

IoT based Horticulture Monitoring System

Monika Rabka¹⁽¹⁾, Dion Mariyanayagam¹⁽²⁾, Dr Pancham Shukla¹⁽³⁾

¹Department of Communications Technology and Mathematics
London Metropolitan University
London, United Kingdom

⁽¹⁾rabka.monika@gmail.com, ⁽²⁾dionysusmiroy@gmail.com,
⁽³⁾p.shukla@londonmet.ac.uk; spancham@yahoo.com

Abstract. With climate change and global warming in mind, vertical farms, hydroponics and urban greenhouses can now be found in many cities worldwide as we transform the ways we produce food. Additionally, recent implications of the COVID-19 pandemic prove that as a society we can harness the benefit of remote monitoring and automation for controlled-environment agriculture and horticulture. The subject matter of this paper is implementation of a solar-powered, Internet of Things (IoT) based Real-time Autonomous Horticulture Monitoring System (RAHMS). The RAHMS integrates a mobile application for viewing the greenhouse crop data and camera feed of plants, and interacts with cloud databases such as Firebase and MATLAB ThingSpeak for the scalability. In particular, a simple and distinctive design of a solar-powered, low energy consuming, and inexpensive greenhouse monitoring system is presented. The paper outlines RAHMS design methodology and showcases a proof-of-concept prototype with its core hardware and software components. The proposed system has a potential to further advance the practical aspects of the remote solutions for the cultivation and monitoring of horticulture and controlled-environment agriculture.

Keywords: Internet of Things (IoT), Cloud, Smart system, Horticulture, Greenhouse monitoring, Image processing

1 Introduction

Increasing food security in the times of climate change and global warming alongside the global population growth - expected to rise by 2.4 billion by 2050[1] has become an important area of research for scientists worldwide. With each degree rise in temperature, it's estimated that we will lose 10% of existing agricultural land. Controlled-environment agriculture (CEA) with vertical farms and urban greenhouses are being encouraged as a sustainable response to the increased demand of food. CEA also reduces the food wastage and production losses as it eliminates the pests and pathogens found in outdoor farms. Additionally, this form of food production reduces

the carbon footprint caused by transportation as CEA farms are mainly placed in and around cities [2][3]. Increase in implementation of indoor farms could lead to freeing up agricultural lands that in turn could be reverted back into forests - resulting in reduction of the greenhouse emissions in the atmosphere [4].

Remote monitoring and control of these farms is the obvious next step in trying to tackle the issues at hand. This is where the Internet of Things (IoT) comes into the fold. IoT is a network of Internet-connected devices that collect user-accessible data using sensors and processors and transfer it over wired or wireless networks. As of 2019, Cisco estimated that approximately 31 billion of IoT devices were connected to the Internet, and that number will rise exponentially over the next decade [5]. Integrating these low-cost and low-powered devices into CEA will help make farmers' daily work more manageable by reducing the amount of time needed for physical monitoring of the crops.

As part of this research, several similar systems in the area of agricultural monitoring and control have been evaluated. The key elements in these systems are sensors, microprocessor and actuators. In the literature review, we carefully looked at the features, strengths, and weaknesses of such systems in comparison to the proposed Real-time Autonomous Horticulture Monitoring System (RAHMS). Our evaluation of the relevant recent work in this area is summarised in Table I.

Table 1. Evaluated studies.

Paper	Features
"IOT Based Environment change Monitoring & Controlling in Greenhouse using WSN" [8] (2018)	<ul style="list-style-type: none"> • Usage of Raspberry Pi 3 as a gateway with Arduino Uno nodes for sensor data collection. • Theoretical control of the watering system, fans, light sliders and heater. • Usage of basic sensors such as the LM35 Temperature sensor, humidity, soil moisture and Light Dependent Resistor (LDR)
"IoT Based Automated Greenhouse Monitoring System" [9] (2018)	<ul style="list-style-type: none"> • Full utilisation of the fairly expensive Raspberry Pi for gathering sensor data, control of the actuators and connection to the Internet. • Usage of basic sensors such as YL69 moisture sensor and DHT11 temperature and humidity sensor. • Integration of cloud computing in the form of MATLAB's ThingSpeak open-source API. • The authors propose the creation of a custom website/application for better visualisation of data. The RAHMS project improves on this study by creation of an asynchronous website along with a bespoke mobile application.
"Design of intelligent greenhouse environment monitoring system based on ZigBee and embedded technology" [10] (2014)	<ul style="list-style-type: none"> • Focus on ZigBee implementation rather than the whole system's functionality. • Usage of local storage (SQLite3 database) along with an LCD screen for local display of gathered data. • Usage of ARM Micr02440 core board with Samsung S3C2440 microcontroller unit.
"IOT based Greenhouse Environment Monitoring and Controlling System using Arduino Platform" [11] (2017)	<ul style="list-style-type: none"> • Stands out with the use of GSM to control the greenhouse with SMS. • Usage of basic sensors such as the DHT11, Soil Moisture and an LDR. • RAHMS project improves on this study with the use of more advanced technologies such as Wi-Fi and 4G/LTE.

<p>“Secured IoT Based Smart Greenhouse System with Image Inspection” [12] (2020)</p>	<ul style="list-style-type: none"> • The strongest of all the above-researched studies. • Usage of the 32-bit MSP432 MCU for sensor data collection and Raspberry Pi for the camera functionality and encryption/decryption of the data. • Implementation of image inspection of the plants using image segmentation and classification using Open-Source Computer Vision (OpenCV) library. • Addition of a CO₂ sensor is an advantage as CO₂ is an integral part of CEA.
--------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

The works presented in [6] and [7] mainly focus on the Image Processing for greenhouse crops. Our work under RAHMS focuses on adapting and enhancing some of the features from all of these studies for a low-cost, solar powered and cloud enabled prototype.

2 Research method

2.1 Design of the system

The top-down design of the Real-time Autonomous Horticulture Monitoring System (RAHMS) is based on a client-server networking architecture model as shown in the block diagram of Fig 1. The microcontrollers (ESP32 MCU) used in this project are inexpensive units commonly used in the industry. In the context of the system, the RAHMS client consists of a standalone camera server (ESP32-CAM) and a sensor board which collects and sends the sensor data to the server. The server (Server ESP32 MCU) transfers the received data over the Internet to cloud APIs from where this data is then processed.

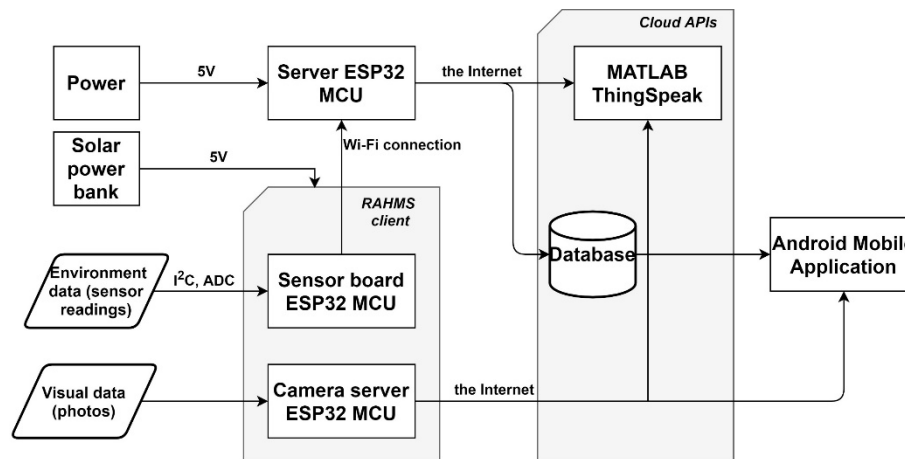


Fig. 1. Block diagram of the RAHMS, where the MCU and servers are ESP32 microcontrollers and database is the Firebase real-time database.

To achieve an autonomous system, RAHMS client is powered with a solar power bank, and the server receiving the sensor data is powered separately with 5V. In this proof-of-concept implementation, in order to be able to access the photos from the camera server MCU in ThingSpeak and Android application, a tunnelling service NGROK was used to expose the local server to the public network.

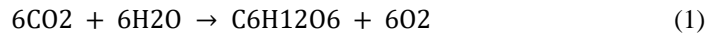
2.2 Electronic components

The sensors used in the RAHMS implementation are shown in Table II.

Table 2. Sensors used in the RAHMS.

Sensor	Description
Dallas Semiconductors DS18B20	Contact digital waterproof thermometer
Bosh Sensortec BME280	Combined air temperature, humidity and pressure digital sensor
AMS CCS811	CO ₂ and total volatile compound levels digital metal-oxide sensor
FC28	Resistive soil moisture sensor
GL55 Light Dependent Resistor	Lux sensor

These sensors were chosen based on the requirement to monitor essential parameters of the controlled-environment agriculture. The photosynthesis process, which produces oxygen and makes the green plants grow, is commonly defined with the chemical formula (1).



Hence, it was important to include soil moisture (water), CO₂, and light levels sensors. Additionally, soil temperature and air humidity, pressure and temperature sensors were added to monitor the overall conditions in the greenhouse. Too high or too low temperature can damage the crop, whereas high air humidity can protect crops from heat damage. Fig. 2 shows the 3D printed casing with the custom made PCB and RAHMS client components.

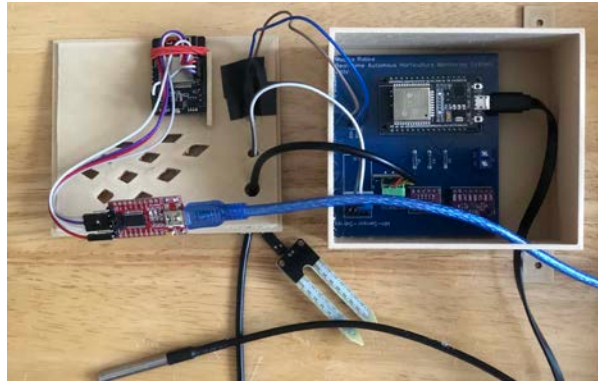


Fig. 2. Proof-of-concept implementation of the RAHMS client.

2.3 Networking

The RAHMS uses the client-server model for networking purposes. Fig. 3 shows a top-down network diagram of the RAHMS. The camera server connects directly to the Internet while the sensor board client (or clients) connects to the Server ESP32 MCU to transfer the sensor values to the cloud APIs.

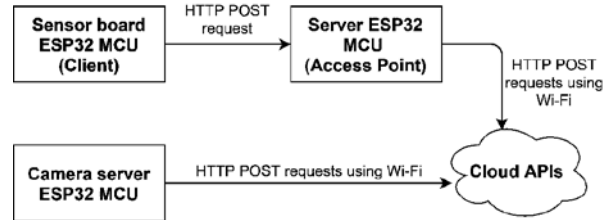


Fig. 3. The RAHMS network diagram.

3 Testing, results and discussion

3.1 Testing

The solar power bank used for system testing contains a 26800mAh battery. By calculating the current consumption as shown in Table III, we estimate that the power bank can support the client for up to 4 days continuously. To lower the power consumption, sleep mode on ESP32 MCUs could be implemented in the firmware to turn off the MCUs for set periods and only turn them on when taking a sensor reading or taking a photo. Sleep mode on ESP32 consumes only 5 μ A. This allows extended power bank battery life in darker months of the year (if used in urban greenhouses). Table III shows the tested and calculated current consumption of the RAHMS client. Fig. 4 shows the deployment of a 3D printed prototype of the RAHMS client in an outdoor (back garden) greenhouse setup.

Table 3. Measured and calculated power consumption

Power Consumption	
Initial Spike	290mA
Peak	260mA
Average	230mA
Wattage per hour (ideal/calculated)	0.0196708 kWh
Wattage per hour (measured)	0.0182835 kWh



Fig. 4. The RAHMS client tested in a greenhouse.

3.2 Firebase and Android Application

Fig. 5 (on the left) shows the Firebase JSON nodes as recorded during the greenhouse client test shown in Fig. 4. The current date and time are accessed using the Network Time Protocol (NTP) and stored in variables that are then used as part of the data path for the JSON object. This allows creation of separate nodes for each day and better organisation of stored data. On the right side of Fig 5, the second screen of the Android app after a successful login is presented. This activity contains a scrollable view of the last five recorded sensor values taken from the Firebase real-time database and the latest photo taken by the camera MCU.

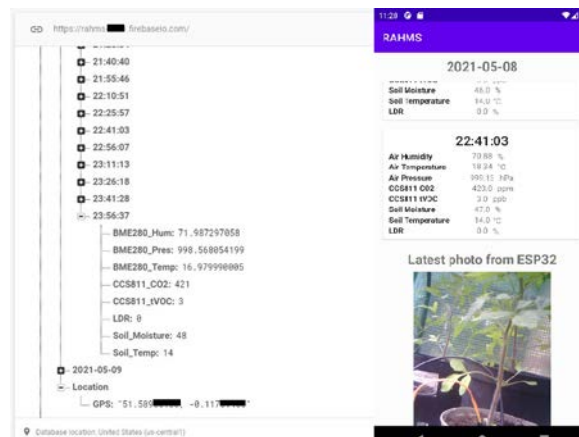


Fig. 5. Firebase real-time database nodes populated with the sensor values from the RAHMS client (left); Android app screen displaying Firebase data and the camera server photo (right)

3.3 MATLAB ThingSpeak

A simple MATLAB code (as shown in Fig. 6) was used for RGB to gray conversion and thresholding of the photo. Fig. 7 shows the ThingSpeak channel with all the sensor data visualised on graphs along with the unprocessed and processed photos of the plant being monitored in the greenhouse.

```
url = 'http://2dc03f624b79.ngrok.io/saved-photo/photo.jpg';
pic = webread(url);
data = imrotate(pic,90);
imphoto = rgb2gray(data);
imA = imphoto>150;
imA = double(imA);
for(i=1:3)
    data(:,:,i)=uint8(double(data(:,:,i)).*imA);
end
figure;
imshow(imA);
%imA=uint8(double(imA)*255);
title('Image from ESP32');
```

Fig. 6. Image processing MATLAB script.



Fig. 7. ThingSpeak channel graphs/visualizations populated with the sensor values from the RAHMS client, and processed and unprocessed images from the camera MCU.

4 Conclusions and future work

To conclude, this paper presented a proof-of-concept, inexpensive solution for remote monitoring of small to medium sized, remote greenhouses that can access the Internet, either in the form of broadband, satellite, or 4G/LTE access points. Some of the obstacles faced during the design process were regarding the cellular communication and weatherproofing the RAHMS client. This has been overcome by using a tunneling service NGROK to expose the camera server to the public network and using an outdoor standard 3D printing filament. In the era of climate change and global warming where controlled-environment agriculture and horticulture are rapidly becoming essential parts of the food-producing industry, we anticipate that these design efforts will stimulate the appetite for more advanced contributions to this topic.

For future work, the networking model could be improved by scrapping the server-side and introducing a 4G/LTE module so the sensor data could be sent directly to the cloud without the need for a Wi-Fi access point. Over-the-air implementation along with UI and image processing improvements in the mobile application would be advantageous. Additionally, a Machine Learning based notification system for diseased plants, an autonomous watering system and a larger range of sensors to increase the data set for yield prediction is under consideration.

As for security, the RAHMS could also implement end-to-end encryption of the data between the communication streams, with unique session IDs per client to prevent spoofing attacks or use unique credentials for each client using a RADIUS server.

5 Acknowledgment

The author, Monika Rabka, would like to thank the co-authors of this paper, Dion Mariyanayagam and her supervisor Dr Shukla for all the generous support and guidance given throughout this project. This work is supported in part by the University's Rescaling Funds awarded to Dr Pancham Shukla.

References

1. United Nations Department of Economic and Social Affairs, "World population projected to reach 9.7 billion by 2050," 2015. <https://www.un.org/en/development/desa/news/population/2015-report.html> (accessed Jun. 27, 2021).
2. D. Despommier, "The vertical farm: controlled environment agriculture carried out in tall buildings would create greater food safety and security for large urban populations," *J. für Verbraucherschutz und Leb.*, vol. 6, no. 2, pp. 233–236, 2011, doi: 10.1007/s00003-010-0654-3.
3. E. W. Stein, "The Transformative Environmental Effects Large-Scale Indoor Farming May Have On Air, Water, and Soil," *Air, Soil and Water Research*, vol. 14. SAGE Publications Ltd, Mar. 12, 2021, doi: 10.1177/1178622121995819.
4. H. Goldstein, "The Green Promise of Vertical Farms," *IEEE Spectrum*, 2018. <https://spectrum.ieee.org/energy/environment/the-green-promise-of-vertical-farms> (accessed Jun. 27, 2021).
5. L. Horwitz, "Internet of Things (IoT) - The future of IoT miniguide: The burgeoning IoT market continues - Cisco," Jul. 19, 2019. <https://www.cisco.com/c/en/us/solutions/internet-of-things/future-of-iot.html> (accessed Jan. 14, 2021).
6. D. Baquero, J. Molina, R. Gil, C. Bojacá, H. Franco, and F. Gómez, "An image retrieval system for tomato disease assessment," *2014 19th Symp. Image, Signal Process. Artif. Vision, STSIVA 2014*, pp. 1–5, 2015, doi: 10.1109/STSIVA.2014.7010156.
7. R. S. Sarkate, N. V. Kalyankar, and P. B. Khanale, "Application of computer vision and color image segmentation for yield prediction precision," *Proc. 2013 Int. Conf. Inf. Syst. Comput. Networks, ISCON 2013*, pp. 9–13, 2013, doi: 10.1109/ISCON.2013.6524164.
8. D. Shinde and N. Siddiqui, "IOT Based Environment change Monitoring Controlling in Greenhouse using WSN," *2018 Int. Conf. Information, Commun. Eng. Technol. ICICET 2018*, pp. 1–5, 2018, doi: 10.1109/ICICET.2018.8533808.
9. M. Danita, B. Mathew, N. Shereen, N. Sharon, and J. J. Paul, "IoT Based Automated Greenhouse Monitoring System," *Proc. 2nd Int. Conf. Intell. Comput. Control Syst. ICICCS 2018*, no. Icccs, pp. 1933–1937, 2019, doi: 10.1109/ICCONS.2018.8662911.
10. W. Qiu, L. Dong, F. Wang, and H. Yan, "Design of intelligent greenhouse environment monitoring system based on ZigBee and embedded technology," *Proc. 2014 IEEE Int. Conf. Consum. Electron. - China, ICCE-C 2014*, pp. 33–35, 2015, doi: 10.1109/ICCE-China.2014.7029857.
11. P. V. Vimal and K. S. Shivaprakasha, "IOT based greenhouse environment monitoring and controlling system using Arduino platform," *2017 Int. Conf. Intell. Comput. Instrum. Control Technol. ICICICT 2017*, vol. 2018-Janua, pp. 1514–1519, 2018, doi: 10.1109/ICICICT1.2017.8342795.
12. S. M. Sundari, J. M. Mathana, and T. S. Nagarajan, "Secured IoT Based Smart Greenhouse System with Image Inspection," *2020 6th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2020*, no. 978, pp. 1080–1082, 2020, doi: 10.1109/ICACCS48705.2020.9074258.