering insights into energy usage. Improving the voltage probe hardware to accommodate positive and negative voltage ranges, potentially using op-amps and adjusting software, would enhance measurement accuracy and system versatility. These enhancements significantly augment the capabilities of the DAQ system for automotive testing applications.

VII. Acknowledgement

We extend our sincere appreciation to all those who contributed and provided valuable input, insights, and assistance throughout the project. Their contributions were essential to the successful completion of this work, and we are truly grateful to their dedication and hard work.

References

[1] D. A. Quintero, J. A. Contreras, and J. D. Pérez, "Development of a Data Acquisition System using LabVIEW and Arduino microcontroller for a Centrifugal Pump Test Bench connected in Series and Parallel," *Journal of Physics: Conference Series*, vol. 1257, no. 1, p. 012002, 2019.

[2] S. Shalgar and R. Bindu, "Design and Development of a Novel Multichannel Data Acquisition System using Labview for an Automobile Air Conditioning Application," *Journal of Thermal Engineering*, vol. 8, no. 1, pp. 14 – 28, 2022.

[3] N. Nuryanti, W. Purnomo, and A. Supriyanto, "Development of Virtual Multimeter in Determining Diode Characteristics Based on LabVIEW," in *2021 3rd International Symposium on Material and Electrical Engineering Conference (ISMEE)*, Nov. 2021, pp. 283–289.

[4] S. Houidi, F. Auger, P. Frétaud, D. Fourer, L. Miègeville, and H. B. Attia Sethom, "Design of an Electricity Consumption Measurement System for Non Intrusive Load Monitoring," in *2019 10th International Renewable Energy Congress (IREC)*, Mar. 2019, pp. 1–6.

[5] P. Sarma and T. Bezboruah, "A Real-Time Data Acquisition System for Monitoring Sensor Data," *International Journal of Computer Sciences and Engineering*, vol. 6, pp. 1–5, Jun. 2018.

[6] A. Sharma, S. Sethi, and G. Kour, "IOT Based Data Acquistion and Automated Control System," in *2018 2nd International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE)*, Sep. 2018, pp. 146–150.

[7] T. Du, H. Geng, Y. Sun, H. Lin, H. Lv, and L. Yu, "Design and Implement of a Low Cost Control System of Active Magnetic Bearings using LabVIEW Interface for Arduino," in *2017 IEEE International Conference on Mechatronics and Automation (ICMA)*, Aug. 2017, pp. 1833–1837.

[8] F. Mujaahid, F. Anjasmara, R. Wiyagi, and K. Trinanda Putra, "Development of Data Acquisition System Using LabVIEW-based Interface for Aircraft Application," *Journal of Electrical Technology UMY*, vol. 2, Jan. 2018.

[9] S. Kumar, T. Kolekar, S. Patil, A. Bongale, K. Kotecha, A. Zaguia, and C. Prakash, "A Low-Cost Multi-Sensor Data Acquisition System for Fault Detection in Fused Deposition Modelling," *Sensors*, vol. 22, no. 2, Jan. 2022. [Online]. Available: https://www.mdpi.com/1424-8220/22/2/517.

[10] E. Aprile, J. Aalbers, F. Agostini *et al.*, "The XENON1T Data Acquisition System," *Journal of Instrumentation*, vol. 14, no. 07, p. P07016, Jul. 2019. [Online]. Available: https://dx.doi.org/10.1088/1748-0221/14/07/P07016.

[11] A. Jamaluddin, L. Sihombing, A. Supriyanto, A. Purwanto, and M. Nizam, "Design real time Battery Monitoring System using LabVIEW Interface for Arduino (LIFA)," in *2013 Joint International Conference on Rural Information & Communication Technology and Electric-Vehicle Technology (rICT & ICeV-T)*, Nov. 2013, pp. 1–4.