

An Arduino–Python Platform for Near Real-Time Signal Acquisition Systems: Design and Three-Phase Implementation

Goran Vuković
University of East Sarajevo
Faculty of Electrical Engineering
Department of Electroenergetics
East Sarajevo, BiH
goran.vukovic@etf.ues.rs.ba

Miodrag Forcan
University of East Sarajevo
Faculty of Electrical Engineering
Department of Electroenergetics
East Sarajevo, BiH
miodrag.forcan@etf.ues.rs.ba

Đorđe Kovačević
University of East Sarajevo
Faculty of Electrical Engineering
Department of Computer and
Information Sciences and Bioinformatics
East Sarajevo, BiH
djordjekovacevic135@gmail.com

Djuradj Budimir
University of Westminster
Wireless Communication Research
Group
London, UK
d.budimir@wmin.ac.uk

Abstract— This paper presents the design and implementation of an Arduino–Python platform for near real-time multi-channel signal acquisition systems, demonstrated through a three-phase electrical measurement application. The proposed platform integrates an analog signal conditioning board, an Arduino MEGA board for analog-to-digital conversion (ADC), and a PC-based Python application for continuous serial data acquisition and processing, visualization, and storage. Communication between the board and the host computer is established via USB serial interface, enabling continuous acquisition and monitoring of six channels (three voltages and three currents). The capabilities of the platform are evaluated experimentally in a laboratory setup involving a three-phase induction motor. A cohesive data processing pipeline that facilitates the seamless transition from hardware acquisition to Python-based analysis is proposed. The results demonstrate the applicability of the developed platform for laboratory-scale three-phase monitoring and signal analysis.

Keywords— Three-phase signals; Data acquisition; Arduino MEGA board; Python application

I. INTRODUCTION

The acquisition and analysis of three-phase voltage and current signals represent a fundamental requirement in both research and educational laboratories for characterizing complex power system phenomena. In recent years, Arduino-based platforms have emerged as a prominent solution for the acquisition and digital processing of three-phase voltage and current signals within the power systems research community, supporting applications such as power quality monitoring [1, 2], smart electricity metering [3], and even phasor measurement units (PMUs) [4]. Arduino boards offer a unique combination of on-board script execution and community-supported scalability, providing a portable and accurate alternative to expensive, dedicated acquisition hardware [5]. The Arduino Integrated Development Environment (IDE) leverages a C-based

programming environment, providing the necessary low-level hardware access and execution efficiency required for high-speed, real-time data acquisition. To overcome the restricted interoperability and data processing constraints of the Arduino IDE, the platform proposed in this article utilizes a Python-based interface, providing a robust framework for scalable multi-channel data logging and advanced visualization. Python's modular architecture and vast ecosystem of specialized libraries make it a robust and versatile choice for the development of automated measurement and data acquisition systems [6]. The work in [7] effectively combines Arduino and Python to create a robust control and data acquisition platform.

The core objective of this article is the development of a versatile Arduino–Python platform for near real-time signal acquisition, enabling the design of customizable systems optimized for specific experimental requirements. Centered on a three-phase implementation, the study highlights the design of a data processing pipeline as its primary contribution for synchronized multichannel acquisition. The system's practical applicability is demonstrated through a laboratory testbed, while detailed metrological calibration is considered beyond the scope of this initial design and implementation phase.

The remainder of this paper is organized as follows: Section II details the hardware design of the signal conditioning board, describing the analog front-end required to scale the three-phase voltages and currents to the ADC's operational range. Section III presents the development of an integrated data processing pipeline designed for multichannel signal acquisition and near real-time logging. In Section IV, the applicability of the proposed platform is evaluated through experimental results obtained from a laboratory testbed featuring a three-phase induction motor. Finally, the article concludes with a summary of the findings and a discussion on future research directions.



II. DESIGN OF THE SIGNAL CONDITIONING BOARD

The analog signal conditioning board serves as an isolation and scaling interface, converting bipolar high-magnitude electrical quantities into the unipolar 0-5 V range required by the microcontroller's ADC. The circuit diagram of the analog signal conditioning board for three-phase voltage and current acquisition is shown in Fig.1.

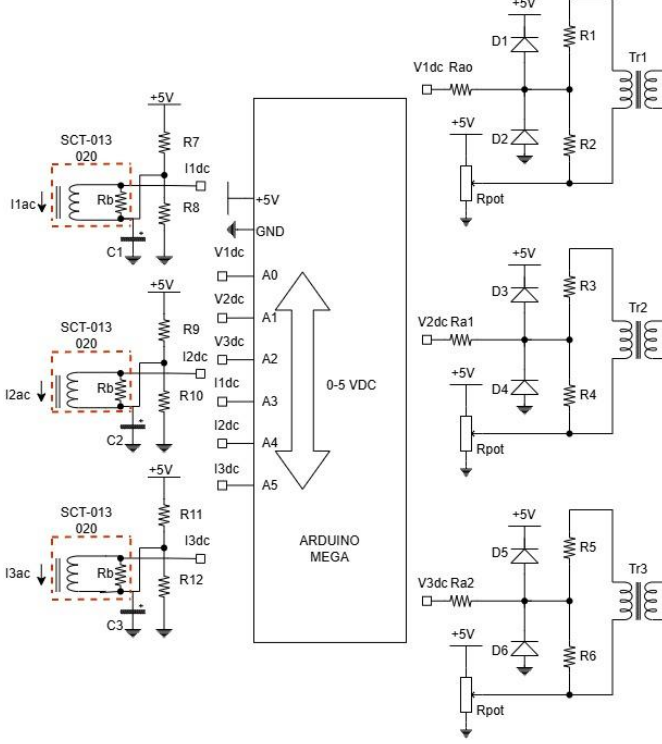


Fig. 1. Circuit diagram of the analog signal conditioning board for three-phase voltage and current acquisition

A. Voltage signal conditioning

The voltage conditioning circuits use three small power transformers (model BV EI 303 2032 [8]) as potential transformers (Tr1, Tr2, and Tr3) to provide galvanic isolation and scale source voltage $V_{sRMS} = 240$ V to a secondary level voltage $V_{1RMS} = 12.52$ V, providing an input for the resistive divider. The potential transformers feature a voltage transformation ratio of 19.167. The resistive voltage dividers, comprising $R_1 = R_3 = R_5 = 120$ k Ω and $R_2 = R_4 = R_6 = 10$ k Ω resistors, further downscale the voltage V_{1RMS} by a factor of 13, $V_{2RMS} = V_{1RMS}/13 = 0.96$ V. To accommodate the unipolar input range of the ADC, the output voltages from the resistive dividers are level-shifted by a 2.5 V DC bias. These offsets are derived from a 5 V DC source using a potentiometers $R_{pot} = 2$ k Ω . The $R_{a0} = R_{a1} = R_{a2} = 1$ k Ω series resistors placed between the voltage conditioning circuits and each ADC input pin are used to limit current into ADC protection diodes, thus providing overvoltage protection. The pairs of diodes, (D1, D2), (D3, D4), (D5, D6), limit ADC input voltages (v_{1dc} , v_{2dc} , and v_{3dc}) to the following range: -0.7 V $< v_{dc}(t) < 5.7$ V, thus clipping abnormal voltage values. The instantaneous voltage at each ADC input pin is defined by the following expression: $v_{dc}(t) = 2.5$ V $+ v_2(t)$.

The processed voltage signals (v_{1dc} , v_{2dc} , and v_{3dc}) are connected to the Arduino MEGA board analog inputs (A0, A1, and A2).

B. Current signal conditioning

The current conditioning circuits use three split-core current transformers (model SCT-013-020 [9]) with internal burden resistors R_b to step down 0 – 20 A primary ac current range into a 0 – 1 V output ac voltage ($v_{ac}(t)$) range. Resistive voltage dividers, consisting of resistors $R_7 = R_8 = R_9 = R_{10} = R_{11} = R_{12} = 10$ k Ω , are employed to establish a 2.5 V DC bias. This bias level shifts the AC output from the current sensor into the unipolar 0 – 5 V range required by the ADC. The instantaneous voltage $v_{dc}(t)$ at each ADC input pin is defined by the following expression: $v_{dc}(t) = 2.5$ V $+ v_{ac}(t)$. The processed current signals (i_{1dc} , i_{2dc} , and i_{3dc}) are connected to the Arduino MEGA board analog inputs (A3, A4, and A5). Capacitors $C_1 = C_2 = C_3 = 4.7$ μ F are used for DC signal stabilization and noise filtering.

C. ADC input impedance considerations

The input source impedances of the conditioning circuits are critical parameters for ensuring the accuracy of the 10-bit SAR (Successive Approximation Register) ADC used by Arduino MEGA board. For the voltage conditioning circuits, the source impedance is $|Z_s| \approx 11$ k Ω , while for the current conditioning circuits, it is maintained at $|Z_s| < 1$ k Ω due to the low-resistance biasing network; these values ensure that the ADC's internal sample-and-hold capacitor reaches a steady-state charge within the allocated sampling window.

D. Limitations of the signal conditioning board

The current version of the signal conditioning board incorporates several design trade-offs to prioritize cost-effectiveness and modularity. Specifically, small encapsulated power transformers were utilized as an alternative to professional-grade potential transformers for voltage scaling. Furthermore, the passive components within the conditioning circuitry, primarily resistors and potentiometers, feature standard tolerances of 5 %. Regarding current acquisition, the performance of the SCT-013 series split-core current transformers have been previously analyzed and documented in [10]. Within the context of this initial design and implementation phase, these hardware limitations are considered acceptable, as the primary objective is the validation of the multi-channel acquisition architecture; a rigorous metrological calibration of the entire chain remains outside the scope of the present study.

III. DESIGN OF AN INTEGRATED DATA ACQUISITION AND PROCESSING PIPELINE

The functional architecture of the proposed signal acquisition system, based on Arduino–Python platform, and including the integrated data processing pipeline, is illustrated in Fig. 2. The proposed data processing pipeline is organized into four primary functional layers: (1) hardware acquisition, (2) data frame decoding and validation, (3) data transformation and buffering, and (4) integrated visualization, storage, and post-analysis. The Arduino Mega 2560 board manages the hardware acquisition, while the host PC is responsible for the remaining three layers.

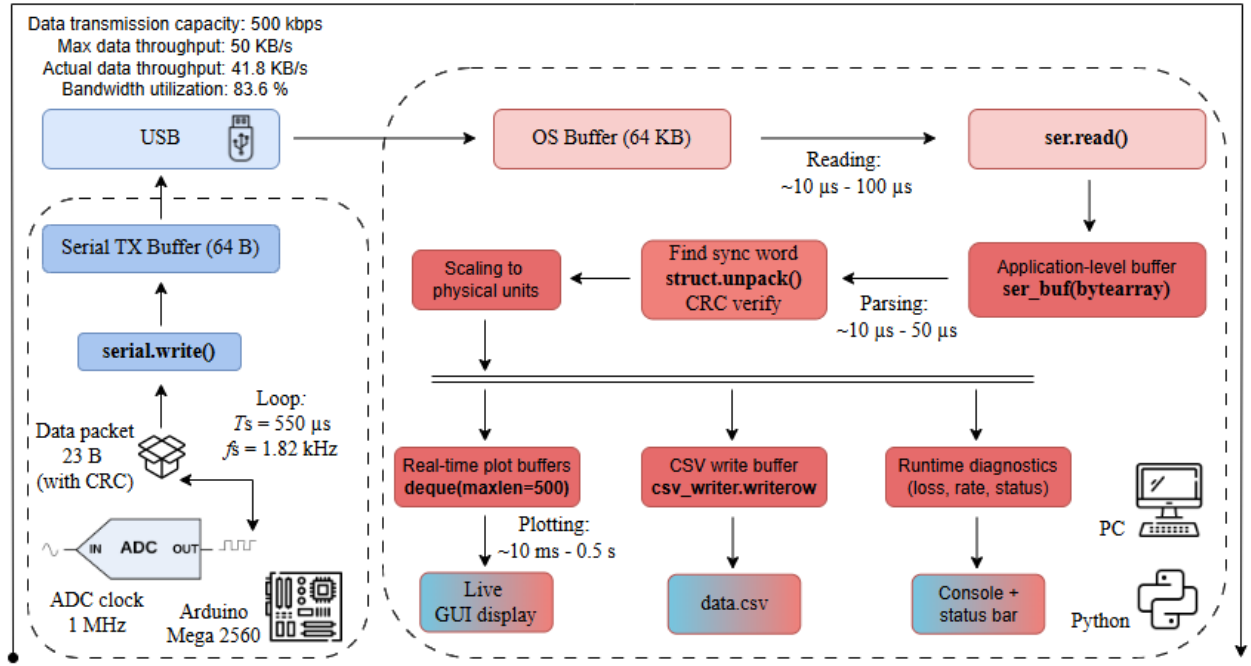


Fig. 2. Functional architecture of an integrated data acquisition and processing pipeline

A. Arduino-based acquisition framework

The acquisition system achieves a sampling frequency of $f_s = 1.82 \text{ kHz}$ by utilizing $T_s = 550 \mu s$ execution period (a single main program loop iteration time). The Arduino MEGA performs quasi-simultaneous acquisition of six analog signals (3 voltages + 3 currents) by executing a sequence of six successive analog-to-digital conversions within a single sampling window. Signal digitization is handled by an integrated 10-bit Successive Approximation Register (SAR) ADC utilizing a stable 5 V reference. To achieve a high sampling frequency while maintaining a multi-channel acquisition, the ADC clock is set to 1 MHz. While this exceeds the recommendation for maximum 10-bit resolution (200 kHz), it reduces the total conversion time for six channels to $81 \mu s$. This enables the system to remain well within the $550 \mu s$ program loop budget. However, this operational trade-off results in an increased Total Unadjusted Error (TUE), the magnitude of which is heavily dependent on the source impedance of the analog conditioning board and the corresponding settling time of the internal sample-and-hold capacitor. The worst case scenario TUE value is estimated to be less than $\pm 5 \text{ LSB}$ ($\pm 24.4 \text{ mV}$).

B. USB-based data transmission

To facilitate efficient serial communication, the acquisition system utilizes the compact binary data packet structure shown in Tab. 1. Field sync (2 B) is a unique marker for packet alignment and stream synchronization on the receiving side (PC). Field counter (4 B) enables detection of missing or out-of-order samples. Field timestamp (4 B) provides a temporal reference. The packet then contains six fields representing three-phase voltage (voltageL1, voltageL2, voltageL3) and current (currentL1, currentL2, currentL3) measurements (12 B), allowing efficient encoding of scaled analog signals. Finally, a crc field (1 B) is included for data integrity verification using a checksum mechanism. Total size of data packet is fixed to $L_p = 23 \text{ B}$.

TABLE I. DATA PACKET STRUCTURE AND FIELD SIZES

Variable (type)	Size (B)
sync (uint16_t)	2
counter (uint32_t)	4
timestamp (uint32_t)	4
voltageL1 (int16_t) voltageL2 (int16_t) voltageL3 (int16_t)	3 x 2
currentL1 (int16_t) currentL2 (int16_t) currentL3 (int16_t)	3 x 2
crc (uint8_t)	1
Total (all variables)	23

The 23-byte data packets are first staged in the 64-byte serial transmit (TX) ring buffer, where the hardware UART asynchronously manages the bit-stream generation for serial transmission. The communication interface was configured with a baud rate of 500 kbps, establishing a maximum theoretical throughput of $\eta_{th} = 50 \text{ KB/s}$ based on a standard 10-bit asynchronous serial frame. The deterministic sampling period ($T_s = 550 \mu s$) and the 23-byte packet size ($L_p = 23 \text{ B}$) establish an effective (actual) data throughput of:

$$\eta_{eff} = \frac{1}{T_s} \times L_p \approx 41.8 \text{ KB/s} \quad (1)$$

Given the effective throughput of 41.8 KB/s against a theoretical maximum of 50 KB/s, the system achieves a bandwidth utilization of 83.6%.

C. Python-based data processing

Upon transmission, the data is transmitted to the operating system's 64 KB serial buffer, providing a hardware abstraction layer that decouples the high-speed bitstream from the Python-

based acquisition loop. The Python application periodically retrieves all available bytes using the `ser.read()` function, operating in a non-blocking mode to avoid acquisition delays. The received data are then sent to an application-level byte buffer (`ser_buf`), which decouples serial data acquisition from packet processing. Packet parsing is subsequently performed on this buffer by searching for the synchronization word, extracting structured fields using `struct.unpack()`, and verifying packet integrity via a CRC check. The pure read operation typically requires approximately $10\ \mu\text{s} - 100\ \mu\text{s}$ depending on the amount of available data, while packet parsing, including synchronization search, unpacking, and validation, requires approximately $10\ \mu\text{s} - 50\ \mu\text{s}$.

The parsed integer values are then scaled to physical units corresponding to voltages and currents. This scaling stage compensates for the gain and offset introduced by the analog conditioning board, including voltage dividers, signal shifting, and sensor/transducer ratios. By applying appropriate conversion factors, the integer values of voltages and currents, $v_{\text{dec}}[nT_s]$ and $i_{\text{dec}}[nT_s]$, are transformed into physical values using the following equations:

$$v_m[nT_s] = C_{v\text{scale}} v_{\text{dec}}[nT_s] + C_{v\text{offset}}, \quad (2)$$

$$i_m[nT_s] = C_{i\text{scale}} i_{\text{dec}}[nT_s] + C_{i\text{offset}}, \quad (3)$$

where: $C_{v\text{scale}} = 1.042$ and $C_{v\text{offset}} = -534\ \text{V}$ are voltage scale and offset constants, respectively,

$C_{i\text{scale}} = 0.097$ and $C_{i\text{offset}} = -48.88\ \text{A}$ are current scale and offset constants, respectively.

Previous constants were derived through empirical observation of signal zeros and peak magnitudes.

After scaling, the processed samples are distributed to three output paths. First, near real-time visualization is implemented using rolling plot buffers based on `deque(maxlen = 500)`, which store the most recent samples for each signal. These buffers feed a live GUI display that supports basic control functionalities such as start, stop, and reset, enabling interactive monitoring of the acquired waveforms. The plotting operation is executed periodically and typically requires between $10\ \text{ms}$ and $0.5\ \text{s}$ depending on the rendering load. Second, the scaled data are written to a CSV file for persistent storage. To reduce disk, I/O overhead, samples are first accumulated in a CSV write buffer and then written using `csv_writer.writerow()`, with the recorded signals stored in a file named `data.csv`. Third, runtime diagnostics are continuously updated, including packet loss detection, effective sampling rate estimation, and system status monitoring. These diagnostic indicators are presented through console output together with a status bar, providing real-time feedback on acquisition performance and data integrity.

IV. LABORATORY TESTBED AND EXPERIMENTAL RESULTS

To provide a comprehensive overview of the experimental setup, the overall architecture of the proposed testbed is organized into three primary functional domains: (I) power section, (II) signal conditioning section, and (III) monitoring section (Arduino–Python platform). Functional block diagram of the experimental setup is shown in Fig. 3.

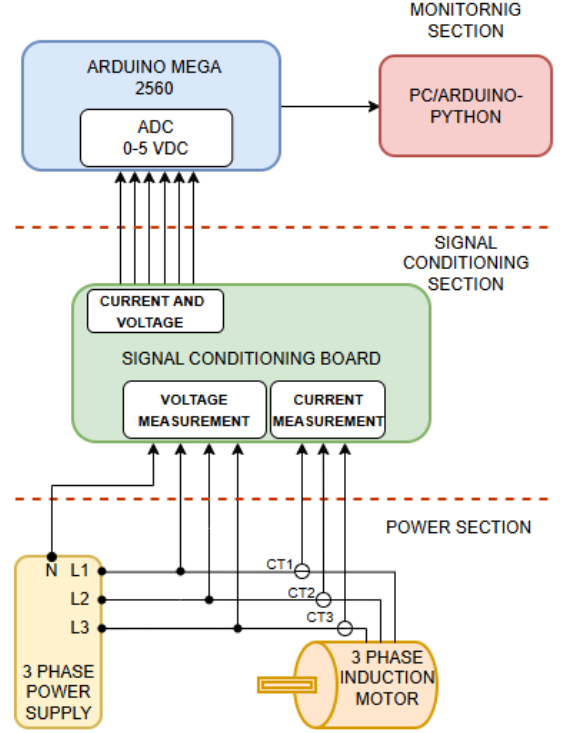


Fig. 3. Functional block diagram of the experimental setup: (I) power section, (II) signal conditioning section, and (III) monitoring section (Arduino–Python platform)

The power section (I) constitutes the primary energy path, consisting of the three-phase grid supply connected to the squirrel cage induction motor under test (model W3375-2M, manufacturer WUEKRO GmbH). The motor features a nominal power of $1.1\ \text{kW}$, a rated voltage of $230\ \text{V} / 400\ \text{V}$, and a nominal power factor of 0.81 . The experimental measurements were conducted while the induction motor was operating under no-load conditions.

The signal conditioning section (II) serves as the critical interface between the high-voltage power section and the low-voltage monitoring section. In this stage, the previously discussed voltage and current conditioning circuits are utilized to scale the three-phase voltage and current waveforms into the $0 - 5\ \text{V}$ range. This transformation is necessary to ensure compatibility with the ADC requirements, while simultaneously providing the galvanic isolation required for safe experimental operation.

Finally, the monitoring section (III) represents the core of the proposed platform, where the Arduino Mega 2560 performs 10-bit sampling of the six input analog signals (voltages and currents). This digitized data is streamed via USB-serial interface to a Python-based processing pipeline, which handles near real-time visualization and CSV logging.

The realization of the proposed platform and its integration within the test environment are presented in Fig. 4, which provides an overview of the experimental setup and laboratory arrangement. Siglent SDS 1102 CML+ digital storage oscilloscopes [11] were employed to track and verify the operation of the acquisition system in real-time.

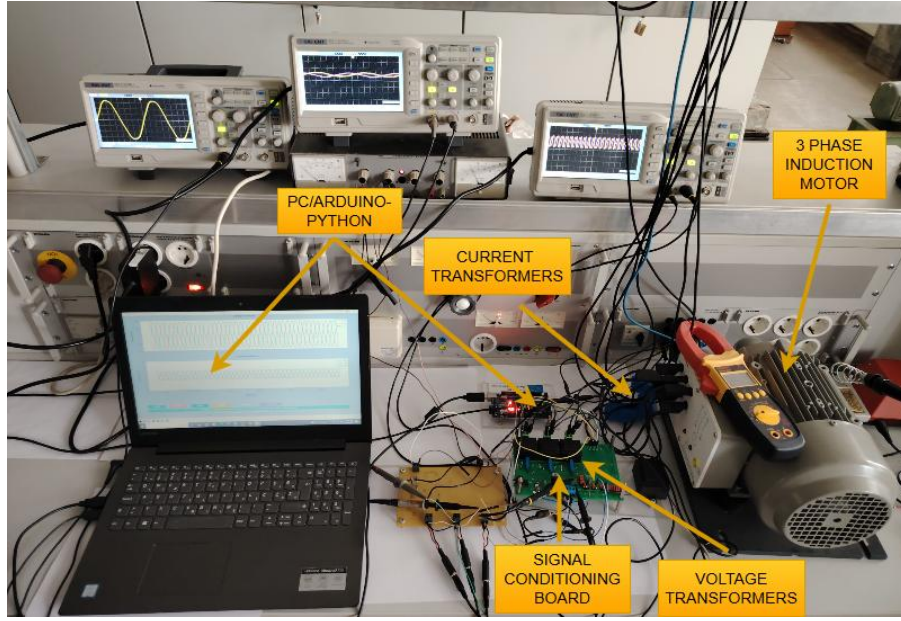


Fig. 4. Overview of the experimental setup and laboratory arrangement

A. Representative experimental results

To evaluate the operational performance of the developed experimental setup for near real-time signal acquisition, this section shows the data captured during motor operation. Fig. 5 displays the three-phase voltage and current signals during induction motor starting, visualized from post-processed CSV data.

Although a comprehensive metrological calibration was beyond the scope of this research, the acquisition system demonstrates a robust capacity for capturing and storing three-phase voltage and current signals during the induction motor's highly dynamic starting transient. The recorded waveforms align with expected theoretical profiles, maintaining signal integrity even throughout the transient period. Deviations from reference shapes are primarily attributed to the inherent design constraints of the low-cost signal conditioning board and the limitations of

the multichannel sampling frequency. In the steady-state period following the motor start, the system achieved relative amplitude errors of less than 5% for both voltage and current signals. To compensate for the absence of detailed calibration, empirical scaling formulas were implemented within the Python processing pipeline, ensuring that the acquired data remains sufficiently accurate for system behavior analysis and performance evaluation.

To further validate the reliability of the proposed platform, a comparative analysis was performed between the voltage signal recorded by the developed acquisition system and the one captured by digital oscilloscope. Fig. 6 illustrates the comparison of phase L1 voltage waveforms, contrasting (1) the experimentally acquired signal via the Arduino–Python platform with (2) the reference signal captured by the Siglent digital storage oscilloscope.

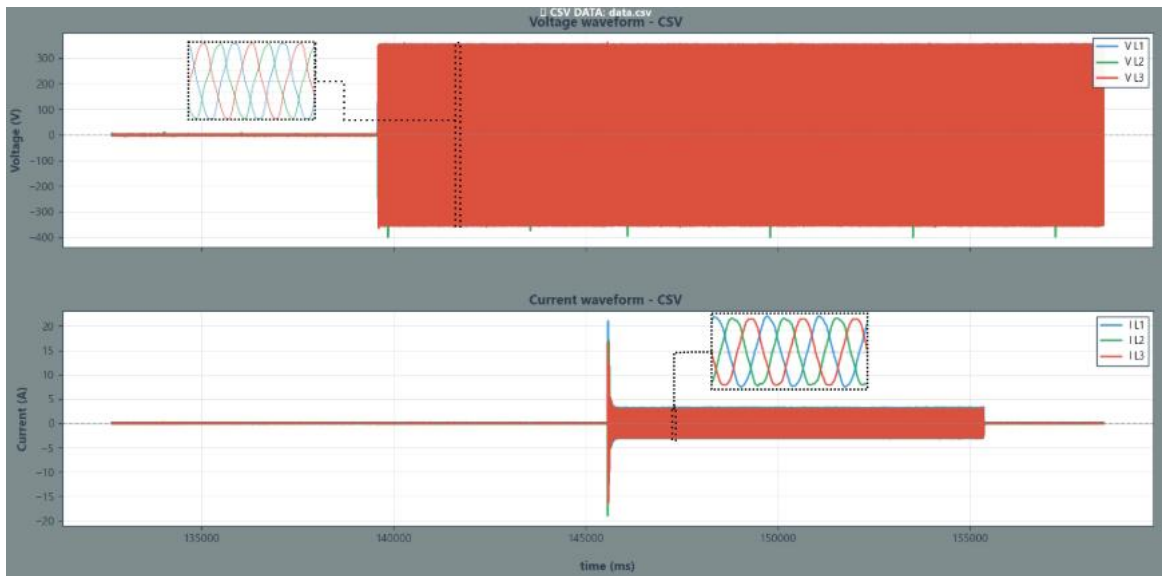


Fig. 5. Three-phase voltage and current signals during induction motor starting, visualized from post-processed CSV data

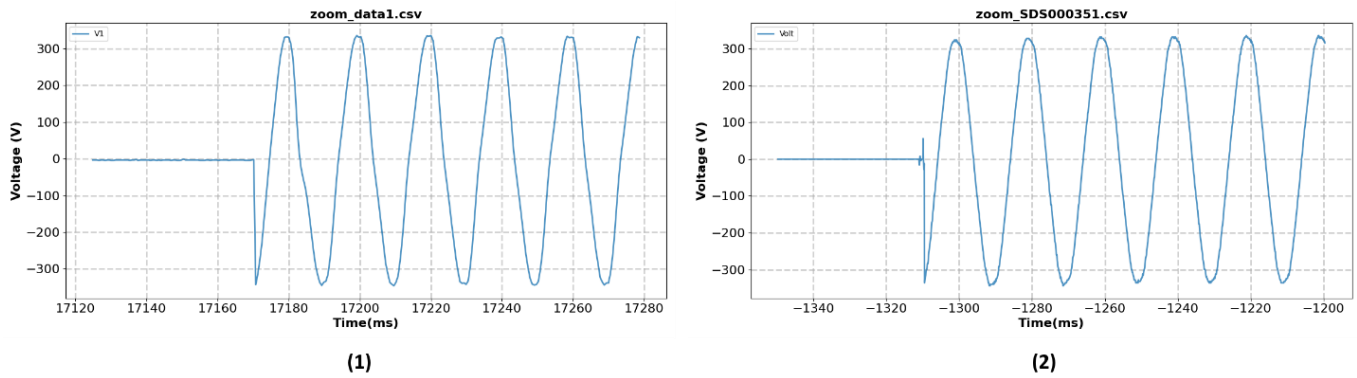


Fig. 6. Comparison of phase L1 voltage waveforms: (1) experimentally acquired signal via the Arduino–Python platform, and (2) reference signal captured by a digital oscilloscope

As demonstrated in the comparison, the waveforms and amplitudes align almost perfectly. While the oscilloscope utilizes a significantly higher sampling rate of 125 kHz, the proposed acquisition system maintains sufficient temporal resolution to accurately reconstruct the voltage waveform.

CONCLUSION

This article presented the design and implementation of an Arduino–Python platform for near real-time three-phase signal acquisition. The proposed architecture demonstrates that a robust and functional acquisition system can be achieved by leveraging the synergy between low-cost hardware and open-source software. A primary advantage of this platform is its high degree of scalability, allowing for the development of customized multichannel acquisition systems tailored to specific requirements regarding sampling frequency, signal count, and desired precision. Given its open-source nature and cost-effectiveness, the system is particularly well-suited for research and educational laboratories. While the current data acquisition system implementation proved effective, future work will be directed toward a comprehensive redesign of the signal conditioning board to further improve accuracy. Additionally, a detailed metrological calibration will be conducted to enhance the absolute accuracy of the system, transforming this initial prototype design into a high-precision tool for advanced electrical engineering applications.

REFERENCES

- [1] B. Mallala, R. Manyam, J. Saini *et al.*, “IoT enabled smart power quality analysis in three phase electrical systems with practical implementation,” *Scientific Reports*, vol. 15, p. 39665, 2025. <https://doi.org/10.1038/s41598-025-23317-z>
- [2] H. O. Garcés, J. Godoy, G. Rizzo, N. F. Sepúlveda, E. Espinosa, and M. A. Ahmed, “Development of an IoT-enabled smart electricity meter for real-time energy monitoring and efficiency,” *Electronics*, vol. 14, no. 6, p. 1173, 2025. <https://doi.org/10.3390/electronics14061173>
- [3] V. V. R. Raju, K. H. Phani Shree, and S. V. Jayarama Kumar, “Development of cost-effective phasor measurement unit for wide area monitoring system applications,” *International Journal of Electrical and Computer Engineering*, vol. 11, no. 6, pp. 4731–4739, 2021. <https://doi.org/10.11591/ijece.v11i6.pp4731-4739>
- [4] D. Medved, P. Šťastný “Electric power quality analyzer design using of opensource platform Arduino,” *Elektroenergetika*, vol. 9, no. 2, 2016. ISSN 1337-6756
- [5] A. D’Ausilio, “Arduino: A low-cost multipurpose lab equipment,” *Behavior Research Methods*, vol. 44, no. 2, pp. 305–313, Oct. 2012. <https://doi.org/10.3758/s13428-011-0163-z>
- [6] P. Pejović, “Application of Python programming language in measurements,” *Facta Universitatis, Series: Electronics and Energetics*, vol. 32, no. 1, pp. 1–23, Mar. 2019. <https://doi.org/10.2298/FUEE1901001P>
- [7] A. Khattak, M. M. Ahmad, and F. U. Khan, “Arduino based control and data acquisition system using Python graphical user interface GUI,” *International Journal of Scientific & Technology Research*, vol. 10, no. 6, June 2021. ISSN 2277-8616
- [8] https://hahn-trafo.com/Englische-Seiten/PDF-Englisch/Flyer/ttechnik/EI_30_Englisch.pdf
- [9] <https://www.alldatasheet.net/datasheet-pdf/pdf/1155092/YHDC/SCT-013-020.html>
- [10] V. Miron-Alexe, “Comparative study regarding measurements of different AC current sensors,” in *International Symposium on Fundamentals of Electrical Engineering (ISFEE)*, Bucharest, Romania, 2016, pp. 1–6. <https://doi.org/10.1109/ISFEE.2016.7803152>
- [11] <https://www.batronix.com/pdf/Siglent/SDS1000CML/SDS1000CML-Datasheet.pdf>