Scientific Papers. Series E. Land Reclamation, Earth Observation & Surveying, Environmental Engineering. Vol. XIV, 2025
Print ISSN 2285-6064, CD-ROM ISSN 2285-6072, Online ISSN 2393-5138, ISSN-L 2285-6064

AN ARDUINO-PYTHON INTEGRATED PREDICTIVE DATA ACQUISITION SYSTEM FOR ENVIRONMENTAL MONITORING

Georgian ROTARU, Gabriel PREDUȘCĂ,
Liana Denisa CÎRCIUMĂRESCU, Emil DIACONU

Valahia University of Targoviste, 13 Sinaia Alley Street, Targoviste, Romania

Corresponding author email: gabriel.predusca@valahia.ro

Abstract

This study presents a predictive data acquisition system that integrates the Arduino Leonardo development platform with the Python programming language for automated environmental monitoring. The system employs a suite of sensors - DHT22 for temperature and humidity, BMP180 for atmospheric pressure, and MQ-7 for gas concentration - to collect real-time data. These measurements are transmitted to a Python-based processing environment, where time-series forecasting algorithms, including ARIMA, are applied to analyze trends and predict environmental variations. The system offers a cost-effective and adaptable solution for air quality assessment, meteorological research, and environmental data analysis, facilitating timely interventions and enhanced resource management.

Key words: *Arduino Leonardo, BMP180, DHT22, MQ3, Python, environmental prediction.*

INTRODUCTION

The automated acquisition of data from sensors deployed in remote or challenging environments is essential for ensuring accuracy and efficiency in environmental monitoring. Automation reduces human error, improves data reliability, and streamlines the information-gathering process - especially in conditions where manual data collection is impractical.

Data acquisition systems (DAQ) play a crucial role in scientific research and engineering, providing solutions for testing, automation, and advanced analysis. These systems utilize various sensors and transducers to convert physical phenomena into measurable signals. Core functions of a DAQ system include signal conditioning, analog-to-digital conversion, and data processing, all of which contribute to a comprehensive understanding of environmental variables and support evidence-based decision-making (Maurizio, 2013).

Two critical components in the data acquisition process are data analysis and prediction. Data analysis enables the identification of patterns, correlations, and statistical relationships within collected datasets, forming the basis for reliable interpretation and trend recognition. Techniques such as descriptive statistics, correlation analysis, and probability distributions are applied to extract actionable insights.

Prediction, in contrast, focuses on estimating future values based on historical data. This is achieved through statistical and machine learning methods, which are used to construct models capable of anticipating environmental changes. These predictive techniques are instrumental in transforming raw data into practical forecasting tools.

Machine learning algorithms form a critical component of this predictive framework. They can be broadly categorized as follows:

- Regression – for forecasting continuous variables such as temperature or pressure.
- Classification – for categorizing environmental conditions (clear, cloudy, rainy).
- Unsupervised Learning – utilized to identify patterns and relationships in data without predefined labels.
- Semi-Supervised Learning – combining labeled and unlabeled data for improved prediction accuracy.

Recent advances in machine learning have significantly enhanced environmental monitoring by enabling automated analysis of complex datasets. In addition to machine learning, statistical methods are essential for predictive modeling, offering a solid theoretical basis for analyzing data relationships. Key statistical approaches include:

- Regression Analysis – used to estimate the relationships between variables, enabling the

prediction of one variable based on the behavior of others.
- Time Series Analysis – focuses on detecting patterns within temporally ordered data, making it especially effective for identifying trends and forecasting future values.
- Bayesian Methods – utilize probabilistic reasoning and prior knowledge to update predictions as new data becomes available, allowing for dynamic and adaptive forecasting (Siahaan et al., 2023).

Given the wide range of signals and parameters that can be sampled and stored, DAQ involves numerous techniques and skills. A DAQ system consists of various components, including sensors, communication links, signal processors, computers, databases, and data acquisition software. These components must function cohesively to ensure the system's reliability and effectiveness (Potter et al., 2012; Singh et al., 2009; Fisher et al., 2012; Sarma et al., 2018).

This study presents the design and evaluation of a predictive DAQ system built on the Arduino Leonardo platform and integrated with Python. The system incorporates key environmental sensors and leverages statistical forecasting methods to facilitate real-time environmental monitoring and predictive analytics. Applications include air quality assessment, meteorological research, and smart environmental management systems.

MATERIALS AND METHODS

Hardware component

Arduino is a popular open-source platform for rapid prototyping and data acquisition. This study uses the Arduino Leonardo, featuring the ATmega32u4 microcontroller with built-in USB communication for efficient data transfer (Arduino Leonardo, 2025; Monk, 2022).

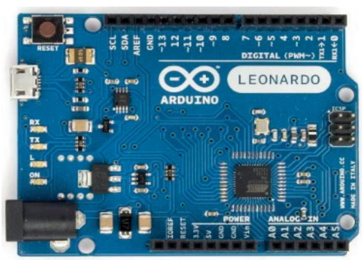

Figure 1. Arduino Leonardo development platform (Arduino Leonardo, 2025)

The Arduino Leonardo board includes 19 digital and 12 analog input/output ports, facilitating compatibility with a wide array of sensors and external modules. This versatility makes it particularly well-suited for customizable applications, such as predictive environmental monitoring.

The board operates at a nominal voltage of 5V, with each I/O pin capable of sourcing or sinking up to 40 mA of current. It features an internal pull-up resistor (disabled by default), which provides enhanced control during circuit design. Additionally, it includes 32KB of flash memory, 2.5KB of SRAM, and runs at a clock speed of 16 MHz.

In the developed system, the ATmega32u4 microcontroller functions as the central unit, responsible for sensor data acquisition and transmission. Its ability to handle multiple I/O operations and communicate via USB makes it an ideal solution for integration with the Python-based processing environment.

Sensor specifications

The proposed system integrates three types of sensors to monitor key environmental parameters – temperature and humidity, atmospheric pressure, and gas concentration.

The **DHT22** sensor (Figure 2, Table 1) provides accurate digital measurements of temperature and relative humidity. It employs proprietary signal acquisition and humidity sensing technology, ensuring both high precision and long-term operational stability. The sensor outputs data into a calibrated digital format, simplifying its integration with microcontroller-based systems.

Figure 2. DHT22 sensor module used for temperature and humidity measurement (DHT22, 2025)

The **BMP180** sensor (Figure 3, Table 1) adds atmospheric pressure monitoring capability to the data acquisition system. It combines a high-resolution barometric pressure sensor with a

temperature sensor, making it suitable for weather-related applications.
As the successor to BMP085, the BMP180 is optimized for accuracy and energy efficiency, and is commonly used in consumer-grade environmental monitoring devices.

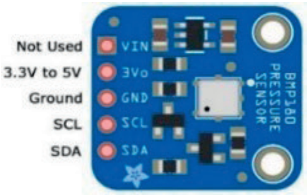

Figure 3. BMP180 sensor module used for atmospheric pressure measurement (BMP180, 2025)

The **MQ-7** sensor (Figure 4, Table 1) is employed for detecting gas concentrations, particularly carbon monoxide (CO) and hydrogen (H_2). It plays a critical role in evaluating air quality, contributing valuable data to the system's predictive analytics. However, it is sensitive to environmental noise and requires a preheating phase to stabilize reading.

Figure 4. MQ-7 sensor module used for gas concentration monitoring (MQ7, 2025)

Table 1. Technical characteristics of sensors

Sensor	DHT22	BMP180	MQ7
Measured parameters	Temperature, humidity	Atmospheric pressure	Gas concentration (CO, H_2)
Accuracy	±0.50°C ±2% RH	±0.12 hPa	Varies by gas
Communication interface	Digital (single wire)	I2C	Analog via ADC
Operating voltage	3.3-6V	1.62-3.6V	5V
Additional notes	Calibrated output, 2s reading interval	Include EEPROM for calibration; temperature compensation	Requires preheating; sensitive to environmenttal noise

Software tools and programming

The Arduino Leonardo microcontroller is programmed using the Arduino language via the Arduino Integrated Development Environment (IDE). This platform allows for the development of custom scripts to control sensor operations, manage collected data, and transmit information to a Python-based environment for further analysis.
Python serves as the core platform for data processing in this study, selected for its readability, versatility, and robust ecosystem of libraries. Its object-oriented structure and extensive community support make it well-suited for implementing complex data analysis and predictive modeling tasks (Fuentes, 2018).

Data processing and prediction pipeline

Figure 5 illustrates the end-to-end data processing workflow of the developed system.
Environmental data from DHT22, BMP180, and MQ-7 sensors is first displayed on the Arduino IDE Serial Monitor. Using Python's serial communication libraries, the data is captured and organized into a Pandas DataFrame for efficient handling. The processed data is then exported to Excel for storage and traceability.

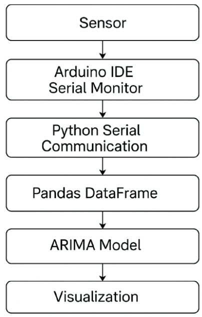

Figure 5. Data processing flow for DAQ system

For predictive analysis, the system applies an ARIMA (AutoRegressive Integrated Moving Average) model to the time-series data, enabling the forecasting of future environmental conditions. Results are visualized through Python-based plotting libraries, providing insights into data trends and supporting real-time, informed decision-making.

RESULTS AND DISCUSSIONS

System implementations

The proposed system utilizes the Arduino Leonardo platform (Figure 6) to acquire data from three environmental sensors: the BMP180 (atmospheric pressure), the DHT22 (temperature and humidity), and the MQ-2 (gas concentration). Sensor readings are initially displayed on the Arduino IDE Serial Monitor, enabling real-time data visualization.

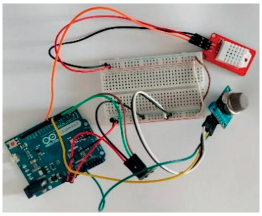

Figure 6. System implementation using Arduino and environmental sensors

Data Collection and Visualization

Sensor data is retrieved via Python using serial communication, structured into data frames, and stored in Excel files for traceability and further analysis. Figures 7 to 10 present graphical outputs generated using the Arduino Plotter, providing a visual overview of temperature, humidity, gas levels, and atmospheric pressure as measured in real time.

This data forms the basis for analyzing environmental conditions and identifying trends over time - supporting applications in air quality monitoring, precision agriculture, and meteorological research.

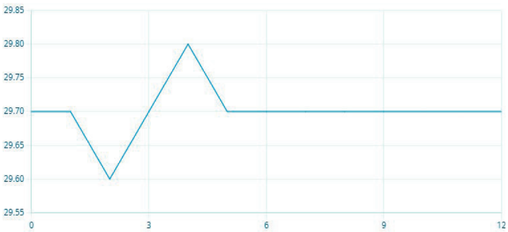

Figure 7. Temperature readings from DHT22 in Arduino Plotter

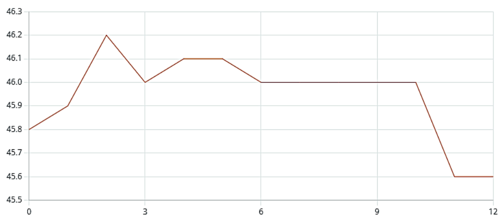

Figure 8. Humidity readings from DHT22 in Arduino Plotter

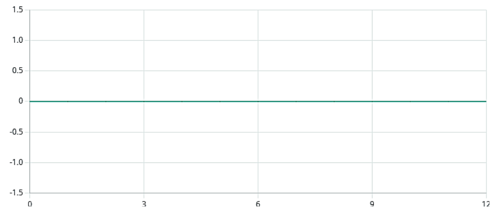

Figure 9. Gas concentration readings from MQ-7 in Arduino Plotter

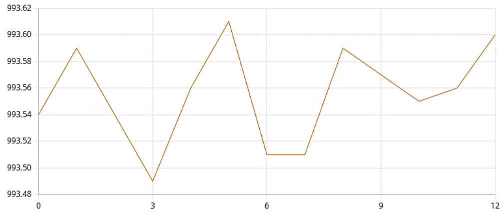

Figure 10. Atmospheric pressure readings from BMP180 in Arduino Plotter

Python's extensive ecosystem enhances the system's data processing and modeling capabilities. Key libraries include:

- Pandas – efficient for managing tabular data, offering tools for filtering, aggregation, and transformation.
- Matplotlib and Seaborn – visualization libraries that support the creation of insightful and interpretable charts.
- Scikit-learn – a machine learning toolkit providing access to models such as linear regression, decision trees, and more.

These tools collectively empower the system to bridge environmental sensing and predictive analytics. To support this, two Python programs were developed - one for data acquisition and formatting, and another for predictive modeling - each playing a vital role in the end-to-end analysis pipeline.

Predictive Modeling with ARIMA

To enable predictive analysis, two Python programs were developed - each with a distinct role in the data acquisition and forecasting pipeline.

In the first program, Python libraries such as serial and pandas were employed to manage communication between the Arduino and the host computer. The serial library facilitated data extraction from the Arduino sensors, while pandas enabled efficient organization, manipulation, and export of the data to Excel format.

This framework ensured clean, structured datasets for further processing.

The second program focused on time-series forecasting using the ARIMA (AutoRegressive Integrated Moving Average) model.

Libraries used included pandas for data handling, statsmodels.tsa.ARIMA.model for implementing the ARIMA algorithm, matplotlib.pyplot for visualization, and NumPy for numerical operations.

ARIMA, a well-established statistical technique, leverages past observations to model and predict future values in a time series - an approach particularly useful in environmental monitoring. An ARIMA model is defined by three parameters:

- p: the autoregressive order (number of lagged observations),
- d: the degree of difference (used to make the data stationary),
- q: the moving average order (size of the error window used for smoothing).

Fine-tuning these parameters allows for the creation of an optimized forecasting model tailored to the nature of the dataset (Wicaksana et al., 2022; Herrera-Gonzalez et al., 2024; Ganghetty et al., 2021; Spyrou et al., 2022; Kulkarni, et al., 2023).

ARIMA models can be adapted into simpler forms - such as AR (AutoRegressive), MA (Moving Average), or ARMA - by setting one or more parameters to zero. While ARIMA assumes that past values influence future trends, it is important to recognize its limitations, particularly in cases where external, non-recurring factors influence environmental conditions (Tibshirani, 2023). To assess the model's predictive accuracy, we evaluated three commonly used performance metrics:

- RMSE (Root Mean Square Error)
- MAE (Mean Absolute Error)
- MAPE (Mean Absolute Percentage Error)

For example, in forecasting humidity using the ARIMA(5,1,0) configuration, the results showed a marked improvement when data was sampled more frequently:

- Case 1 (one reading/day): RMSE = 1.34, MAE = 1.09, MAPE = 2.06%
- Case 2 (three readings/day): RMSE = 0.92, MAE = 0.71, MAPE = 1.28%

These results demonstrate that increasing the sampling frequency significantly enhances forecasting accuracy. The findings underscore the importance of data resolution in predictive modeling, particularly in dynamic environmental contexts.

Table 2. Values for the once-per-day reading frequency

Day	DHT22H (%)	DHT22T (°C)	MQ-7 (ppm)	BMP180 (hPa)
1	59.5	9.4	21	987.41
2	49.6	7.9	0	989.76
3	48.5	9.8	0	988.34
4	52.3	10.1	0	987.98
5	54.3	10.4	3	988.85
6	50.4	10.6	0	987.43
7	52.8	11.2	0	989.32
8	51.5	11.3	0	988.32
9	49.7	11.8	2	988.54
10	53.7	11.9	0	989.53
11	52.4	11.4	0	987.98
12	51.8	12.1	0	988.93
13	50.4	12.3	0	987.78
14	53.2	12.4	0	985.32

Comparative Analysis - Case 1

Table 2 presents the sensor data collected with a sampling frequency of one reading per day.

Notably, the MQ-7 sensor - designed to detect flammable gases such as carbon monoxide (CO) and hydrogen (H_2) - is also sensitive to other environmental variables. This high sensitivity may result in considerable fluctuations in the recorded values, as observed in the dataset.

These unexpected variations in reading highlight the need for careful analysis and interpretation to accurately assess the system's performance and

the environmental conditions affecting the MQ-7 sensor. The corresponding analysis results are illustrated in Figures 11 to 14.
Eight ARIMA model configurations were tested by varying the parameters p and d, as shown in Table 3. Among these, the model with parameters ARIMA (scaled_data, order = (5,1,0)) yielded the best results, producing predicted values that closely matched the actual sensor readings presented in Table 2.

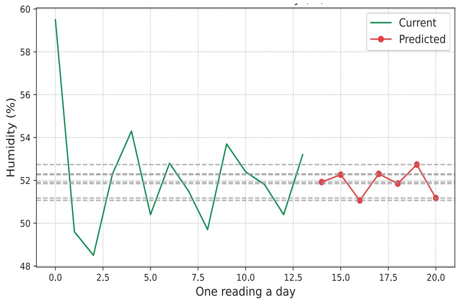

Figure 11. The graph and humidity prediction using DHT22

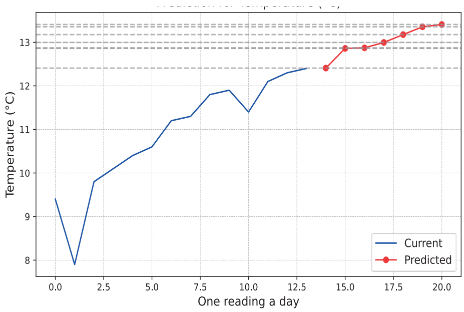

Figure 12. The graph and temperature prediction using DHT22

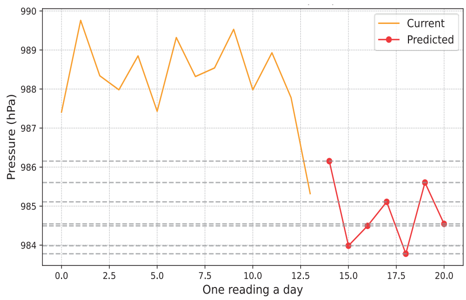

Figure 13. The graph and atmospheric pressure prediction using BMP180

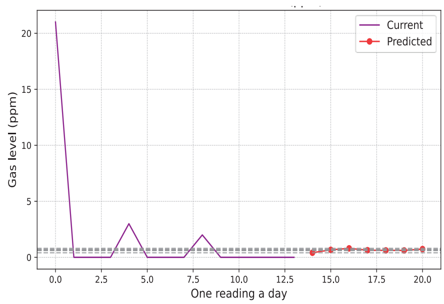

Figure 14. The graph and gas level prediction in the atmosphere using MQ-7

The comparative analysis indicates that the MQ-7 sensor is highly sensitive to environmental changes. This is reflected in its output, which often fluctuates from baseline values (typically zero) to peaks between 2 and 21 ppm in the measured data (Table 2), and from 0.13 to 6.96 ppm in the predicted data (Table 3). Such sensitivity underscores the need for precise calibration and controlled measurement conditions to ensure reliable predictions.

Table 3. Testing the ARIMA model for data acquisition once a day for 7 days

Cases	Sensor	Day						
		1	2	3	4	5	6	7
(3,2,0)	DHT22H	52.05	51.16	51.66	51.69	51.60	51.39	52.13
	DHT22T	12.63	13.07	13.05	13.42	13.66	13.86	14.07
	BMP180	985.01	982.01	980.93	979.33	975.95	975.59	972.25
	MQ7	0	0	0	0	0	0	0
(4,2,0)	DHT22H	52.79	51.30	51.85	53.09	52.55	51.79	52.50
	DHT22T	12.46	12.94	12.98	13.17	13.42	13.65	13.77
	BMP180	985.35	982.15	981.29	980.06	977.22	977.26	974.51
	MQ7	0.73	1.74	3	3.97	4.96	5.83	6.96
(5,2,0)	DHT22H	52.24	52.29	51.22	52.66	52.24	52.71	51.68
	DHT22T	12.48	12.77	12.88	13.17	13.23	13.47	13.64
	BMP180	985.32	982.15	981.18	979.83	976.83	976.66	973.76
	MQ7	0.21	0.71	1.35	1.98	2.53	3.06	3.69
(6,2,0)	DHT22H	52.29	52.03	52.19	51.64	52.36	52.87	52.79
	DHT22T	12.49	12.59	12.69	12.79	12.89	12.99	13.09
	BMP180	982.86	980.40	977.94	975.48	973.02	970.56	968.10
	MQ7	0.003	0.006	0.008	0.011	0.014	0.017	0.020

(3,1,0)	DHT22H	52.15	51.01	51.68	52.85	51.57	51.24	52.29
	DHT22T	12.72	12.69	12.86	12.90	12.93	13.01	13.00
	BMP180	987.11	984.43	984.66	985.06	982.86	984.43	983.14
	MQ7	0	0	0	0	0	0	0
(4,1,0)	DHT22H	52.82	50.69	51.71	52.99	52.04	51.10	52.11
	DHT22T	12.53	12.95	12.87	13.12	13.27	13.40	13.46
	BMP180	986.33	984.02	984.58	985.28	984.01	986.12	985.18
	MQ7	0	0	0	0	0	0	0
(5,1,0)	**DHT22H**	51.92	52.26	51.06	52.30	51.84	52.73	51.17
	DHT22T	12.40	12.86	12.87	12.99	13.17	13.35	13.40
	BMP180	986.15	983.98	984.49	985.10	983.77	985.60	984.54
	MQ7	0.40	0.66	0.80	0.65	0.62	0.61	0.73
(6,1,0)	DHT22H	51.49	52.33	52.02	51.33	51.44	52.03	52.30
	DHT22T	12.44	12.72	12.80	13.06	13.08	13.28	13.40
	BMP180	986.16	984.121	984.54	985.17	983.90	985.75	984.89
	MQ7	0	0.13	0.27	0.41	0.30	0.30	0.26

Comparative Analysis - Case 2

Table 4 shows sensor data collected at a higher frequency - three readings per day. Compared to Case 1, the values display greater variability, especially for the MQ-7 sensor.

Table 4. Data acquisition with a reading frequency of three times a day

Day	DHT22H (%)	DHT22T (°C)	MQ-7 (ppm)	BMP180 (hPa)
1	59.5	9.3	18	987.43
	65.5	8.6	3	987.47
	48.9	8	0	989.75
2	49.5	7.9	0	988.67
	54.9	6.2	0	989.95
	51.4	7.5	0	989.89
3	49.7	9.8	0	988.54
	53.5	9.9	0	988.12
	54.6	10	2	989.68
4	57.2	10.1	1	989.26
	54.3	10.3	0	987.93
	59	10.5	0	988.77
5	50.1	10.4	0	989.11
	59.1	10.5	2	987.85
	52.7	10.6	0	988.29
6	56.1	10.8	0	987.77
	55.7	10.9	0	989.39
	58.1	11	0	989.02
7	49.4	11.1	0	989.53
	52.4	11.2	0	988.64
	59.5	11.3	0	987.7
8	52.9	11.5	2	987.42
	57.5	11.4	3	988.86
	50.9	11.5	0	987.67
9	58.4	11.6	0	988.06
	52.3	11.7	0	989.82
	50.8	11.8	0	989.7
10	59.4	11.9	0	989.16
	56.7	12	0	989.84
	53.7	12.1	0	987.54
11	51.9	12.2	1	989.35
	50.3	12.3	0	987.64
	52.2	12.4	0	989.57
12	59.4	12.5	0	988.49
	51.4	12.6	0	987.98
	54	12.7	3	988.43
13	49.9	12.8	0	989.88
	59.2	12.9	0	988.74
	55.3	13	0	989.04
14	58.8	13.1	0	988.21
	54.7	13.2	0	989.24
	56.3	13.3	0	988.45

Predicted trends for humidity, temperature, gas concentration, and pressure are shown in Figures 15 to 18, demonstrating improved accuracy with increased sampling.

The most accurate results were obtained using the ARIMA model with the configuration ARIMA(scaled_data, order = (4,2,0)), based on multiple test iterations.

The outcomes of these tests are summarized in Table 5.

The results shown in Figures 11–14 and Tables 2-3 correspond to Case 1 (one reading per day), while Figures 15-18 and Tables 4-5 represent Case 2 (three readings per day).

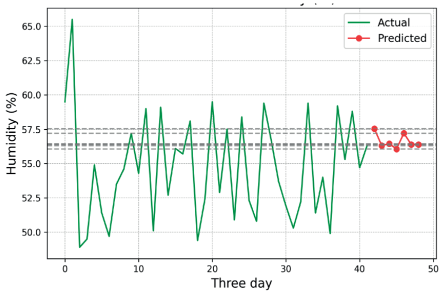

Figure 15. The graph and humidity prediction using DHT22

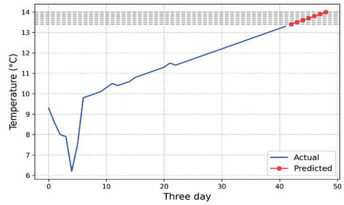

Figure 16. The graph and temperature using DHT22

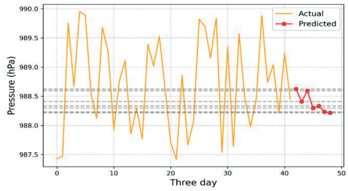

Figure 17. The graph and atmospheric pressure prediction using BMP180

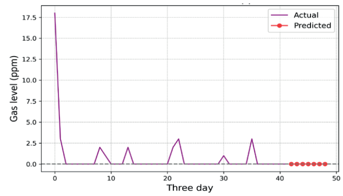

Figure 18. The graph and gas level prediction in the atmosphere using MQ-7

Table 5. Testing the Arima model for data acquisition three times a day

Cases	Days	DHT22H	DHT22 T	BMP 180	MQ7
(3,2,0)	1	52.05	12.63	985.01	0
	2	51.16	13.07	982.01	0
	3	51.66	13.05	980.93	0
	4	51.69	13.42	979.33	0
	5	51.60	13.66	975.95	0
	6	51.39	13.86	975.59	0
	7	52.13	14.07	972.25	0
(4,2,0)	1	52.79	12.46	985.35	0.73
	2	51.30	12.94	982.15	1.74
	3	51.85	12.98	981.29	3
	4	53.09	13.17	980.06	3.97
	5	52.55	13.42	977.22	4.96
	6	51.79	13.65	977.26	5.83
	7	52.50	13.77	974.51	6.96
(5,2,0)	1	52.24	12.48	985.32	0.21
	2	52.29	12.77	982.15	0.71
	3	51.22	12.88	981.18	1.35
	4	52.66	13.17	979.83	1.98
	5	52.24	13.23	976.83	2.53
	6	52.71	13.47	976.66	3.06
	7	51.68	13.64	973.76	3.69
(6,2,0)	1	52.29	12.49	982.86	0.003
	2	52.03	12.59	980.40	0.006
	3	52.19	12.69	977.94	0.008
	4	51.64	12.79	975.48	0.011
	5	52.36	12.89	973.02	0.014
	6	52.87	12.99	970.56	0.017
	7	52.79	13.09	968.10	0.020

(3,1,0)	1	52.15	12.72	987.11	0
	2	51.01	12.69	984.43	0
	3	51.68	12.86	984.66	0
	4	52.85	12.90	985.06	0
	5	51.57	12.93	982.86	0
	6	51.24	13.01	984.43	0
	7	52.29	13.00	983.14	0
(4,1,0)	1	52.82	12.53	986.33	0
	2	50.69	12.95	984.02	0
	3	51.71	12.87	984.58	0
	4	52.99	13.12	985.28	0
	5	52.04	13.27	984.01	0
	6	51.10	13.40	986.12	0
	7	52.11	13.46	985.18	0
(5,1,0)	1	51.92	12.40	986.15	0.40
	2	52.26	12.86	983.98	0.66
	3	51.06	12.87	984.49	0.80
	4	52.30	12.99	985.10	0.65
	5	51.84	13.17	983.77	0.62
	6	52.73	13.35	985.60	0.61
	7	51.17	13.40	984.54	0.73
(6,1,0)	1	51.49	12.44	986.16	0
	2	52.33	12.72	984.121	0.13
	3	52.02	12.80	984.54	0.27
	4	51.33	13.06	985.17	0.41
	5	51.44	13.08	983.90	0.30
	6	52.03	13.28	985.75	0.30
	7	52.30	13.40	984.89	0.26

To illustrate why Case 2 yields better performance, we compare measured and predicted values for each sensor.

For the DHT22H sensor (humidity):

- Case 1: Measured values ranged from 48.5% to 59.5% (Table 2), with predicted values between 51.06% and 52.73% (Figure 11, Table 3).
- Case 2: Measured values ranged from 48.9% to 65.5% (Table 4), while predictions ranged from 56.06% to 57.54% (Figure 15, Table 5).

In both cases, predictions tend to cluster around the mean, underestimating peak values. However, Case 2 shows better alignment with the actual trend, thanks to the increased sampling frequency.

Table 6 summarizes the average measured and predicted humidity values for both cases.

For the DHT22T sensor (air temperature):

- In Case 1, measured values ranged from 7.9°C to 12.4°C (Table 2), while predicted values ranged from 12.4°C to 13.4°C (Figure 12, Table 3).
- In Case 2, measured temperatures varied from 6.2°C to 13.3°C (Table 4), with predictions between 13.4°C and 14.0°C (Figure 16, Table 5).

In both cases, the predicted values exceed the actual measurements, indicating a consistent overestimation. This effect is more pronounced in Case 1, where predictions are centered around a higher average. While Case 2 shows improved responsiveness due to more frequent sampling, the predicted values still reflect a bias toward higher temperatures. The maximum measured temperature in Case 2 was 13.3°C, compared to a predicted peak of 14.0°C.

Table 7 provides a comparison of average measured and predicted temperature values for both cases.

For the MQ-7 sensor (gas concentration):

- In Case 1, measured values ranged from 0 to 21 ppm (Table 2), while predicted values remained between 0.4 and 0.8 ppm (Figure 13, Table 3).
- In Case 2, measured values ranged from 0 to 18 ppm (Table 4), whereas predicted values remained fixed at 0 ppm (Figure 17, Table 5).

In Case 1, although the predicted values generally follow the average trend, they fail to capture peak variations, indicating a degree of underfitting. In Case 2, despite the increased sampling frequency, the model consistently predicts 0 ppm, failing to reflect actual fluctuations. This persistent underestimation may result from sensor calibration issues or excessive environmental noise sensitivity.

Table 8 compares the average measured and predicted gas concentrations for both cases.

Table 6. Comparison of results for DHT22H sensor (air humidity)

Cases	Measured value			Predicted value		
	Min	max	average	min	max	average
1	48.5	59.5	52.15	51	52.7	51.89
2	48.9	65.5	54.69	56	57.5	56.62

Table 7. Comparison of results For DHT22T sensor (air temperature)

Cases	Measured value			Predicted value		
	Min	max	average	min	max	average
Case 1	7.9	12.4	10.9	12.4	13.4	13
Case 2	6.2	13.3	11.05	13.4	14	13.7

Table 8. Comparison of results for MQ-7 sensor (gas level in the atmosphere)

Cases	Measured value			Predicted value		
	Min	max	average	min	max	average
Case 1	0	21	1.87	0.4	0.8	0.63
Case 2	0	18	0.83	0	0	0

For the BMP180 sensor (atmospheric pressure):

- In Case 1, measured values ranged from 985.32 to 989.76 hPa (Table 2), while predicted values ranged from 983.77 to 986.15 hPa (Figure 14, Table 3).
- In Case 2, measured values ranged from 987.42 to 989.95 hPa (Table 4), with predicted values between 988.22 and 988.62 hPa (Figure 18, Table 5).

In Case 1, predicted values were relatively close to actual measurements but tended to underestimate the pressure, clustering around a lower average. In Case 2, with more frequent sampling, the predicted values showed improved alignment, though the model still slightly underpredicted the peak pressure (988.62 hPa vs. the actual 989.95 hPa).

A comparison of prediction performance between the two sampling intervals reveals notable improvements:

- Humidity (DHT22): RMSE reduced by ~23%
- Pressure (BMP180): RMSE reduced by ~19%

These reductions highlight the benefits of increased measurement frequency for enhancing prediction accuracy.

Table 9 summarizes the average measured and predicted pressure values for both cases.

Table 9. Comparison of results for BMP180 sensor (atmospheric pressure)

Cases	Measured value			Predicted value		
	Min	max	average	min	max	average
1	985.32	989.76	988.24	983.77	986.15	984.80
2	987.42	989.96	988.71	988.22	988.62	988.38

CONCLUSIONS

The proposed system successfully achieved its primary objective: developing an integrated, low-cost solution for environmental data acquisition and predictive analysis with a satisfactory level of accuracy. Among the ARIMA configurations tested, the ARIMA(5,1,0) model yielded the most reliable results, particularly in forecasting humidity and temperature trends.

Regarding sensor performance, the DHT22 sensor demonstrated high reliability for both temperature and humidity measurements, while the BMP180 sensor provided consistent and accurate atmospheric pressure readings. Conversely, the MQ-7 gas sensor exhibited considerable variability, which negatively impacted the accuracy of gas concentration predictions. In both sampling scenarios, predicted gas values were either consistently zero or significantly underestimated, suggesting potential issues related to sensor calibration or sensitivity to environmental noise.

Despite this limitation, the system establishes a solid foundation for real-time environmental monitoring and predictive analytics. Future improvements should focus on refining gas sensor calibration and enhancing the robustness of predictive models for volatile environmental parameters.

ACKNOWLEDGEMENTS

This article was written as part of the authors' activities within the Center for Networking and Career Development in Research in the South-Muntenia Region (CNDCSM), under the project PNRR-III-C9-2022 – I10, contract no. 5/16.11.2022, carried out at Valahia University of Targoviste.

REFERENCES

Aosong Electronics Co., Ltd. (2025). *DHT22 temperature and humidity sensor datasheet*. SparkFun Electronics. [Online], available: https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf, accessed on February 22, 2025.

Arduino. (2025). *Arduino Leonardo pinouts*. Arduino Documentation. [Online], available: https://docs.arduino.cc/resources/pinouts/A000057-full-pinout.pdf, accessed on February 22, 2025.

Bosch Sensortec. (2025). *BMP180 digital pressure sensor*. [Online], available: https://pdf1.alldatasheet.com/datasheet-pdf/view/1132068/BOSCH/BMP180.html accessed on February 22, 2025.

Fisher, D. K., & Gould, P. J. (2012). Open-source hardware is a low-cost alternative for scientific instrumentation and research. *Modern Instrumentation*, 1, 8–20. https://doi.org/10.4236/mi.2012.12002

Fuentes, A. (2018). *Hands-on predictive analytics with Python*. Packt Publishing.

Ganghetty, A., Kaur, G., & Malunje, U. S. (2021). Time series prediction of temperature in Pune using seasonal ARIMA model. *International Journal of Engineering Research & Technology*, *10*(11). https://doi.org/10.17577/IJERTV10IS110075

Herrera-Gonzalez, J. L., et al. (2024). Time series (ARIMA) as a tool to predict the temperature-humidity index in the dairy region of the northern desert of Mexico. *Peer J. Environmental Science*, *1*, 1–13. https://doi.org/10.7717/peerj.18744

Kulkarni, S., Bhadane, A., & Gajjar, H. (2023). Raspberry Pi-based weather prediction and analysis of temperature and humidity using ARIMA model. *BioGecko: A Journal for New Zealand Herpetology*, *12*(3), 5525–5533.

Maurizio, D. P. E. (2013). *Data acquisition systems: From fundamentals to applied design*. Springer. DOI:10.1007/978-1-4614-4214-1

Monk, S. (2022). *Programming Arduino: Getting started with sketches* (3rd ed.). McGraw Hill.

Potter, D., & Eren, H. (2012). Instrumentation engineers handbook: Process software and digital networks (Vol. 3, 4th ed., Chapter 19). CRC Press.

Sarma, P., Singh, H. K., & Bezboruah, T. (2018). A real-time data acquisition system for monitoring sensor data. *International Journal of Computer Sciences and Engineering*, *6*, 539–542. https://doi.org/10.26438/ijcse/v6i6.539542

Siahaan, V., & Sianipar, R. H. (2023). *Time-series weather: Forecasting and prediction with Python*. Balige City, North Sumatera.

Singh, H. K., Gogoi, R., & Bezboruah, T. (2009). Design approach for a web-based data acquisition and control system. In *International Conference of Internet Computing* (pp. 16–19).

SparkFun Electronics. (2025). *Technical data: MQ-7 gas sensor*. [Online], available: https://www.sparkfun.com/datasheets/Sensors/Biometric/MQ-7.pdf, accessed on February 22, 2025.

Spyrou, E. D., Tsoulos, I., & Stylios, C. (2022). Applying and comparing LSTM and ARIMA to predict CO levels for a time-series measurements in a port area. *Signals*, *3*(2), 235–248. https://doi.org/10.3390/signals3020015

Tibshirani, R. (2023). Autoregressive integrated moving average models. In *Introduction to time series*. University of California, Berkeley.

Wicaksana, H., Ananda, N., Budiawan, I., Santoso, B., Handoko, R., Maulana, A., Suciarti, S., & Utoro, A. (2022). Air temperature sensor estimation on automatic weather station using ARIMA and MLP. *Ultimatics: Jurnal Teknik Informatika,* *14*(2), 103–110. https://doi.org/10.31937/ti.v14i2.2865