Greenhouse System Standardization Using the Matter Protocol

Elek-Ákos Kovács*, Levente Szász*, Csaba Gálfi†, Zsolt Szabó†, Áron-Tamás Sipos†, Csaba Sulyok*
*Faculty of Mathematics and Computer Science, Babeș-Bolyai University
RO-400084 Cluj-Napoca, Romania
E-mail: {elek.kovacs, levente.szasz}@stud.ubbcluj.ro, csaba.sulyok@ubbcluj.ro
†Codespring
RO-400458 Cluj-Napoca, Romania
E-mail: {galfi.csaba, szabo.zsolt, sipos.aron}@codespring.ro

***Abstract*—The Sixth Sense Sprinkler project aims to create a simple and smart greenhouse system that can transmit real-time data on the status of the greenhouse using different sensors. It also provides remote control of irrigation and ventilation systems through a mobile application. All components are based on the Matter standard, which gives the system its modular character; any third-party sensors, lights, and locks also following the Matter standard are compatible out of the box. The core of the network is a server called the Sprinkler Hub, through which the microcontrollers in the foil housing can be controlled. These controllers are equipped with various sensors such as temperature, soil moisture and air humidity sensors, which monitor the state of the environment. Access to the Sprinkler Hub is provided through an Android mobile application, which allows users to commission the devices, remotely control them and view historical measurements via charts. The paper describes how the system works, provides insights into the technologies used, and gives guidance on how to use the functionalities.**

***Index Terms*—Matter protocol, Thread network, sensor data, smart greenhouse**

I. Introduction

Operators of greenhouses and greenhouse systems face many challenges [1] to protect their crops. Most problems are caused by an inadequately designed environment. A critical point in crop production is monitoring and maintaining adequate temperatures, but also the lack of accurate irrigation and ventilation, which can lead to negative consequences such as dehydration, the risk of fungal growth or the attraction of harmful insects to the greenhouse. Designing a system that takes into account all the relevant aspects of this sector of crop production [2] is therefore no easy task.

Many Internet of Things (IoT) technologies have already been integrated into agriculture [3], [4] and there are many solutions available to facilitate greenhouse operations with the tools of digitalization, such as *Semios*[1] insect control systems and *Sensaphone Sentinel*[2] various sensing and monitoring devices. However, the appropriate selection of these can present additional challenges for novice and experienced farmers alike. Consider the wide variety of manufacturers and products to choose from to best meet the needs of your current crop, not to mention the compatibility issues [5], [6] that arise when consolidating several different systems. These systems can also increase the complexity of workflows.

The Sixth Sense Sprinkler is designed for small-scale farming, particularly family businesses, as a key tool in smart precision agriculture. Tailored to the needs of smaller operations, it optimizes resource use, reduces manual labor, and boosts crop yields. By focusing on accessibility and practicality, the project ensures that even small-scale farmers can benefit from the latest IoT technologies.

The Sixth Sense Sprinkler project aims precisely to create a smart system that makes greenhouse management more efficient and convenient. This system aims to change the way agriculture is digitised by using the novel Matter [7] protocol primarily known for its use in smart homes but also applicable to other IoT fields, as demonstrated in this project. The system will allow users to monitor the current status of their greenhouses through a mobile application. To do this, it uses soil moisture sensors and a complex device that measures air humidity and internal temperature. The data is collected via a local network, but by connecting the Sprinkler Hub, this information can be accessed remotely via a chart showing historical data.

Similar solutions already exist in the global market, mainly as mobile apps like *Google Home*[3] or *Apple Home*[4], which are the default smart home apps in Android and iOS, respectively. The main issues with these systems include the complexity of the pairing process, the need to use third-party apps or accounts, and the need to link with Google Home. In addition, neither system provides historical measurements.

The *Home Assistant*[5] as a software system could also be a popular choice for smart home automation, including greenhouses, mainly due to the configurability and flexibility of the system. However, it has certain limitations and difficulties that may make it not an option for everyone. Home Assistant, although beneficial in terms of customizability, also carries significant complexity, which can be challenging for

[1] https://semios.com/solutions/insect-pest-management
[2] https://www.sensaphone.com/sentinels
[3] https://home.google.com/get-app/
[4] https://www.apple.com/home-app/
[5] https://www.home-assistant.io/

those who lack technical knowledge. As it is an operating system or platform to be run locally, the user must have some technical background knowledge.

The remainder of the article is structured as follows: Section II details the building blocks of the greenhouse system. Section III covers the mobile application, and Section IV discusses the Sprinkler Hub. Finally, Section V describes the functions and applications of the system.

II. Architecture

The architecture of the project (see Fig. 1) consists of electronic components, smart plugs, microcontrollers, a server called Sprinkler Hub, the Sixth Sense Sprinkler Android application and the Google Nest Hub 2nd Gen[6]. The current section describes the role of these components and the communication between them in detail.

The system provides two networks, called Matter fabrics, for Matter devices. Users can deploy them using the mobile application to create a local Android fabric. The users can share their devices with the server by connecting them to the Sprinkler Hub's fabric if they want their devices to save measurements or participate in automation[7]. The mobile application and the server communicate via a REST (Representational State Transfer) API (Application Programming Interface).

The microcontrollers together with the Google Nest Hub form a Thread [8] network. The established network can transmit sensor readings to the Nest Hub without internet access. As a Thread Border Router, the Nest Hub transmits the received data via WiFi to the mobile application as well as to the Sprinkler Hub.

A. The Model of the Greenhouse

In order to integrate the greenhouse system developed in the framework of the project into a realistic environment, the concept of a miniaturized greenhouse model (see Fig. 2) is proposed already at the design stage. The inspiration is a SENAPSKÅL decorative model reminiscent of a small greenhouse, chosen for its price, practical size, and ease of transport.

The frame has a plastic pot as its central feature, housing not only the plant but also the soil moisture sensors. A 0.5-litre water tank at the bottom of the frame supplies water to the irrigation system. A DC 5-volt motor-driven pump at the bottom of the water tank transfers water from the tank to the plants through silicone tubes. In addition to the irrigation functions, special attention is paid to simulating the ventilation system. A DHT22 [9] type sensor is used to monitor temperature and humidity, while ventilation is simulated by a DC 5-volt fan, also located close to the sensor, whose rapid airflow helps to reduce temperature and humidity.

The fan and pump require a 5-volt DC supply and cannot be connected directly to the mains. To overcome this problem, we use AC-DC step-down power adapters to convert the 230 VAC voltage to 5 VDC. These adapters are each connected to a smart plug so that the fan and pump can be easily controlled by switching them on/off.

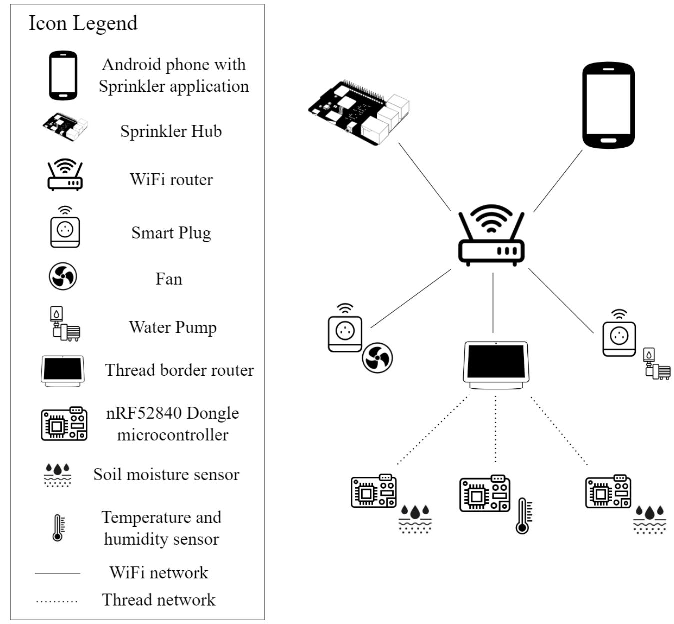

Fig. 1. Complete architecture diagram.

B. Measurement and Sensing

The microcontrollers are responsible for reading and transmitting the soil moisture, temperature, and air humidity measurements via the Thread protocol.

Many microcontrollers are on the market, but the project requires an *SoC* (System on a chip) that supports Matter over Thread communication [10]. The most popular choices include Silicon Labs' EFR32 series, Espressif's ESP32-C3 series and Nordic Semiconductor's nRF52 series. These brands are part of the Connectivity Standard Alliance[8], which builds and supports Matter.

Several criteria were considered when selecting the microcontrollers, including features, price, and size (see Table I).

The EFR32xG24 Explorer Kit[9] looks promising, but is expensive, especially since more microcontrollers are needed. The ESP32-C6-DevKitM-1[10] would be a good choice, but the documentation for the nRF52840 Dongle[11] proved to be more detailed and elaborate. In addition, Nordic Semiconductor provides many useful webinars and user-friendly tools to help with system development. Furthermore, the dongle is small, and the USB connector on the board means there is no need for a micro-USB cable to connect to a PC. In conclusion, it

[6] https://store.google.com/us/product/nest_hub_2nd_gen

[7] This functionality is not available at the moment

[8] https://csa-iot.org

[9] https://www.silabs.com/wireless/thread

[10] https://docs.espressif.com/projects/espressif-esp-dev-kits/en/latest/esp32c6/esp32-c6-devkitm-1/user_guide.html

[11] https://www.nordicsemi.com/Products/Development-hardware/nRF52840-Dongle

TABLE I
COMPARISON OF CONSIDERED MICROCONTROLLERS

	EFR32xG24 EExplorer Kit	ESP32-C6-DevKitM-1	nRF52840 Dongle
Clock Frequency	78 MHz	160 MHz	64 MHz
Flash Memory	1.5 MB	4 MB	1 MB
WiFi	Supported	Supported	Supported
Bluetooth	Supported	Supported	Supported
Thread	Supported	Supported	Supported
Power Consumption	Low	Medium	Very Low
Community Support	Good	Good	Excellent
Price	36$	8$	14$

was decided that the nRF52840 Dongle would fill the role of the microcontroller used.

Two types of measuring devices have been developed: Weather Station and Moisture Sensor, one is equipped with a DHT22 type sensor and the other with a Capacitive Soil Moisture Sensor V2.0 [11]. The microcontrollers are also equipped with a factory reset button, a battery holder, and a Matter QR code which is required for commissioning. Different values are obtained depending on the environment as the microcontroller with the soil moisture sensor reads an analog signal. To make the measurements as accurate as possible, these measuring devices are also equipped with a red button to help users calibrate the sensor by reading the analog signal in air and clean water.

The firmware for the microcontrollers is written in C++ using the nRF Connect SDK[12].

C. Irrigation and Ventilation

It is important to emphasize that, in line with the greenhouse model (see Fig. 2), only low-power actuators are used, which simulate the operation of an environment rather than providing the functionality of a full-scale smart greenhouse.

To simulate the ventilation, a 5-volt fan is used, which is placed directly next to the DHT22 sensor mentioned earlier. This reduces the measured heat by approximately 2-3 degrees Celsius when the fan is activated.

A sprinkler system is an essential part of the greenhouse. In this case, irrigation is achieved through a silicone tube with a diameter of 5 mm, which transports the water from the water source to the plant. The system is driven by an electric 5-volt submersible pump.

These two actuators are powered by Meross Smart Plug Mini[13] smart sockets, which is of particular importance as these devices already support the Matter protocol over WiFi. In addition, the use of a smart plug is a more cost-effective solution than purchasing pre-built smart appliances.

III. SPRINKLER MOBILE APPLICATION

The Sprinkler mobile application is a Matter Controller written in Kotlin for Android. It allows the end user to easily interact with its devices through a user-friendly interface. The application is currently available in English only, and its usage is discussed in Section V.

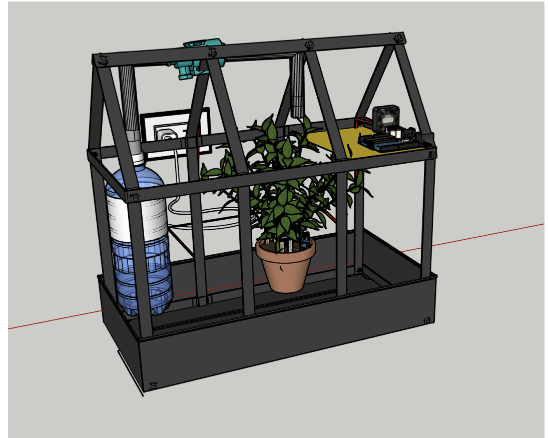

Fig. 2. Greenhouse model illustrating the placement of components

The application creates a local Fabric to which users can connect and control their devices. It provides two environments for this, called *Greenhouse* and *Garden*. These are, of course, only symbolic environments to help people organise their various devices. The application also provides the ability to turn devices on and off, share them with similar services, and retrieve statistics via Sprinkler Hub.

Before using the application, users must pass an *Auth0* authentication. To do this, they must log in with their Google account or with an email address and password registered in the system.

The application communicates with the Sprinkler Hub through a REST API, detailed in Section IV-D. It uses a Ktor client[14] to send HTTP requests, which simplifies request formulation and provides proper error handling.

IV. SPRINKLER HUB

Outlined below are problems that arose during development, that neither the Matter protocol nor the developed Android application can solve:

- The measurements made by the sensors are only momentary values that are not stored. As a result, the application is not able to provide statistical data on the state of the greenhouse to the user.
- The Matter standard creates a local area network, which requires the presence of a central unit, the Matter Controller, usually located in the greenhouse and close to its Matter devices. If this role is filled by the user's mobile phone, for example via the Sixth Sense Sprinkler Application, the connection between the devices is lost when the phone leaves the greenhouse area, creating a potential *single point of failure*
- During development, the possibility of implementing automation routines that would operate 24 hours a day,

[12]https://www.nordicsemi.com/Products/Development-software/nRF-Connect-SDK
[13]https://www.meross.com/en-gc/smart-plug/matter-smart-plug/159
[14]https://ktor.io/

regardless of the mobile application or the physical presence of the user, was raised.

Based on the summary, it became necessary to create a tool that could connect to the Matter network, monitor the status changes of the devices, and store the information in a database. This allows users to access this data via their mobile devices.

A. Structure

The Sprinkler Hub is a program written in Node.js in Typescript and deployed on a Raspberry Pi 4 [12] hardware and placed near the greenhouse.

The hub uses the Matter technology through an official TypeScript implementation named *matter.js*, taking on a role in the network as a Commissioner and Controller, managing devices. For data storage, it uses SQLite database, to which the TypeORM framework provides a connection. The data is made available to the mobile application by a web server based on Express.js [13].

B. Matter Network Control

A Matter fabric can only be created by a Matter Controller, and to connect to additional devices, a Commissioner role must be implemented. On restart, the hub will attempt to connect to the previously paired devices, if the connection is not successful (for example, because the device is no longer part of the network), the hub will stop attempting and instead move on to the next device in the queue.

As the Raspberry Pi does not have a built-in camera, it is not possible to use QR codes when pairing devices, instead, a 13-digit pairing code is used. This allows devices to be identified on the network using a *discriminator*. This process does not require another Bluetooth Low Energy (BLE) connection between the hub and the device, as the mobile application has already connected the device to the local network, so it can be accessed via the multicast DNS protocol.

Once a device is successfully connected, the hub subscribes to all attributes in the device's Cluster and monitors their changes.

Each Matter device has a unique NodeID, both in the Android fabric and in the hub fabric. Since the mobile application does not know the NodeIDs assigned by the hub, these identifiers must be communicated to the hub during requests. The Sprinkler Hub keeps a database of device network ID mappings to ensure device identification and management.

C. Data Storage

Sprinkler Hub stores additional data in its database beyond the basic Matter data model.

- The *shared_node* table stores the NodeID mappings of devices between different fabrics, i.e. the unique identifiers received from the mobile application and the hub.
- The *device_attribute* table stores the data measured during device state changes, including NodeID, EndpointId, clusterId (which identifies the attribute), attribute name, measured value, and time of measurement.

The connection between the SQLite database and the program is provided by TypeORM, a TypeScript-based Object Relational Mapping (ORM) solution. ORM allows the conversion of classes into database tables and vice versa. It provides interfaces for performing CRUD (create, read, update, delete) operations, which simplifies the basic operations on data. In addition, TypeORM's *QueryBuilder* function allows you to build elegant queries using TypeScript syntax that are converted into SQL commands.

D. API

Application Programming Interface is a set of rules that allows communication and data exchange between software components. The web server running on the hub uses a REST-based API that accesses resources using HTTP requests.

The Sprinkler Hub is able to accept HTTP GET and POST requests. These requests allow new devices to be integrated into the network and provide the ability to serve historical data to clients. These resources can be accessed by specifying the appropriate path, or by specifying query parameters to filter or group resources.

The main paths include *POST /commission*, which introduces an unknown Matter device into the system. The request is required to specify the NodeId from the Android fabric, and the pairing code in the form of a *JSON* object. The *GET /values* path can be used to retrieve data measured by devices already present in the system, which can optionally be filtered using the *from* and *to* query parameters. However, to query for a specific attribute of a device, the Endpoint and Cluster id in the Matter specification must be added to the path: */values/{nodeId}/endpoint/{endpointId}/cluster/{clusterId}*.

Since the frequency of measurements involves a lot of data transfer, it is also possible to retrieve monthly, daily and hourly averages of the values.

To control the devices remotely, an HTTP request must be sent to */toggle*, */turnOn*, or */turnOff*.

V. Usage

In this section, it is demonstrated how users can build their smart greenhouse system using Sixth Sense Sprinkler tools. The use of the buttons on the microcontrollers will be described, and special emphasis will be placed on the management of the Android application, showcasing the practical usage of its available functionalities.

A. Setting Up Devices

1) Setting up the Thread Border Router: To use the sensors, a Thread Border Router is required. The Nest Hub 2nd Gen from Google, which has been frequently mentioned, is utilized for this purpose and must be set up before use. This process involves downloading the Google Home application to a phone, scanning the QR code on the Nest Hub using the Add → Google Nest or partner device button, and following the instructions in the application. Within a few minutes, the Nest

Hub will start up, creating the Thread network. Alternatively, another Border Router can be used.

2) Setting up the microcontrollers and sensors: After the battery has been correctly inserted into its holder for the first time, the LED on the microcontroller starts flashing yellow. This means that it is advertising itself via the BLE protocol. At this point, the user can commission it using the Sprinkler application as described in Section V-D. If the commissioning fails, the white button on the microcontroller can be pressed and held for 6 seconds to factory reset the device. The yellow LED will then flash, indicating that it is ready to be commissioned.

Before usage, the soil moisture sensor requires calibration. The user should press the large red button on the microcontroller, causing an LED to flash red. At this point, the sensor is waiting to be held in the air. The user should hold the sensor by its bottom so that it senses only the air and not the hand. Once this is done, pressing the button again confirms the action. The LED will light up red for 3 seconds, indicating that it is measuring the moisture in the air, which should be low; this value will correspond to 0%. The LED will then start flashing blue. At this stage, the user should place the sensor in a glass of water, taking care not to get any water beyond the white line, as this could damage the sensor. Pressing the button one last time will confirm this action. The LED will glow blue for three seconds, indicating that it is measuring the moisture in the water, which will correspond to 100% moisture. Finally, the LED will turn green for one second, indicating successful calibration and that the sensor is ready for use.

B. Log In

To use the application, the user must first log in. On the opening screen, a button labeled *"Sign in with Google"* will appear, which will redirect the user to a web page where they can sign in with a Google account or register with an email address and password. After a successful login, the application will redirect the user to the main page.

C. Main Page

At the top of the main page, on the right, a text displaying the name of the current environment greets the user. Next to it is an icon that, when clicked, brings up a drop-down menu for changing the environment. On the left, the profile picture of the logged-in account is visible. At the bottom of the screen is the navigation bar, which has three options: *Devices*, which takes the user to the main page; *History*, which directs the user to the screen that generates the charts; and *Settings*, which takes the user to the settings. This navigation bar is also present on all other screens. Above the navigation bar, there is an *Add Device* button for adding devices.

In the center of the screen, the devices already in use are displayed as cards. Each card shows the name of the device, an icon indicating the type of device (e.g. soil moisture or temperature and humidity sensor), and a label indicating whether the device is ON or OFF. If no devices are installed, the center of the screen remains blank. Fig. 3 shows the main page with commissioned devices.

D. Commissioning Devices

To commission devices, the user should press the *Add device* button. The application will then attempt to add the device to the actively selected environment. After pressing the button and enabling WiFi, Bluetooth, and the camera on the application, the Matter QR code on the device must be scanned. If the environment is too dark, the flashlight icon on the screen can be pressed to activate the phone's flash. Alternatively, the user can click on the *Set up without QR code* button to utilize the device pairing code. The next step involves waiting 1-2 minutes for the device to be added to the Android fabric. Once this process is complete, a pop-up window will appear, allowing the user to give the device a custom name. It is important to ensure that the device is placed as close as possible to the Nest Hub (preferably right next to it) during the commissioning process and that it is set up correctly.

E. Managing Devices

With the devices that have been commissioned, users can perform several actions: they can turn them off/on, share them with other services such as Google Home or Sprinkler Hub, and remove them.

To turn devices on and off, the user should tap the card of the desired device from the main page. When the user long-presses on the card, they will be directed to the device's screen, where they can also toggle it. On this page, various other operations for the device can also be performed.

To share the device, the user should press the *Share Device* button. In the window that appears, they should select the service they want to share it with. Notably, Sprinkler Hub will not be displayed in this window. If the user wants to share with Sprinkler Hub, they will need to select *Use pairing code*. After copying the pairing code that appears, the user should navigate back to the device's screen. Here, they should press the *Add to Sprinkler Hub* button. In the text box that appears, the code should be pasted, followed by pressing the *Confirm* button. After a short time, the operation will be completed.

To remove the device, the user should press the *Remove Device* button. The application will then prompt the user to confirm their intention to remove the device from the local fabric. After pressing the *Confirm* button, the device will be deleted from the system.

F. Retrieval of Charts

To retrieve data from past measurements and generate charts, the user should navigate to *History* option. Here, three drop-down menus allow the user to select the desired sensor, measurement, and resolution (per minute, hourly, daily, or monthly) for viewing the data. The *Open Date Picker* button enables the user to display a calendar, where they can enter a start and end date. Alternatively, the user can manually enter the two dates by pressing the pencil icon. After completing the selection, clicking on the tick will display the chart.

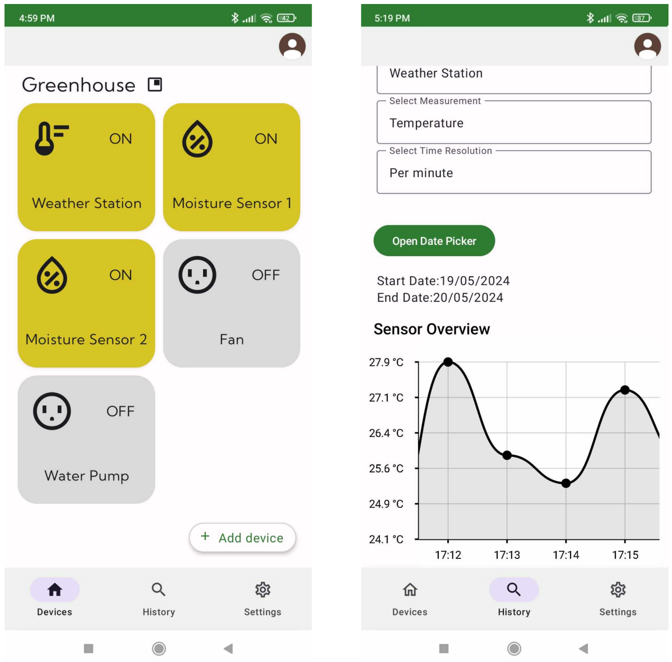

Fig. 3. The main page with commissioned devices (left) and the chart generated from Weather Station's temperature (C) measurements (right)

Fig. 3 shows a chart generated from the Weather Station's temperature measurements. It's important to note that the selected device must be shared with Sprinkler Hub, as that's where the data will come from. Otherwise, no chart will be displayed.

VI. Conclusion and Future Development

As part of the Sixth Sense Sprinkler project, Matter-compatible devices have been developed to measure various environmental factors and transmit data to users. An Android application has been created to allow users to manage these, and third-party devices, thereby making greenhouse maintenance more efficient.

The Android application successfully integrates both third-party and custom Matter devices, offering a diverse range of options for agricultural modernization. The measuring devices have an estimated battery life of 21 days and can be positioned 7 to 15 meters apart to maintain a stable connection.

During testing, five Matter devices efficiently managed a hobby-level garden. However, as the number of devices increases, it is recommended to employ an external database or expand the Raspberry Pi's storage. This enhancement would enable the Sprinkler Hub to provide users with comprehensive access to a range of historical measurement statistics.

Future efforts will focus on enhancing the project with additional features. Further development of the application will include the creation of a user interface that enables the customization of automation rules. This feature is critical, as each crop requires different environmental conditions, and farmers need to be able to tailor the system to the specific needs of their crops.

Another key goal is to design the application to send status notifications, such as when high temperatures activate the ventilation system or if a device disconnects. Additionally, the internationalization of the application will be addressed, as it is currently available only in English. Future updates will include support for multiple languages to cater to a broader user base.

Currently, devices are grouped into 'Greenhouse' and 'Garden' rooms. Plans include allowing users to create and name their environments to better reflect their individual greenhouses.

Lastly, an Over The Air update feature for devices will be introduced. This will enable future updates to be installed and maintained consistently and remotely, ensuring continuous improvement and maintenance of the system.

References

[1] I. Yilmaz, C. Sayin, and B. Özkan, "Turkish greenhouse industry: past, present, and future," *New Zealand Journal of Crop and Horticultural Science*, vol. 33, pp. 233–240, 2005.

[2] S. Ghani, F. Bakochristou, E. M. A. A. ElBialy, S. M. A. Gamaledin, M. M. Rashwan, A. M. Abdelhalim, and S. M. Ismail, "Design challenges of agricultural greenhouses in hot and arid environments," *Engineering in Agriculture, Environment and Food*, vol. 12, no. 1, pp. 48–70, 2019.

[3] S. Jaiganesh, K. Gunaseelan, and V. Ellappan, "IOT agriculture to improve food and farming technology," in *2017 Conference on Emerging Devices and Smart Systems (ICEDSS)*, 2017, pp. 260–266.

[4] K. Verma, N. Chandnani, G. Bhatt, and A. Sinha, *Internet of Things and Smart Farming*. Cham: Springer International Publishing, 2022, pp. 283–303. [Online]. Available: https://doi.org/10.1007/978-3-030-77528-5_15

[5] V. R. Konduru and M. R. Bharamagoudra, "Challenges and solutions of interoperability on IoT: How far have we come in resolving the IoT interoperability issues," in *2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon)*, 2017, pp. 572–576.

[6] S. S. Albouq, A. A. A. Sen, N. Almashf, M. Yamin, A. Alshanqiti, and N. M. Bahbouh, "A Survey of Interoperability Challenges and Solutions for Dealing With Them in IoT Environment," *IEEE Access*, vol. 10, pp. 36 416–36 428, 2022.

[7] W. Zegeye, A. Jemal, and K. Kornegay, "Connected Smart Home over Matter Protocol," in *2023 IEEE International Conference on Consumer Electronics (ICCE)*, 2023, pp. 1–7.

[8] S. B. A. Khattak, M. M. Nasralla, H. Farman, and N. Choudhury, "Performance Evaluation of an IEEE 802.15.4-Based Thread Network for Efficient Internet of Things Communications in Smart Cities," *Applied Sciences*, vol. 13, no. 13, p. 7745, 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/13/7745

[9] Y. A. Ahmad, T. Surya Gunawan, H. Mansor, B. A. Hamida, A. Fikri Hishamudin, and F. Arifin, "On the Evaluation of DHT22 Temperature Sensor for IoT Application," in *2021 8th International Conference on Computer and Communication Engineering (ICCCE)*, 2021, pp. 131–134.

[10] A. Nafkha, "Implementation and comparison of low power wireless protocols in a mesh topology," Ph.D. dissertation, Instituto Politecnico de Braganca (Portugal), 2021.

[11] P. Placidi, L. Gasperini, A. Grassi, M. Cecconi, and A. Scorzoni, "Characterization of Low-Cost Capacitive Soil Moisture Sensors for IoT Networks," *Sensors*, vol. 20, no. 12, p. 3585, 2020. [Online]. Available: https://www.mdpi.com/1424-8220/20/12/3585

[12] T. Maragatham, P. Balasubramanie, and M. Vivekanandhan, "IoT Based Home Automation System using Raspberry Pi 4," *IOP Conference Series: Materials Science and Engineering*, vol. 1055, no. 1, p. 012081, 2 2021. [Online]. Available: https://dx.doi.org/10.1088/1757-899X/1055/1/012081

[13] D. Herron, *Node.js Web Development*, 5th ed. Packt Publishing, 2020. [Online]. Available: https://www.packtpub.com/product/node-js-web-development-fifth-edition/9781838987572