Pol J Aviat Med Bioeng Psychol 2022; 28(3): 32-40
DOI: 10.13174/pjambp.30.06.2025.04

RECORDING OF ENVIRONMENTAL PARAMETERS USING A WIRELESS SENSOR NETWORK

Mariusz KREJ[1], Marcin PIOTROWSKI[2], Mirosław DEREŃ[1]

1 Department of Psychophysiological Measurements and Human Factor Research, Military Institute of Aviation Medicine, Warsaw, Poland

2 Department of Simulator Studies and Aeromedical Training, Military Institute of Aviation Medicine, Warsaw, Poland

Source of support: Own sources

Author's address: M. Piotrowski, Department of Simulator Studies and Aeromedical Training, Military Institute of Aviation Medicine, Krasińskiego Street 54/56, 01-755 Warsaw, e-mail: mpiotrowski@wiml.waw.pl

Abstract: The Military Institute of Aviation Medicine possesses a specialized research device—a thermobaric chamber—designed to simulate environmental conditions occurring during flight inside an aircraft, as well as in emergency situations such as damage to the aircraft's external structure. The simulator enables the generation of conditions with reduced atmospheric pressure and a wide range of variable temperature and humidity levels. Research involving both humans and technical equipment is conducted in the thermobaric chamber. The chamber is equipped with a wired system for monitoring the environmental conditions inside it. However, due to the specific nature of some studies, it is necessary to use multiple additional temperature sensors placed in various locations within the tested object. Therefore, the use of sensors with wireless data transmission is the most advantageous. Simultaneous measurement using multiple such sensors requires the implementation of a measurement data acquisition system.

The aim of this work was to develop and construct a functioning prototype of a system for recording environmental conditions using a wireless sensor network. The proposed solution met the required functional specifications. The system enabled the recording of humidity and temperature, as well as the archiving of measurement data.

Keywords: Message Queueing Telemetry Transport (MQTT), temperature measurement, Wireless Sensor Network (WSN), Zigbee

Cite this article: Krej M, Piotrowski M, Dereń M: Recording of Environmental Parameters Using a Wireless Sensor Network. Pol J Aviat Med Bioeng Psychol 2022; 28(3): 32-40. DOI: 10.13174/pjambp.30.06.2025.04

INTRODUCTION

The specific nature of research conducted in low-pressure and low-temperature chambers, or other research devices, often requires the use of multiple sensors for measuring physical quantities, particularly temperature and humidity. Studies involving individuals in climate chambers are associated with their physical activity, which in many cases makes it impossible to use sensors connected to recorders via cables. Therefore, the use of wireless sensors is a favorable solution. Such an approach enables system-level management of measurement data acquisition.

Wireless Sensor Networks (WSNs) are groups of autonomous measurement devices that combine the functions of a sensor and a wireless communication unit. Typical WSN devices consist of:

– a measurement subsystem, responsible for measuring physical quantities (e.g., temperature, humidity, pressure, light intensity, or presence of chemical compounds) and providing them as electrical signals;
– a wireless communication subsystem, responsible for data transmission between network nodes and for sending information to a central data collection and processing system;
– a controller, which manages device operation, processes data from the measurement subsystem, oversees wireless communication, and controls power consumption;
– a power source, typically a battery or rechargeable cell, enabling long-term autonomous operation of the sensor node. Other power solutions also exist, such as energy harvesting [3], which involves obtaining power for the device from the surrounding environment;
– auxiliary interfaces, such as debugging, configuration, or device functionality extension interfaces.

The aim of this work was to develop and construct a functioning prototype of a system for recording environmental conditions using a wireless sensor network.

REVIEW OF SELECTED WIRELESS COMMUNICATION STANDARDS AND TEMPERATURE SENSORS

In this technical note, the authors focus exclusively on solutions and devices that, in their assessment, offer low energy consumption, compact size, and low unit purchase cost.

IEEE 802.11 Standard and Wi-Fi

Wi-Fi technology operates using radio waves that can transmit data between devices such as computers, smartphones, and routers. The transmission mechanism is based on modulation of the radio signal, which carries digital information. Wi-Fi uses advanced encoding and encryption techniques to ensure the security of transmitted data.

The architecture of a Wi-Fi network allows for the creation of small local networks as well as large-scale networks covering entire buildings. Devices can communicate directly or via access points, which extend the range of the network. The key features of the IEEE 802.11 standard and Wi-Fi include:

1. Network size: In practice, an unlimited number of nodes; transmission range up to several hundred meters.
2. Radio signal frequency range: 2.4 GHz and 5 GHz.
3. Security: Communication based on the WPA3 standard, use of 192-bit encryption keys, and the Simultaneous Authentication of Equals (SAE) protocol.
4. Advantages: High throughput, widespread adoption, ease of integration.
5. Disadvantages: Relatively high energy consumption compared to other solutions.

IEEE 802.15.1 Standard and Bluetooth LE

The Bluetooth Low Energy (Bluetooth LE) wireless communication protocol was designed to minimize power consumption through short and infrequent data transmissions between devices. It is particularly useful in applications such as medical

Tab. 1. Selected temperature sensors that can be used in the discussed technical solution of the system, using Wi-Fi communication.

Manufacture/ Model	Measured temperature range [°C]/ Temperature accuracy [°C]	Measured humidity range/ Humidity accuracy	Supply	References
Emylo/ AC859	-9.9°C to 60°C/ ±1°C	0%RH~99%RH/ ±5% RH	3 x AAA	[7]
Govee/ H5179	-20°C to 60°C/ ±0.3°C	0%RH to 99%RH/ ±3%RH	3 x AAA	[9]
Ubibot/ WS1	20°C to 60°C/ N/A	10%RH to 90%RH/ N/A	2 x AA	[30]

Tab. 2. Selected temperature sensors that can be used in the discussed technical solution of the system using Bluetooth LE communication.

Manufacture/ Model	Measured temperature range [°C]/ Temperature accuracy [°C]	Measured humidity range/ Humidity accuracy	Supply	References
Govee/ H5104	-20°C to 60°C/ ±0.3°C	0% to 99%RH/ ±3%RH	2 x AA	[10]
INKBIRD TECH/ IBS--TH2	-40°C to 60°C/ ±0.3°C	0%RH to 99%RH/ ±3%RH	2 x AAA	[12]
MOCERO/ ST6	-20°C to 60°C/ ± 0.3°C	0% RH to 100%RH/ ±3%RH	Rechargeable Battery	[16]
Setti/ SS302	-9.9°C to 60°C/ ± 1°C	0%RH to 99%RH/ ±5% RH	CR2032	[23]
Shelly/ H&T Gen3	0°C to 40°C/ N/A	30% RH to 70 % RH/ N/A	4 x AA	[24]

Tab. 3. Selected temperature sensors that can be used in the discussed technical solution of the system using Zigbee communication.

Manufacture/Model	Microcontroller/ Sensor	Measured temperature range [°C]/ Temperature accuracy [°C]/ Response time (temperature)	Measured humidity range/ Humidity accuracy/ Response time (humidity)	Supply	References
Lumi United Technology (Aqara)/ WSDCGQ11LM	NXP JN5169/ Sensirion AG SHT30	-40°C to 125°C/ ±0.2°C/2 s	0%RH to 100%RH/±2%RH/8 s	CR2032	[15] [17] [22]
Shenzhen Sonoff Technologies Co./ SNZB-02P	Silicon Labs EFR32MG22C22/ Sensirion AG SHT40	-40°C to 125°C/ ±0.2°C/2s	0%RH to 100%RH/±1.8%RH/4 s	CR2477	[25] [26]
ThirdReality/ 3RTHS24BZ	N/A	-10°C to 50°C/ ±1°C/N/A	0%RH to 95%RH/±2%RH/N/A	2 x AAA	[27]
Tuya Inc./ TS0201	Tuya Inc. TYZS5 / Sensirion AG SHT30	-40°C to 125°C/ ±0.2°C/2s	0%RH to 100%RH/±2%RH/8 s	2 x AAA	[28] [29] [22]
Heiman/HS1HT	N/A	-10°C to 50°C/ ±1°C/N/A	0%RH to 95%RH/N/A/N/A	CR2450	[11]

sensors, consumer electronic devices, and other compact equipment that require long-term operation using batteries as the power source. The key features of the IEEE 802.15.1 standard and Bluetooth LE include:

1. Network size: Up to 8 active nodes in a piconet (IEEE 802.15.1) or 1 central with multiple peripherals (Bluetooth LE); transmission range up to several dozen meters.
2. Radio frequency range: 2.4 GHz.
3. Security: Based on AES-128 encryption keys (BLE) or E0 algorithm (IEEE 802.15.1) and pairing mechanisms such as Just Works, Passkey Entry, and Numeric Comparison (primarily BLE).
4. Advantages: Energy efficiency (especially BLE), low implementation cost.
5. Disadvantages: Limited number of active nodes, relatively short range.

IEEE 802.15.4 Standard and Zigbee

A key feature of this standard is the ability to create large-scale sensor networks that can include dozens or even hundreds of devices. Protocols based on IEEE 802.15.4 are well-suited for use in industrial and home automation systems, as well as monitoring systems. The key features of the IEEE 802.15.4 standard and Zigbee and Z-Wave include:

1. Network size: In practice, an unlimited number of nodes; transmission range up to several dozen meters.
2. Power consumption: Very low.
3. Frequency range: 868 MHz (in Europe) and 2.4 GHz.
4. Advantages: Very low energy consumption, ability to create large-scale sensor networks, support for mesh network topology, communication protocols dedicated to Wireless Sensor Networks (WSNs) and the Internet of Things (IoT) [31].
5. Disadvantages: Limited transmission range, lower throughput compared to Wi-Fi.

ITU-T G.9959 Standard and Z-Wave

Similar to devices based on the IEEE 802.15.4 standard, a key feature of the ITU-T G.9959 standard is the ability to create large-scale sensor net-

Tab. 4. Selected temperature sensors that can be used in the discussed technical solution of the system using Z-Wave communication.

Manufacture/Model	Measured temperature range [°C]/ Temperature accuracy [°C]	Measured humidity range/ Humidity accuracy	Supply	References
Aeotec/ZWA039-C	-10°C to 65°C/±0.2°C	10%RH to 95%RH/N/A	CR2477	[1]
Aeotec/ZWA024	-10°C to 50°C/±0.2°C	20%RH to 80%RH/±5 %RH	2 x CR123A	[2]
Philio/PST02-B	-10°C to 40°C/N/A	0%RH to85%RH/N/A	CR123A	[18]
ZOOZ/ZSE44	-15°C to 40°C/±0.2°C	N/A	CR2450	[33]

works that can include dozens or even hundreds of devices. The key features of the ITU-T G.9959 Standard and Z-Wave include:

1. Network size: The maximum number of devices in a single Z-Wave network is 232 devices.
2. Frequency range: 868.42 MHz (in Europe) and 908.42 MHz (in North America).
3. Transmission speed: 9.6-100 kbps with a range of up to approximately 30 meters.
4. Power consumption: Very low.
5. Disadvantages: Vendor dependency - some features of devices may be limited to a specific manufacturer's ecosystem (vendor lock-in).

SENSOR SELECTION

The selection of the sensor to be used in the prototype of the measurement system for recording environmental conditions in the thermobaric chamber was based on the following requirements:

Overall dimensions and mounting method

Since the device is intended to be used, among other things, for measuring the environment surrounding the human body, its dimensions must allow placement on the outer side of a hand inserted into a glove or on the shin at the height of the boot upper, above the ankle joint. For studies involving technical objects, a desirable feature is the ease of attaching the sensor to components made of steel with magnetic properties. Some sensor housings are equipped with a magnet, which can facilitate quick attachment to the test object without the need for additional mounting elements.

Measurement range

The thermobaric chamber can generate thermal conditions in the range of –60°C to +70°C. For testing technical objects, it is recommended that the usable measurement range of the temperature sensor corresponds to the full range of temperatures that can be produced in the chamber. In tests involving human subjects, the required temperature measurement range of the sensor should cover the warm comfort zone, taking into account the threshold temperatures that may cause hypothermia or hyperthermia.

Power Supply

The sensor is required to have its own internal power source. It is undesirable for the sensor to rely on an external power source, despite being capable of wireless data transmission.

Cost

It is assumed that the unit purchase cost of a single sensor should not exceed 15 euros.

Technical Documentation

Access to technical information about the microcontroller and type of measurement sensor used is required. It is also mandatory to have access to technical documentation that enables the development of custom software for wireless communication, supporting multi-channel acquisition of measurement data.

As a result of the analysis of technical parameters and the available documentation, the sensor designated SNZB-02P was selected for work on the measurement system prototype. This sensor provides measurements within the required temperature range and allows for the use of custom software in a wireless measurement network.

PROTOTYPE OF THE ENVIRONMENTAL PARAMETERS RECORDING SYSTEM

The result of the work is a functioning prototype of an environmental parameters monitoring system, developed using devices communicating via Zigbee and Message Queuing Telemetry Transport (MQTT) protocols, as well as specialized scripts written in the Python programming language, which form the highest application layer of the solution. The prototype of the described system consists of:

– Temperature and humidity sensors, model SN-ZB-02P,
– USB-Zigbee adapter, model ZBDongle-P,
– PC-class computer running GNU/Linux operating system, Debian 12.0 distribution,

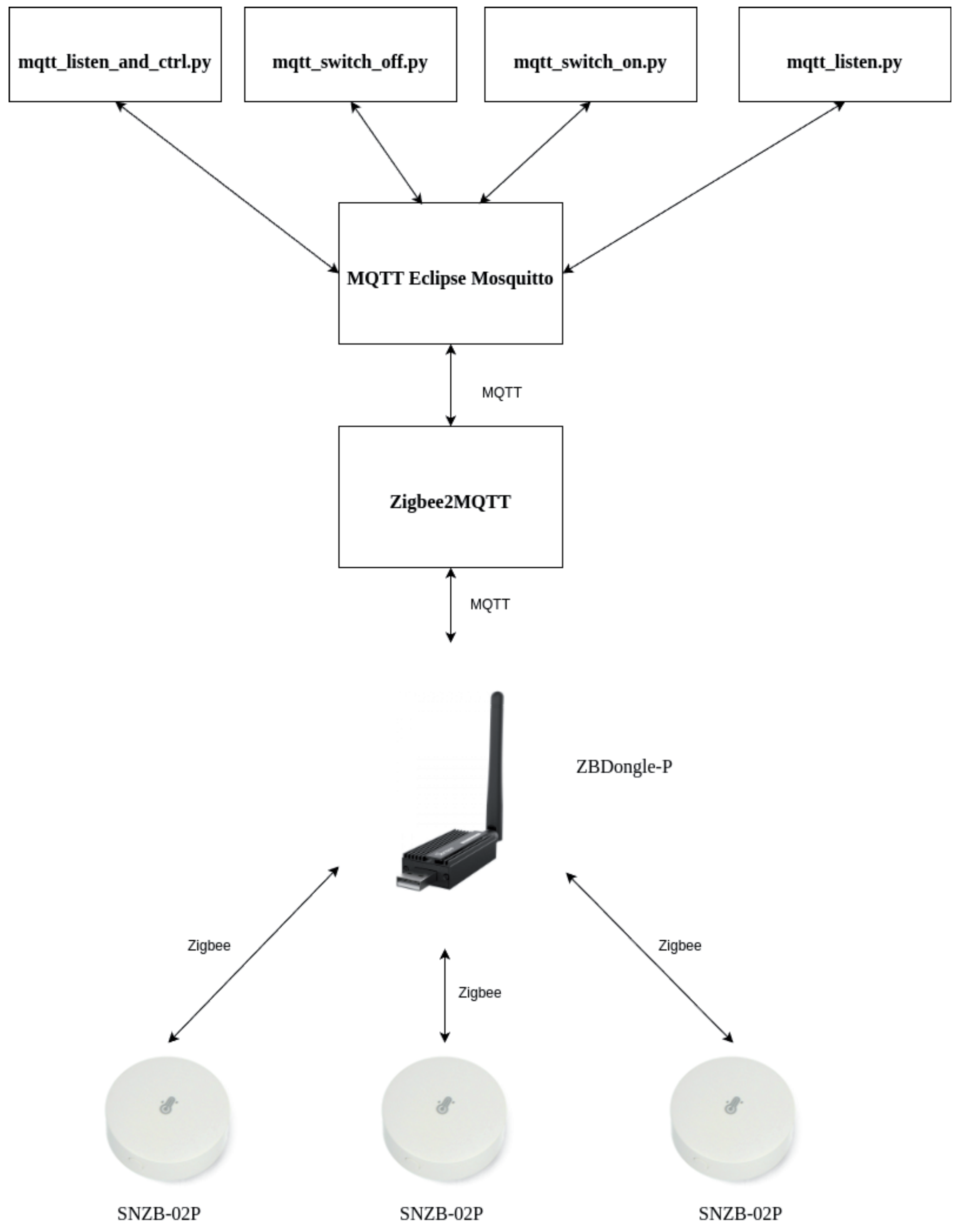

Fig. 1. Block diagram of the prototype environmental parameters recording system. Python scripts: mqtt_listen.py, mqtt_switch_off.py, mqtt_switch_on.py, mqtt_listen_and_ctrl.py. MQTT broker: MQTT Eclipse Mosquitto. Software: Zigbee2MQTT (custom development).

– Software, the source code of which (written in Python) is available at: https://github.com/mkmkmk/MqttZigbeePy [14].

The MQTT data transmission protocol is designed for devices with limited resources or operating in networks with limited bandwidth, for example, in IoT systems [32]. This protocol has been recognized as an OASIS (Organization for the Advancement of Structured Information Standards) standard and has received ISO recommendation status (ISO/IEC 20922). MQTT is intended for automation systems, enabling, among other things, efficient management of IoT device networks. In Fig. 1, a block diagram of the prototype environmental parameters recording system is shown.

The presented work is based on the official documentation of the Zigbee2MQTT project [12], but has been adapted to the GNU/Linux operating system environment, Debian distribution, offering an alternative to the original configuration for the Raspberry Pi Single Board Computer (SBC). Zigbee2MQTT is open-source software (OSS) that serves as a bridge between Zigbee devices and automation systems using the MQTT protocol.

To ensure isolation of the operating system's system libraries from the libraries used by the developed prototype software, a Python virtual environment (venv) [19] was installed and configured.

The development of the system prototype began with the configuration of the MQTT broker Eclipse Mosquitto [6], which is the central communication node between devices supporting the MQTT protocol. Eclipse Mosquitto is an open-source MQTT broker software that acts as an intermediary in communication between devices operating under the publish-subscribe architecture.

Installing the MQTT broker consists of executing the following commands:

– sudo apt install -y mosquitto mosquitto-clients
– sudo systemctl enable --now mosquitto

The above commands install the MQTT broker and configure it to start automatically with the operating system. Next, the Node.js runtime environment [4] version 18.x was installed. Node.js is a JavaScript runtime environment that allows JavaScript code to be executed outside a web browser, primarily on the server side. Node.js provides the necessary runtime environment for Zigbee2MQTT [13]. The following commands were used for its installation:

– curl -fsSL https://deb.nodesource.com/setup_18.x | sudo -E bash -
– sudo apt-get install -y nodejs

The next step in the system configuration was installing the Zigbee2MQTT software. This process began with creating and configuring the appropriate installation directory and downloading the software code from the repository hosted on GitHub:

– sudo mkdir -v /opt/ zigbee2mqtt
– sudo chown -Rv ${USER}: /opt/ zigbee2mqtt
– git clone --depth 1 https://github.com/Koenkk/zigbee2mqtt /opt/ zigbee2mqtt

Next, the Node.js code was compiled by executing the following commands:

– cd /opt/ zigbee2mqtt
– npm ci
– npm run build

Particular attention must be paid to configuring system permissions for the USB port, which is critical for proper communication with the USB-Zigbee adapter used. This required executing the following commands:

– sudo usermod -a -G tty $USER
– sudo chmod 666 /dev/ttyUSB0

The system also required appropriate configuration of the Zigbee2MQTT software by editing the configuration.yaml file located at: /opt/zigbee2mqtt/data/configuration.yaml

This file specifies parameters such as the address, unique Zigbee device identifier (Extended PAN ID / IEEE address), serial port number used by the USB-Zigbee, and other settings specific to the installation.

To run the software as a system service that starts automatically on system boot, it was necessary to create and configure the following file:

– /etc/systemd/system/zigbee2mqtt.service

After creating the service file, it was activated in the system using the following commands:

– sudo systemctl daemon-reload
– sudo systemctl enable --now zigbee2mqtt

The system operation can be monitored and diagnosed by checking the service status and logs:

– sudo systemctl status zigbee2mqtt
– sudo journalctl -xe -u zigbee2mqtt.service -f

As part of the software prototype development, a series of example Python scripts were prepared demonstrating the basic functionalities of the system.

The script mqtt_listen.py [14] serves as a basic tool for monitoring MQTT communication within the system. Its main components are:

– on_connect() function — responsible for establishing a connection with the MQTT broker and subscribing to all topics with the prefix "zigbee2mqtt/#",
– on_message() function — processes received messages by decoding JSON data and displaying temperature and humidity information,
– error handling for loading JSON files and other exceptions to ensure stable operation,
– MQTT client configuration using the paho-mqtt library [8].

The script uses callback mechanisms for asynchronous message processing, which ensures efficient operation even with a large number of devices.

The scripts mqtt_switch_on.py [14] and mqtt_switch_off.py [14] are used for direct device control. Its key elements are:

– the use of environment variables (.env) to store device identifiers,
– the switch_control() function implementing control logic by publishing appropriate MQTT messages,
– automatic management of the connection with the broker (connect/disconnect),
– serialization of commands in JSON format.

The most complex software component is the script mqtt_listen_and_ctrl.py [14], which combines monitoring and control functions. Its main features include:

- an extended on_message() function containing control logic based on sensor values,
- the ability to extend control based on humidity (sample code commented out),
- advanced error handling and filtering of unwanted messages.

This script demonstrates the capability to create complex automation scenarios where the state of devices controlled via the Zigbee and MQTT protocol stack is dynamically adjusted to environmental conditions. The code structure allows for easy addition of further rules and control conditions.

All scripts use the paho-mqtt library, which ensures reliable communication with the MQTT broker, and the python-dotenv library [21], which is used for secure management of application script configuration. The script code has been optimized for readability and ease of future modification, facilitating adaptation of the system to individual needs.

It is important to emphasize that the system requires particular attention during initial configuration. One must ensure that the USB-Zigbee adapter is properly connected before starting the service, and any issues with the prototype's operation can be diagnosed by analyzing system logs. In some cases, it may be necessary to reboot the system after changing USB port permissions.

DISCUSSION

The above prototype is a flexible solution allowing for further expansion. The proposed standard communication protocols and other widely used IT technologies enable easy integration of the prototype with other IoT automation solutions.

The presented solution can be successfully applied in various practical applications, such as monitoring temperature at multiple points on the surface of subjects' bodies inside a climate chamber, among others.

The prototype is characterized by the simplicity of its basic configuration as well as the capability to create complex operational algorithms.

A significant advantage of the described solution is its modular construction. All system components — from the MQTT broker, through the Zigbee2MQTT software, to the Python scripts — can be independently updated and adapted to specific needs. This flexibility allows gradual system expansion with new functionalities without the need to rebuild already functioning components.

The use of standard components and GNU/Linux operating system mechanisms, such as systemd for service management, simplifies diagnosing and resolving potential issues.

All system components can be updated using the operating system's standard mechanisms, enabling prompt responses to potential security threats. The ability to configure authentication and encryption in the MQTT protocol ensures secure communication among all system elements.

By employing the Python programming language for control scripts, the system offers opportunities for expansion and integration with other operating systems. The rich Python package library [20] allows easy addition of new function-

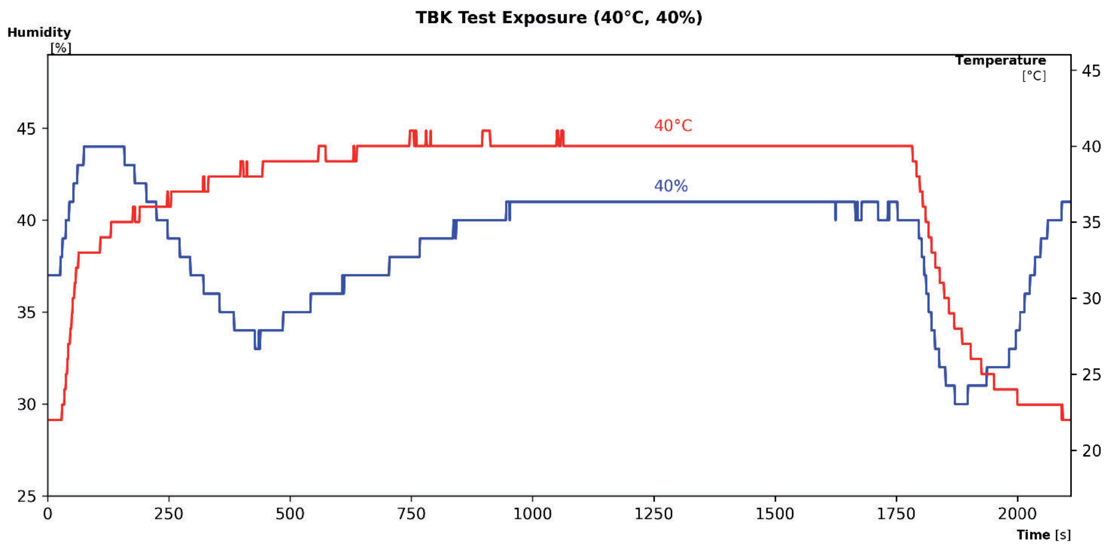

Fig. 2. Humidity and temperature measurement results recorded by the prototype measurement system during a test exposure in the thermobaric chamber (TBK) (own work).

alities, ranging from simple measurement data logging to databases, through graphical chart generation and measurement data statistics, up to advanced control algorithms and system automation. Additional materials and information describing the solution, together with example configuration files and the full source code of developed scripts, are available in a public repository on GitHub. They can be found at https://github.com/mkmkmk/MqttZigbeePy [14].

Depending on the requirements imposed on the measurement system, the frequency of recording measurement results may be specified. The frequency of recording measurement results may be influenced by the optimization of the measurement system aimed at minimizing energy consumption. In such cases, energy-efficient measurement sensors that communicate periodically with the recorder may be used. Example recording results using energy-efficient sensors are presented in the form of a graph (Fig. 2). The stepwise character of the graph results from periodic data transmission of the measurement results.

CONCLUSIONS

The objective of the work, consisting in the development and implementation of a functional prototype system for recording environmental conditions using a wireless sensor network, has been achieved. The proposed solution met the assumed functional requirements. The system enables the registration of humidity and temperature, as well as the archiving of measurement data. It seems justified to continue the work in order to develop and implement a graphical user interface that allows users to easily operate the system.

AUTHORS' DECLARATION

Study Design: Mariusz Krej, Marcin Piotrowski, Mirosław Dereń. **Manuscript preparation:** Mariusz Krej, Marcin Piotrowski, Mirosław Dereń. The Authors declare that there is no conflict of interest.

REFERENCES

1. Aeotec. Technical data: ZWA039-C [Internet]. Aeotec; [cited 2025 May 22]. Available from: https://aeotec.freshdesk.com/support/solutions/articles/6000227919-a%C3%ABrq-temperature-and-humidity-sensor-technical-specification
2. Aeotec. Technical data: ZWA024 [Internet]. Aeotec; [cited 2025 May 22]. Available from: https://aeotec.freshdesk.com/support/solutions/articles/6000246153-multisensor-7-technical-specifications
3. Ávila BYL, Vázquez CAG, Baluja OP, Cotfas DT, Cotfas PA. Energy harvesting techniques for wireless sensor networks: A systematic literature review. Energy Strategy Reviews. Elsevier; 2025 Jan; 57:101617.
4. Dahl R. Node.js [Internet]. [cited 2025 May 22]. Available from: https://nodejs.org/
5. Eclipse Foundation. Eclipse Mosquitto An open source MQTT broker [Internet]. Eclipse Foundation; [cited 2025 May 22]. Available from: https://mosquitto.org/
6. Emylo. Technical data: AC859 [Internet]. Emylo; [cited 2025 May 22]. Available from: https://www.emylo.com/product/emylo-mini-wifi-wireless-temperature-and-humidity-sensor-ac859
7. Fersing P, Light R. paho-mqtt [Internet]. [cited 2025 May 22]. Available from: https://pypi.org/project/paho-mqtt/
8. Govee. Technical data: H5179 [Internet]. Govee; [cited 2025 May 22]. Available from: https://eu.govee.com/products/wi-fi-thermo-hygrometer
9. Govee. Technical data: H5104 [Internet]. Govee; [cited 2025 May 22]. Available from: https://eu.govee.com/products/goveelife--bluetooth-hygrometer-thermometer-white
10. Heiman. Technical data: HS1HT [Internet]. Heiman; [cited 2025 May 22]. Available from: https://www.heimantech.com/product/?type=detail&id=12
11. INKBIRD TECH. Technical data: IBS-TH2 [Internet]. INKBIRD TECH; [cited 2025 May 22]. Available from: https://inkbird.com/products/hygrometer-ibs-th2
12. Kanters K. Zigbee2MQTT [Internet]. [cited 2025 May 22]. Available from: https://github.com/Koenkk/zigbee2mqtt

13. Kanters K. Zigbee2MQTT [Internet]. [cited 2025 May 22]. Available from: https://www.zigbee2mqtt.io/

14. Krej M. MqttZigbeePy [Internet]. [cited 2025 May 22]. Available from: https://github.com/mkmkmk/MqttZigbeePy

15. Lumi United Technology. Technical data: WSDCGQ11LM [Internet]. Lumi United Technology; [cited 2025 May 22]. Available from: https://www.aqara.com/en/product/temperature-humidity-sensor/specs/

16. MOCERO. Technical data: ST6 [Internet]. MOCERO; [cited 2025 May 22]. Available from: https://mocreo.com/st3-st6/

17. NXP. Technical data: JN5169 [Internet]. NXP; [cited 2025 May 22]. Available from: https://www.nxp.com/docs/en/data-sheet/JN5169.pdf

18. Philio. Technical data: PST02-B [Internet]. Philio; [cited 2025 May 22]. Available from: https://static1.squarespace.com/static/5c1afbd1266c07479bc41525/t/611b660f656eac5a62c3060c/1629185553631/PST02_Manual-A1-20210721.pdf

19. Python Software Foundation. Python virtual environmets (venv) [Internet]. [cited 2025 May 22]. Available from: https://docs.python.org/3/library/venv.html

20. Python Software Foundation. Python Package Index [Internet]. [cited 2025 May 22]. Available from: https://pypi.org/

21. Saurabh K. python-dotenv [Internet]. Available from: https://pypi.org/project/python-dotenv/

22. Sensirion AG. Technical data: SHT30 [Internet]. Sensirion AG; [cited 2025 May 22]. Available from: https://sensirion.com/media/documents/213E6A3B/63A5A569/Datasheet_SHT3x_DIS.pdf

23. Setti. Technical data: SS302 [Internet]. Setti; [cited 2025 May 22]. Available from: https://setti.pl/produkty/bezpieczenstwo/czujniki/czujnik-temperatury-i-wilgotnosci-ss302

24. Shelly. Technical data: H&T Gen3 [Internet]. Shelly; [cited 2025 May 22]. Available from: https://www.shelly.com/products/shelly-h-t-gen3-matte-white

25. Shenzhen Sonoff Technologies Co. Technical data: SNZB-02P [Internet]. Shenzhen Sonoff Technologies Co.; [cited 2025 May 22]. Available from: https://sonoff.tech/product/gateway-and-sensors/snzb-02p/

26. Silicon Labs. Technical data: EFR32MG22C22 [Internet]. Silicon Labs; [cited 2025 May 22]. Available from: https://www.silabs.com/documents/public/data-sheets/efr32mg22-datasheet.pdf

27. ThirdReality. Technical data: 3RTHS24BZ [Internet]. ThirdReality; [cited 2025 May 22]. Available from: https://3reality.com/product/temperature-and-humidity-sensor-lite/

28. Tuya Inc. Technical data: TS0201 [Internet]. Tuya Inc.; [cited 2025 May 22]. Available from: https://solution.tuya.com/projects/CMa4oc9d30jori

29. Tuya Inc. Technical data: TYZS5 [Internet]. Tuya Inc. [cited 2025 May 22]. Available from: https://developer.tuya.com/en/docs/iot/tyzs5-zigbee-module-datasheet?id=K97gu1imy1m3u

30. Ubibot. Technical data: WS1 [Internet]. Ubibot; [cited 2025 May 22]. Available from: https://ubibot.pl/en/temperature-and--humidity-logger-wifi-ws1

31. Wang Y, Zhang K, Liang M. Application case and circuit design optimization of temperature and humidity control system by using Internet of Things. Int J Syst Assur Eng Manag [Internet]. Springer Science and Business Media LLC; 2023 Jul 28 [cited 2025 May 22]; . Available from: https://link.springer.com/10.1007/s13198-023-02036-6

32. Waseem H, Montonen JK, Rahman AM, Salo TH, Halme AP, Vanhala JJ. Development of an IoT platform for Wearable Biosignal Monitoring Systems. IFAC-PapersOnLine. Elsevier; 2024 Jan; 58(9):31–6.

33. ZOOZ. Technical data: ZSE44 [Internet]. ZOOZ; [cited 2025 May 22]. Available from: https://www.getzooz.com/downloads/zooz-z-wave-plus-700-series-temperature-humidity-xs-sensor-zse44-manual.pdf