BLUETOOTH ENVIRONMENTAL MONITORING SYSTEM

Bianca-Laura HUMINIC

Faculty of Electrical Engineering and Computer Science, "Transylvania" University of Braşov, Romania
(bianca.huminic@student.unitbv.ro)

DOI: 10.19062/1842-9238.2023.21.2.8

Abstract: *The primary goal of this project is to construct an environmental monitoring system integrating a Bluetooth HC-06 module. Drawing upon acquired expertise, the project resulted in the development of a functional system capable of gathering sensor data, transmitting it wirelessly to a mobile device via the Bluetooth module, and displaying it in real-time. The utilization of the FreeRTOS operating system facilitated synchronous data collection, orchestrated through tasks synchronized by semaphores, thereby guaranteeing the integrity of real- time data acquisition and transmission.*

Keywords: Bluetooth, Arduino, Monitoring System, Display, Breadboard, FreeRTOS, Sensors

1. INTRODUCTION

Environmental quality monitoring is a fundamental service targeted at observing and predicting the dynamics of both qualitative and quantitative aspects of environmental components. It involves a systematic characterization and surveillance of environmental conditions, essential for understanding prevailing states and trends. Environmental monitoring systems play a crucial role in ensuring human health, safety, and well-being by continuously assessing various environmental parameters. In recent years, Bluetooth environmental monitoring systems have emerged as effective tools for real-time monitoring of environmental conditions in various settings. However, existing solutions often face challenges related to sensor integration, task management, and user interaction. In this paper, I present a novel Bluetooth environmental monitoring system that addresses these challenges through the integration of a flame sensor and a DHT11 sensor, the utilization of FreeRTOS for task management, and the development of a custom phone application for user interaction. By leveraging these innovations, this system offers enhanced functionality, reliability, and usability compared to existing solutions.

In this paper, a comprehensive examination of environmental monitoring utilizing Bluetooth technology is presented. The purpose is to provide both a state-of-the-art overview of current methodologies and a user-friendly tutorial suitable for novice readers. Moreover, seasoned researchers seeking to delve deeper into the subject matter will find valuable insights within this discourse. The paper is structured as follows to fulfill these objectives: commencing with a detailed exposition on the system overview (Section 2) and subsequent delineation of system implementation (Section 3), followed by an elucidation of the proposed system design (Section 4) and presentation of experimental findings (Section 5).

While the primary focus remains on environmental monitoring via Bluetooth technology, a brief discussion on the significance of employing FreeRTOS for task prioritization is also included. This aspect is particularly crucial for ensuring safety, as even a slight delay can have significant implications.

The conclusion emphasizes the successful integration of sensors, real-time task management, and user interaction capabilities, resulting in enhanced functionality, reliability, and usability of the system. Additionally, a roadmap for potential future endeavors in environmental monitoring utilizing Bluetooth technology is outlined, offering avenues for further exploration and advancement in the field.

2. SYSTEM OVERVIEW

Hardware components

A. Bluetooth module HC-06

The Bluetooth module employed in this system adheres to the Bluetooth 2.0 protocol and operates solely as a slave device. Utilizing the frequency hopping spread spectrum technique, denoted FHSS, the HC-06 module mitigates interference from other devices and facilitates full-duplex transmission.

Key advantages associated with the HC-06 module include its ability to facilitate wireless communication over short distances (typically less than 100 meters), as well as its affordability and user-friendly nature.

Communication via the HC-06 module is facilitated through the Universal Asynchronous Receiver Transmitter (UART) interface. This interface enables seamless connectivity of the module with any microcontroller or personal computer equipped with a RS232 (Recommended Standard 232) port.

B. DHT11 temperature and humidity sensor

The DHT11 sensor incorporates a negative temperature coefficient (NTC) thermostat for temperature measurement and a capacitive sensor for humidity measurement. Additionally, it integrates an 8-bit microcontroller responsible for transmitting temperature and humidity values as serial data. The sensor is capable of measuring temperatures ranging from -40°C to 80°C and humidity levels from 0 to 100 percent, with an accuracy of ±1°C for temperature and ±1 percent for humidity.

C. Flame sensor

The flame sensor serves the purpose of detecting flames or radiation sources within the wavelength range of 760 to 1100nm. It employs a YG1006 sensor, characterized by a high-speed NPN silicon photo resistor designed for infrared radiation detection.

Comprising an electronic circuit with a receiver, the sensor offers superior responsiveness and accuracy compared to conventional heat detectors, owing to its specialized mechanism.

Table 1. Measurement Unit

Sensors	Name	Symbol
Bluetooth module HC-06	Meters	[m]
Temperature	Celsius degrees	[°C]
Humidity	Percent	[%]
Flame sensor	Nanometers	[nm]

The values read by the sensors have a range of values for an optimal working environment. The HC-06 Bluetooth module has a range of approximately 8-10 meters, being perfect for this type of project. For the DHT11 temperature and humidity sensor, the range is between 20 and 60 degrees Celsius. The flame sensor can detect flames in the wavelength range of 760.

– 1100 nanometers. Small flames, like a lighter flame, can be detected at about 0.8m, and the detection angle is about 60 degrees, making the sensor particularly sensitive.

Table 2. Range of acceptable values

Parameter	Min range	Max range
Meters [m]	8	10
Temperature [C]	20	60
Humidity [%]	5	95
Flame [nm]	760	1100

Development environments

FreeRTOS is an open-source real-time operating system (RTOS) designed for embedded systems, specifically those with limited resources like microcontrollers and microprocessors. It offers a robust and flexible platform for developing embedded applications requiring deterministic real-time behavior, multitasking, and resource management.

FreeRTOS provides a preemptive, priority-based real-time kernel that ensures tasks are executed with deterministic timing. Tasks, representing individual application functions, can be dynamically managed, suspended, resumed, and deleted to optimize resource usage. The OS offers synchronization and communication mechanisms including for example semaphores, mutexes, queues, and event flags.

Communication protocols for wireless networks

Bluetooth protocol

Bluetooth operates within the 2.4 GHz frequency range, commonly utilized in wireless technologies, with a transmission power typically reaching up to 2.5 milliwatts. From a software standpoint, Bluetooth employs the Host Controller Interface (HCI) to establish communication between the Bluetooth host, including a laptop, and the core.

It also interfaces with protocols like Service Discovery Protocol (SDP), Radio Frequency Communication (RFCOMM), and Telephony Control Protocol (TCS) via the Logic Link Control and Adaptation Protocol (L2CAP). L2CAP is specifically designed to segment and reassemble large data packets for transmission to the baseband service via Bluetooth.

IEEE 802.15.1

IEEE functions as a networking standard that establishes the foundational framework for Bluetooth technology. The primary aim of this standard is to delineate specifications concerning the physical layer (PHY) and Media Access Control (MAC) for wireless connectivity among fixed or portable devices within a Personal Operating Space (POS).

3. SYSTEM IMPLEMENTATION

One of the initial setups adopted to facilitate Bluetooth transmission involved a rudimentary configuration utilizing a breadboard as the foundation. Integrated onto this breadboard were pivotal sensors including those dedicated to monitoring temperature, humidity, and flame detection. For the purpose of Bluetooth data transmission, the setup utilized an HC-06 module.

This module served as the intermediary conduit through which pertinent environmental data could be wirelessly transmitted to a compatible smartphone device. Subsequently, this transmitted data was made accessible for viewing and analysis via the interface of the phone’s serial monitor application.

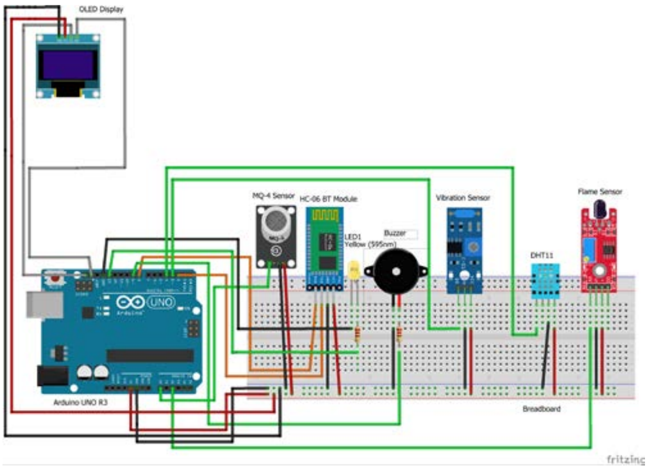

FIG. 1. Connection diagram

A. Sensor implementation

DHT11 temperature and humidity sensor

The sensor operates within a voltage range of 3.3V to 5.5V DC. It is imperative that the module adheres to voltage constraints, prohibiting the transmission of instructions beyondthe specified threshold.

In terms of communication and signal transmission, the DHT11 sensor utilizes a single bus communication protocol.

This bus architecture facilitates data exchange via a single dataline, enabling seamless communication between the microprocessor and the DHT11 sensor. Data transmission encompasses both integral and decimal components of relative humidity (RH) and temperature (T), each represented by 8 bits. The DATA pin of the sensor is connected to an I/O pin of the microcontroller unit (MCU), employing a 5K resistor for signal stability.

Flame sensor

The module offers dual voltage options of 3.3V or 5V, contingent upon the specifications of the development board. Communication with the development board is contingent upon the selected output mode, which can either be digital or analog. To fine-tune the sensitivity of the sensor, the LM393 comparator is employed.

In the case of digital output, connectivity is established by linking pin *D0* to the designated digital pin. Utilization of the *digitalRead(pin)* command enables users to ascertain thepresence of flame detection.

Alternatively, analog output provides valuable information regarding flame intensity, allowing for customized responses from the development board. Pin A0 is designated for connection to the analog pin.

For optimal performance and enhanced accuracy, preference is advised for analog output mode, complemented by the analog-to-digital conversion process.

B. Bluetooth module implementation

The Bluetooth module is supplied power from a +5V powersource. In order to enable its functionality, the Tx pin of the HC-06 module must be interconnected with the Rx pin ofthe Arduino microcontroller, while the Rx pin of the HC-06is coupled with the Tx pin of the Arduino board, employing a voltage divider. This voltage divider serves the purpose of converting the 5V signal emanating from the Arduino into a 3.3V signal, ensuring compatibility with the HC-06 module.

C. Real-time performance analysis

Using FreeRTOS, the following changes were made:

- The response time of both sensors exhibits enhancement, as evidenced by a reduction of 1 second in data acquisition when employing FreeRTOS, contrasting with lengthened data acquisition time in its absence.
- The adoption of FreeRTOS facilitates streamlined code organization through task management, fostering a structured and efficient execution framework.
- Task synchronization is achieved through the utilization of semaphores, ensuring harmonized operation among concurrent tasks. The incorporation of tasks introduces the concept of task prioritization, enabling the systematic arrangement of tasks. Priority is allocated to data transmission alongside the temperature monitoring task (Priority 1), followed bythe flame detection task (Priority 2).

4. THE PROPOSED SYSTEM DESIGN

System proposal

The project is committed to a comprehensive exploration and application of Bluetooth technology for transmitting gathered data. To accomplish this scholarly pursuit, the project intricately combines an Arduino development board with a Bluetooth HC-06 module.

This collaborative setup plays a crucial role in facilitating the smooth transmission of data collected by DHT11 temperature and humidity sensors, along with a flame sensor. The decision to utilize Arduino UNO modules is grounded in their inherent capability to connect with a wide array of peripheral devices via GPIO (General Purpose Input/Output) pins.

Additionally, the adoption of the Arduino IDE is essential for its scholarly accessibility and its ability to streamline programming processes, thereby enabling precise managementof the system's operational complexities.

Hardware design

This configuration was used in order to build a compact environmental monitoring system with uses like temperature and humidity measurement and flame detection [3].

In order to concentrate the ground and voltage pins, but alsoto keep the temperature as accurate as possible avoiding the Joule effect stimulated by the Arduino, we used the red and blue strips on the breadboard.

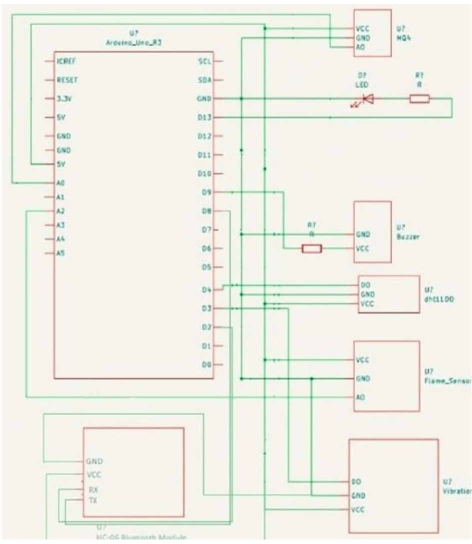

FIG. 2. Electric diagram

Software design

The firmware running on the microcontroller assumes responsibility for acquiring, processing, and transmitting sensor data. To manage tasks and ensure deterministic behavior, FreeRTOS is employed as the real-time operating system.

Additionally, a smartphone application has been developed for both Android and iOS platforms, allowing users to interactwith the system and visualize sensor data.

Incorporating appropriate libraries for individual sensors, the Arduino microcontroller, FreeRTOS, SPI communication, and semaphores has been essential.

Notably, special consideration was given to defining the pin and type of the temperature and humidity sensor separately. This was necessary due to the composition of the DHT11 mod- ule, which includes a thermistor for temperature measurement and a capacitive sensor for humidity measurement, requiringa specific library.

The DHT11 module is connected to pin 8 of the Arduino board. Furthermore, to enable communication with the Blue- tooth module, a SoftwareSerial object was initialized with Tx and Rx parameters, synchronized with the Arduino at a baud rate of 9600.

Monitoring commences once data transfer becomes avail- able via the Bluetooth module. Task creation utilized the xTaskCreate() function, with parameters specifying task name,FIFO size, and task priority.

```
xTaskCreate(
  TaskTrimiteDate
  ,  "task1"
  ,  128
  ,  NULL
  ,  1
  ,  NULL );
xTaskCreate(
  TaskTemperatura
  ,  "task2"
  ,  128
  ,  NULL
  ,  1
  ,  NULL );

   xTaskCreate(
  TaskFlacara
  ,  "task3"
  ,  128
  ,  NULL
  ,  2
  ,  NULL );

vTaskStartScheduler();
```

FIG. 3. Tasks creation

The xSemaphoreCreateBinary function is employed to in- stantiate a semaphore, a synchronization mechanism utilized for inter-task communication. Upon semaphore creation, memory allocation takes place within the FreeRTOS heap, enabling its subsequent utilization.

Following semaphore creation, the xSemaphoreGive (xBinarySemaphore) function is invoked to release the semaphore.Upon successful release, this function returns pdTRUE, indi- cating successful execution, whereas any encountered errors prompt a return of pdFALSE.

Subsequently, the xSemaphoreTake (xBinarySemaphore, 5) function is utilized to acquire the semaphore.

The parameter '5' signifies the duration, expressed in ticks, within which the semaphore is anticipated to become available [4]. Upon suc- cessful acquisition, this function returns pdTRUE; otherwise,it returns pdFALSE.

The interchange between states of the semaphore, from a state of 0 to being acquired by the xSemaphoreTake() function, is facilitated by the xSemaphoreGive() and xSemaphoreTake() functions. Should xSemaphoreTake() fail to return pdTRUE, the subsequent semaphore operations are not executed.

```
if (xSemaphoreTake(xSerialSemaphore, (TickType_t) 5) == pdTRUE)
  {
    sprintf(transmiteT, "Temperature:%d", temperatureData);// buffer

    xSemaphoreGive(xSerialSemaphore);
  }
```

FIG. 4. Semaphore

5. EXPERIMENTAL RESULTS

The experiments demonstrate the effectiveness and performance of the Bluetooth environmental monitoring system. The integration of multiple sensors allows for comprehensive environmental monitoring, with real-time data transmission via Bluetooth ensuring timely updates.

The utilization of FreeRTOS ensures efficient task management, enabling simultaneous execution of sensor reading tasks without interference. Furthermore, the custom phone application provides an intuitive interface for users to access and visualize sensor data, enhancing the system's usability and accessibility.

Table 3. Comparison of execution times

Simulation	*Temp/Flame without FreeRTOS*	*Temp/Flame FreeRTOS*
1	5.09/5.11sec	179.10/179.07ms
2	5.15/5.12sec	179.09/179.10ms
3	5.01/4.98sec	179.07/179.06ms
4	5.30/5.26sec	179.07/179.14ms
5	5.50/5.55sec	178.99/179.13ms
6	4.89/4.94sec	179.12/179.16ms
Average	5.15/5.16sec	179.07/179.11ms

As evident in the system output, the serial monitor displays the time allocated for sensor measurements, facilitated by strategic task prioritization embedded within the system architecture. This intentional prioritization has led to a noticeable improvement in sensor responsiveness, significantly reducing the time required for data acquisition.

Specifically, the sensor response time has been shortened from an initial 5-second interval to approximately 200 milliseconds, highlighting a considerable enhancement in the system's operational efficiency and real-time data processing capabilities.

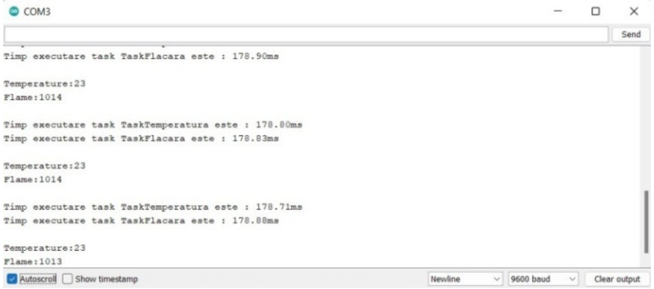

FIG. 5. Snapshot of Serial monitor

Table 4. Bill of materials for the project components

Nr.	Component	Cantity	Price
1	Arduino UNO R3	1	40 RON
2	HC-06 Module	1	30 RON
3	DHT11 Sensor	1	15 RON
4	Flame sensor	1	10 RON
5	Breadboard	1	20 RON
6	Jumper wires	10	15 RON
Total cost			130 RON

The phone in this case acts as a receiver, taking data from the Arduino board and displaying it on the display [6].

When the user presses the ”1” key on the phone, the mobile app displays the current temperature readings from the sensors integrated into the monitoring system.

Similarly, when the user presses the ”2” key on the phone, the app retrieves and displays the real-time flame value detected by the flame sensor, as shown in figure [6]

This functionality offers users immediate access to critical environmental data with just a few taps on their mobile de- vice. It enhances user convenience and accessibility, allowing individuals to monitor temperature and flame levels in their surroundings in real-time, without the need for additional hardware or complex interfaces.

Experimental studies

Study 1 - Normal Indoor Conditions (20°C, 50 % RH) This condition represents typical indoor environmental parameters, characterized by a temperature of 20°C (68°F) and a relative humidity (RH) of 50 %. It serves as a baseline for comparison against other conditions.

Location: Office Building

The office building serves as a representative indoor environment typical of many commercial establishments. It consists of office spaces, meeting rooms, and common areas where individuals work and interact daily.

FIG. 6. Phone app

Experimental Setup

- The environmental monitoring system is installed within a designated office space on one of the building's floors.
- Sensors are strategically placed to capture temperature and humidity levels representative of standard indoor conditions.
- Data collection occurs during regular office hours to capture fluctuations in environmental parameters throughout the day.

Study 2 - Elevated Temperature (30°C) and Humidity (70 % RH)

In this condition, the temperature is increased to 30°C (86°F), simulating warmer indoor environments, while the relative humidity is elevated to 70 %. This condition assesses the system's response to heightened thermal and moisture levels.

Location: Greenhouse

The greenhouse provides a controlled environment for cultivating plants, characterized by elevated temperature and humidity levels conducive to plant growth.

Experimental Setup

- The environmental monitoring system is placed within the greenhouse, positioned to capture temperature and humidity readings across different areas.
- Temperature is artificially increased to 30°C (86°F) using heating systems installed within the greenhouse.
- Humidity levels are raised to 70 % RH through misting systems or by controlling irrigation frequency.
- Data collection occurs over several hours to observe variations in environmental parameters under greenhouse conditions.

Study 3 - Low Temperature (10°C) and Humidity (30% RH)

This condition involves lowering the temperature to 10°C (50°F) and reducing the relative humidity to 30 %, mimicking cooler and drier indoor settings. It examines the system's performance under conditions of decreased thermal and moisture levels.

Location: Cold Storage Facility

The cold storage facility is designed to maintain low temperatures for preserving perishable goods like food and pharmaceuticals.

Experimental Setup

- The environmental monitoring system is deployed within the cold storage facility, situated among stored goods to monitor temperature and humidity levels.
- The facility's cooling systems are adjusted to achieve a temperature of 10°C (50°F) throughout the storage area.
- Humidity levels are reduced to 30 % RH using dehumidification systems installed within the facility.
- Data collection spans over an extended period to assess the stability of environmental conditions within the cold storage facility.

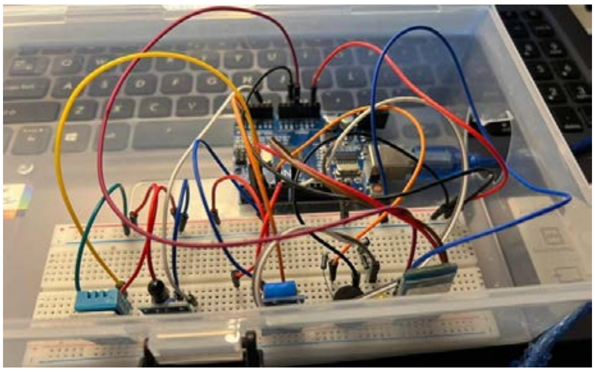

FIG. 7. Assembly

6. CONCLUSIONS

The presented Bluetooth environmental monitoring system represents a significant advancement in environmental monitoring technology. By addressing the limitations of existing solutions and introducing innovative features including sensor integration, real time task management, and user interaction capabilities, the system offers enhanced functionality, reliability, and usability.

Moving forward, continued research and development efforts can further optimize the system's performance, expand its capabilities, and explore new avenues for environmental monitoring and management.

In future iterations, the purpose is to downsize the device forseamless integration into smaller spaces. Additionally, plans involve introducing an environmental control mechanism, for example an automated sprinkler system activated in response to elevated temperatures. Moreover, there is a proposal to develop a mobile application enabling remote control of these regulatory devices.

This application is anticipated to incorporate features for sending email alerts in cases where users do not engage with the application or fail to receive notifications on their mobile devices.

Analyses and issues encountered

One of the challenges encountered was ensuring the accuracy and reliability of sensor readings, particularly during the calibration process. Variations in sensor performance and environmental conditions posed challenges in calibrating the sensors to provide accurate measurements consistently. Addressing these issues required careful calibration procedures and ongoing monitoring to identify and rectify any discrepancies in sensor readings.

There were also problems related to connecting the Bluetooth module to the existing application in the case of the iOS operating system.

In terms of compatibility, the iOS operating system supports a narrower range of Bluetooth protocols than an Android device.

Maintaining stable Bluetooth connectivity and ensuring reliable data transmission between the monitoring system and the mobile application presented challenges, especially in environments with high levels of interference or obstacles.

Issues like signal interference, range limitations, and packetloss affected the reliability of data transmission, requiring optimization of Bluetooth communication protocols and strategiesfor error detection and correction.

REFERENCES

[1] J. Lygeros, S. Member, S. M. T. Arampatzis, *A Survey of Applications of Wireless Sensors and Wireless Sensor*, IEEE International Symposium on Midterrean Conference (2005);

[2] Environmental Monitoring System with Arduino, California, 2012;

[3] V. Manimohzi, D. Nivetha, P. Nivethitha, V. Anupriya, *SmartEnvironmental Monitoring System Using Labview*, 2017;

[4] K. Noordin, C. Chow, M. Ismail, *Microcontroller-based Weather Monitoring System*, January 2015, 2006;

[5] O. Chinenye, O. Kingsley, *Open Access Design of a Supervisory Control and Data Acquisition System for Monitoring Temperature and Light Intensity of a Workstation*, American Journal of Engineering Research, 2017;

[6] H. Halvorsen, *Introduction to Arduino An open-Source Prototyping Platform*, Oslo, 2011;

[7] Monitoring System Wireless Sensor-based Agricultural System, 2014;

[8] T. Murphy, *Introduction to the Arduino hands-on Research in Complex Systems what is a Microcontroller*, 2014.