Monitoring of Agricultural Conditions in Greenhouse Crops Using Edge Computing and IoT.

Jesús Gil Ruiz

Universidad Internacional de La Rioja

Franklin Guillermo Montenegro

Universidad De Cundinamarca

Daissi Bibiana Gonzalez Roldan

Universidad Distrital Francisco José de Caldas: Universidad Distrital Francisco Jose de Caldas

Gabriel Vargas Monroy

Universidad Distrital Francisco José de Caldas: Universidad Distrital Francisco Jose de Caldas

Carlos Enrique Montenegro-Marin (✉ cemontenegrom@udistrital.edu.co)

Universidad Distrital Francisco Jose de Caldas <https://orcid.org/0000-0002-3608-7158>

Javier Parra Fuente

Universidad Internacional de La Rioja

Research Article

Keywords: Monitoring, IoT, agricultural, computing.

Posted Date: August 4th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-756345/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Abstract

This work presents an Internet of Things (IoT) system and edge computing for monitoring crop conditions by developing a system to collect information on parameters related to the crop weather conditions, this data is recollected with edge computing system. The data is sent to the server for processing, then forwarded to the user via the IoT protocols and procedures. The purpose is to collect data in real time for analysis and allow decision-making by the system and the farmer. The user can interact with the system remotely and receive the specified alerts and conditions. The initial results show that the system provides complete information on the status of the parameters, enabling management of greenhouse crops.

Introduction

carried out in agricultural production have historically played a fundamental role as the precursor of activities such as a sedentary lifestyle and the different forms of negotiation in society. They are now influenced by current progressive social and technological advances. Technology is vital to establish new requirements regarding the production and consumption of products, seek effective production according to the growth of cities, diversify commercial activities and consider the process of care and monitoring product development. Manual monitoring is consolidated through the use of labor, different methodologies, and highly complex processes. In the same way, it is necessary to propose new tools that allow the monitoring of products making analogous use of technology, leading to the use of tools such as those proposed by the Internet of Things (IoT) and Edge computing; elements that provide a practical implication for the realization of control processes, measurement, and care of the environment.

Problem Statement

Due to the decisions that need to be made in cropping and agricultural systems, the IoT and Edge computing must meet the following requirements:

- Modern hardware development for the IoT and Edge computing to collect data in different sensors based on crops' needs and comply with the security protocols of sending data in real time.
- The IoT should cover most of the plantation area.
- Fault tolerance.
- A system that shows the data in real time.
- Automation of reports and data control (dashboard).
- Data storage and backup for possible future applications such as machine learning and big data.
- Scalability and ease of use.

The next step is to generate an architecture model that solves the general problem and can be upscaled to use minimal energy consumption, data, and money.

General Objective

Analyse the environment of a group or a plant using the IoT and Edge computing tools, distribute the analysis through an Application Programming Interfaces, (API) that, based on the implementation of elements such as concurrent programming, allows the structuring of the project so that the process of analysis manipulation, and modification of the monitored information can be followed up in a controlled and efficient manner when performed simultaneously on the server.

Specific objectives

- Build a structure that allows the acquisition of information from the plants' environment using the different sensors.
- Structure an API, considering the different procedures necessary to guarantee the security and confidentiality of the transmitted information.
- Using the API, and with the client's construction uncoupled, monitor reading-service consumption through the network.
- Analyse the acquired information, ensuring adequate user representation.
- Implement security at different levels to adequately cover the information acquired by the sensors and the network transmission process.
- Utilise the user's comprehensible visualisation channel to manipulate, modify, download, monitor the plants' status, and verify the captured data in real time.
- Implement a service for the user to generate dynamic reports of the monitored elements.

Justification

The measurement and control of the environment and its different conditions are a process that is scarce in crops and greenhouses in Colombia in technical and technological terms. Automation is available for processes such as generating or collecting products derived from certain crops and animals. However, the current data collection process for the agricultural conditions of crops is extremely cumbersome and expensive, as humans resources are needed to carry out the measurement tasks leading to limited product development. The monitoring process is a useful element that could be used as a basic tool to support a complete system operating model through progressive automation processes and generate advantages for the care and study of products, progressively increasing the profits obtained from a crop or an animal pen.

State Of The Art

The process of agricultural care requires excessive use of human resources; these activities can be technologically managed. Processes and procedures to be managed are mainly those involving repetitive activities defined by conditions that are governed by a specific protocol. The monitoring carried out in product care could be used as a study subject in different instances, thus not relegating the application to a specific type of product, but rather developing automation implementation to correct current problems.

Progressive growth of the population has increased these needs. Urban areas in 2014 accounted for 54% of the total world population, up from 34% in 1960, and they continue to grow. The growth of the urban population, in absolute numbers, is concentrated in the less developed regions of the world. It is estimated that, in 2017, even in the least developed countries, most people live in urban areas. The world's urban population was estimated to grow by 1.84% per year between 2015 and 2020 and will increase by 1.63% per year between 2020 and 2025, and 1.44% per year between 2025 and 2030 (WHO,2020). This phenomenon is a precursor to the need for products to support these populations and the development of agricultural activities directly related to maintaining population distribution where commercial and economic activities follow their demographic disposition. It has been observed in several countries that the agricultural workforce is ageing and that adaptation of agricultural technologies and policies could help keep older people engaged in agriculture-related activities. The number of older people is expected to grow more rapidly in the next 15 years in Latin America and the Caribbean, with a projected increase of 71% in the population aged 65 and over, followed by Asia (66%), Africa (64%), Oceania (47%), North America (41%), and Europe (23%). Therefore, the need for growth in agricultural production rises. In the period 1991–2014, levels of agricultural investment increased in all groups of countries, albeit at different rates. In high-income countries, investment increased from around \$120 billion to \$190 billion, while investment in the remaining low- and middle-income countries grew from \$45 billion to \$115 billion.

As global temperatures rise, agricultural losses are likely. An example is Europe, where heat waves have become more frequent and intense, causing significant damage to agriculture and forests (ONU,2020). The monitoring process is necessary to maintain the quality of the environment where different products are developed, opening doors to plant population monitoring processes that include aspects such as

- Measurable: Can be registered and analysed qualitatively or quantitatively.
- Accurate: Can be similarly defined by anyone.
- Consistent: Always measures the same and does not change over time.
- Sensitive: Changes proportionally in response to actual changes in the entity measured.

That would allow the adequate control of different monitoring processes to be carried out on a population group of plants, leading to a monitoring process that requires the implementation of systematic data collection and analysis to explore trends and change its interest over time to provide relevant information for conservation management. Attributes that account for the viability of the species can be implemented; for example, attributes that evaluate whether populations can persist in the long

term while preserving their natural ecological and evolutionary dynamics (Lopez,2015). However, there are various limitations; the high use of human resources to measure conditions in a complete production area and their control depend on repetition intervals, and limitations leading to loss of information are also exposed in the study process that could be a determining factor in quality, effectiveness, and the development process of a product. Viñatigo Wineries is an example of implementing a monitoring process based on continuous data collection and structured storage within an information system. but also that its control depends on the repetition intervals with which said process is carried out, thus presenting various limitations in the complete study process of the conditions to which the product under study is exposed, leading to the loss of information that can be a determining factor in the quality, effectiveness and development process of a product. (Mendez,2006). In implementing this project, environmental measurement tools of less impact were used, beginning with treating the crop superficially and making the decision on whether to water or not. The University of Bari studied the influence of saline irrigation water on broccoli and the effect of water salinity in broccoli cultivation, making comparisons in seven different salinity levels obtained by mixing freshwater with seawater. Irrigation was applied at 30% depletion of available water to bring the soil to field capacity using an irrigation volume that included leaching (Gucci,1996). Micro-organisms were studied using an analogue method. The study's main objective was to explore the impact of the substrate concentration, and the anode area regulated the proportion between the morphological structure and the proportion of the anode's living cells (Gao,2021). The sensor in charge of measuring soil moisture used an Arduino UNO, calibrated through moisture measurement, taking a total of 16 samples to develop the relationship between gravimetric soil moisture and output voltage, and more samples were used to validate the developed relationship (Johri,2021), progressively recording the improvement of the sensors, academic studies, and generally deepening the structuring of the systems serving as tools.

In works such as (Zhang,2020), the need to use edge computing and IoT as a mechanism for data capture to continue data processing in agriculture is raised.

The fundamental objective of this research is to propose a new development architecture that is adaptable and allows ease of monitoring, constituted as an essential monitoring tool in production and process research. The main objective is data collection to implement active monitoring of environmental conditions. The conditions for analysis are as follows:

- Humidity
- Soil moisture
- Air quality (Gas concentration)
- CO2
- Temperature
- Ultraviolet light

The measurements will be taken by a service-oriented architecture. A design suitable for an architectural microservice system is crucial for its success. In distributed systems, the size of the component directly impacts non-functional requirements, such as performance, flexibility, and reusability (Schmidt,2020). Architectures such as Onion and Clean Architecture can be implemented as essential tools in building in a corporate environment. However, the characteristics of microservices cause them to have cross-platform compatibility and inconsistent call-standard problems in the development process (Zuo,2020). Therefore, communication tools are proposed by structuring as representational state transfer (REST) API.

Developing

Explanation of physical assembly

A Raspberry Pi will be used to manage the project readers, as it is a small, low-cost computer built with Arduino UNO to allow the management sending and progressive treatment of information collected by the readers.

The proposed structure for the project is data collection by the Raspberry Pi as the Arduino Uno reader will allow the implementation of measurements to maintain the conditions.

The Arduino Uno is an ATmega328P based microcontroller board. It has 14 digital input and output pins, six analogue inputs, a 16 MHz ceramic resonator, a USB connection, a power connector, an ICSP header, and a reset button.

DHT11 sensor

The DHT11 is a low-cost, medium precision relative humidity and temperature sensor. The supplied output is digital type using only 1 data pin.

The pins are defined as

- **VCC:** Feeding
- **I / O:** Data transmission
- **NC:** Does not connect, pin to air
- **GND:** Ground connection

This is how the assembly was given in such a way that it is connected:

Sensor MQ-135

This is an electrochemical sensor that varies resistance when exposed to certain gases. Its internal heater increases the internal temperature for the sensor to react with the gases causing a change in the

resistance value. Depending on the model, the heater may require voltage between two and five volts. It is used in air-quality control equipment for buildings and offices and is suitable for detecting NH₃, NO_x, alcohol, benzene, smoke, CO₂, and other gases:

The sensor's characteristics are:

- Working voltage: 5V.
- Main Chip: MQ-135 Air quality sensor.
- Double output: Analogue and TTL level output.
- Mounted on module with connection pins.
- Adjustable sensitivity with the potentiometer.
- Rank of detection: 10-1000ppm.
- Good sensitivity to harmful gases in wide range.
- High sensitivity to ammonia, sulphur, and benzene.

Its assembly is given as follows:

MIKROE-1630 sensor

The air quality sensor is suitable for detecting ammonia (NH₃), nitrogen oxides (NO_x), benzene, smoke, CO₂, and other harmful or poisonous gases that affect air quality. It contains an MQ-135 sensor unit and has a sensor layer made of tin dioxide (SnO₂), an inorganic compound that has lower conductivity in clean air than when polluting gases are present:

The characteristics are:

- Product: Add-on boards
- Type: MQ-135 Air quality
- Operating supply voltage: 5 V
- Series: Sensors-gas-click
- Brand: Mikro
- Product Type: Multiple function sensor development tools
- Commercial name: mikroBUS

DFROBOT sensor

This is an analogue soil moisture sensor for the Arduino. It can read the amount of moisture present in the soil around it. It is a low-tech sensor but ideal for monitoring an urban garden. The two probes pass

the current through the soil and then read the resistance to get the relative humidity level. With more water, the soil conducts electricity more easily. The characteristics are:

- Power supply: 3.3v or 5v.
- Output voltage signal: 0 4.2v.
- Current: 35 mA.
- Pin definition: Analogue output (blue wire), GND (black wire), power (red wire).
- Size: 60x20x5mm.
- Range of values: 0–300: dry soil, 300–700: wet soil, 700–950: in water.

The respective assembly:

GUVA P1918 Sensor

The sensor widens the spectrum of light to be detected. It uses a UV photodiode that can detect a light range of 240-370 nm. The photodiode signal level is very small, at the nano amp level, so an opamp is launched to amplify the signal to a more manageable volt level.

Characteristics:

- Dark Current I_D $V_R = 0.1 \text{ V} \rightarrow 1 \text{ nA}$.
- UVA Lamp, $1 \text{ mW} / \text{cm}^2 - 113 - \text{nA}$ Photo Current $I_{PD} 1 \text{ UVI} - 26 - \text{nA}$.
- ITC UVA Lamp imperative coefficient - $0.08 - \% / ^\circ \text{C}$ Responsivity $R_\lambda = 300 \text{ nm}$, $V_R = 0 \text{ V} - 0.14 - \text{A} / \text{W}$.
- Spectral detection range λ 10% of R 240 - 370 nm.

Considering the previously defined outputs, we have a definition of the respective power pins, data output, and GND, The assembly is shown below:

Safety mechanism explanation

The tools included in the IoT architecture process provide the system status for the basis of the current system.

The first implementation of the IoT is to listen on various communication channels with several ports specified in the MQTT protocols and TCP / IP solutions based on an HTTP format. Initially, it is necessary to consider the different tools used for the construction of the general project.

API: This uses:

- **JavaScript:** System health and security implementations under a simple control flow.

- **Python:** Everything will run under this technology. It is the communication and transaction language, making it concurrent and simple for the multiplatform system that condenses the proposed system.
- **Flask:** Minimalist philosophy technology for microservices conceived in code controlled by the developers and managed under current standards.
- **Arduino C ++:** Arduino offers technologies for low-level communication. In addition to Python, data programmed in the CORE of the system is implemented in an organised way in the microprocessor.

Client: This uses:

- **NodeJs:** Technology to provide simple front-end services and flow control in general.
- **AxiosJs:** Library of TCP / IP protocols with HTTP asynchronously with any server specialised in communication under the REST API architecture.
- **Chartjs:** Library for user monitoring and control.
- **ReactJS:** SPA technology for administration and control flows.
- **Django:** Modern Python technology to robustly display services.

Architecture

The architecture generates everything essential for the software solution because the decision was made to cover all the system's problems correctly.

Requirements

Due to the reporting and decision-making needs of plantations, or places where data collection take place, the system should meet the following requirements:

- Cover a large area within a plantation.
- Remain fault-tolerant.
- Show captured data in real time.
- Use software that automates data for reporting, and data control in a dashboard.
- Use software with intermediate security for the flow of the system.
- Data should be stored in a system with capacity for future big data and machine learning processes.
- The system must be easy to scale and be straightforward to control (graphically).

Proposed solution

The proposed software solution is arranged as follows:

Figure 17: General architecture design for alert and real-time data capture

The vast majority of the problems in the system requirements subsection have been considered, and the following architectural components are suggested:

- A service specialised in close communication with the hardware and data collection. This implies that the system must be installed in the kernel of a working system. The system will also send status alerts to warn of any failure.
- A service that will receive requests from the system, organise time, and measure and manage data in an organised way in a database ready for massive entries.
- A database management system that is scalable, ready for massive data input and output, and has capacity to serve in the future as a big-data tool.
- A text message and email alert system as elements of many other alerts, ready to generate and send in a correct and efficient manner.
- A front-end system capable of showing the results of the SGDBD (database management system) that demonstrates simple interaction for the user to obtain reports and verify the status of the system. The interface security must have sole user control.
- A system that periodically activates system events and ensures the data obtained is stored so that massive data can be transferred efficiently.

Scope and limitations

One of the characteristics of constructing a complete solution is the relationship between devices and software in decision-making relates to the convergent process of a holistic solution that highlights the relationship.

A problem in hardware solutions is that possible damage can lead to incorrect data. A technician's presence to correct problems helps the toleration of hardware errors. Another problem is a poor internet connection, indicating that an offline relationship should be modelled to allow for delayed sending of data, and software that carries a dynamic solution. The diagram memory should point solutions to their definitive software and relate them to each other and the diagram that they share with the model.

Another problem that can arise is errors around system modelling for massive data control.

Several situations were found. The software can reduce its optimisation relative to the visual, showing that the characteristics provided in the development environment must be arranged in a high-scale environment determined by the administrator (client).

Cost–benefit relationship

An approximation of the cost of processing in greenhouses would bring the control of data regarding mass production closer, implying that the product will provide value that will lead to improvement in high

impact production and ensure a constant and homogeneous production value.

Data in relation to the cost information is provided as follows:

Table 1: Control of data related to improvement

Cost Type	Cost USD	Mode
Controller	\$ 158	Permanent
Reader	\$ 8	Permanent
Licence	\$ 52	Permanent
Deployment	\$ 26	Annual
Infrastructure	\$ 26	Permanent
Medium	\$ 52	Biannual
Internet access	\$ 13	Monthly

Conclusions

- The security processes are vital as they will guarantee the information's reliability and confidentiality. Significant importance is given to the data encryption process, making it essential to highlight libraries and work frameworks that allow managing the process in a dynamic and highly secure way by using OpenSSL to generate TSL / SSL security.
- In constructing an electronic system, it is essential to consider the requirements of the elements, as the system's stability and the connection process depend on this when using the management controller as an extension of the team or if they are implemented as a separate measurement system.
- For data analysis generation, it is essential to consider the information acquisition process. In visualising the data, the information will influence processes included in the correlation of environmental condition comparisons.
- Security should be installed at different levels of the service when implementing a security system. It is essential to consider the environment where this process will occur and the hardware and software elements necessary for its implementation with IoT.
- Architecture-oriented microservices provide a high degree of scalability for the construction and implementation of software that optimally allows the system's adaptability, provides independence between its functionalities, and ensures its operation in different conditions to those exposed.
- Using the architectures of microservices and services used in conjunction with development philosophies, such as Onion Architecture or Clean Architecture, allows the development and implementation of larger work teams to effectively restructure logic and functionalities.

References

Adafruit *Analog UV Light Sensor Breakout - GUVVA-S12SD*. <https://www.adafruit.com/product/1918>

Arduino. *Arduino Uno Rev3*

<https://store.arduino.cc/usa/arduino-uno-rev3>

Communications. *Advances in Intelligent Systems and Computing*, vol 1165. Springer, Singapore.

https://doi.org/10.1007/978-981-15-5113-0_55

Cucci, G and De Caro, A. (1996) Influence of saline irrigation water on broccoli. Acta Hortic. 407, 339-346.

<https://doi.org/10.17660/ActaHortic.1996.407.43>

Datasheet *GUVA-S12SD*.

<https://cdnshop.adafruit.com/datasheets/1918guva.pdf>

Datasheet Notes d'applications modules Click Board™ et Arduino™.

[https://www.lextronic.fr/lextronic doc/Click, App TOME1.pdf](https://www.lextronic.fr/lextronic%20doc/Click,%20App%20TOME1.pdf)

Gao, Y., Wang, S., Yin, F., Hu, P., Wang, X., Liu, Y., & Liu, H. (2021). Enhancing sensitivity of microbial fuel cell sensors for low concentration biodegradable organic matter detection: Regulation of substrate concentration, anode area and external resistance. *Journal of Environmental Sciences (China)*, 101, 227–235. <https://doi.org/10.1016/j.jes.2020.08.020>

Johri A., Anchal, Prakash R., Vidyarthi A., Chamoli V., Bhardwaj S. (2021) IoT-Based System to Measure Soil Moisture Using Soil Moisture Sensor, GPS Data Logging and Cloud Storage. In: Gupta D., Khanna A., Bhattacharyya S., Hassanien A.E., Anand S., Jaiswal A. (eds) *International Conference on Innovative Computing and*

Kumbhar P., Barbade S.R., Jain U.H., Chintakind C.L., Barhanpurkar A.H. (2021). Implementation of Black Box System for Accident Analysis Using Raspberry Pi. https://doi.org/10.1007/978-3-030-51859-2_47

Lopez, C. (2015). *Monitoreo de poblaciones de plantas para conservación*
<https://grupoeeco.org/gallery/lopez-gallego,c.%202015%20-%20manual%20-%20monitoreo%20de%20poblaciones%20de%20plantas%20para%20conservacion.pdf>

Martin, R. C. (2017). *Clean Architecture: A Craftsman's Guide to Software Structure and Design*. Prentice Hall, 432. <https://www.amazon.com/Clean-Architecture-Craftsmans-Software-Structure/dp/0134494164%0Ahttps://www.safaribooksonline.com/library/view/clean-architecture-a/9780134494272/>

Mendez, J. (2006) *Viticultura de precision: Una radiografia para el diagnostico*.

http://www.acenologia.com/ciencia73_03.htm

Moussa M., Benachenhouch A., Belghit S., Adda Benattia A., Boumehdi A. (2021) An Implementation of Microservices Based Architecture for Remote Laboratories. In: Auer M., May D. (eds) *Cross Reality and Data Science in Engineering*. REV 2020. *Advances in Intelligent Systems and Computing*, vol 1231. Springer, Cham. https://doi.org/10.1007/978-3-030-52575-0_12

Mouser *MIKROE-1630*.

<https://co.mouser.com/ProductDetail/Mikroe/MIKR1630?qs=QhAb4EtQfbUP1yCGjXhiRA==>

Naylamp *Sensor de temperatura y humedad relativa DHT11*. Disponible

en: <https://naylampmechatronics.com/sensores-temperatura-y-humedad/57-sensor-de-temperatura-y-humedad-relativa-dht11.html>

Naylamp *Tutorial sensores de gas MQ2, MQ3, MQ7 y MQ135*. Disponible

en: <https://naylampmechatronics.com/blog/42Tutorial-sensores-de-gas-MQ2-MQ3-MQ7-y-MQ13.html>

Programa fácil. Como utilizar el sensor DHT11 para medir la temperatura y humedad con Arduino.

<https://programarfácil.com/blog/arduino-blog/sensor-dht11-temperatura-humedad-arduino/>

Schmidt, R., & Thiry, M. (2020). Microservices identification strategies : A review focused on Model-Driven Engineering and Domain Driven Design approaches. *2020 15th Iberian Conference on Information Systems and Technologies (CISTI)*, 1-6.

<https://doi.org/10.23919/CISTI49556.2020.9141150>

ONU. (2020). Food and Agriculture. *The future of food and agriculture: Trends and challenges*

http://www.fao.org/3/a-i6583e.pdf#i6583EN_2017-11.indd%3A18727%3A279

ONU. (2020). *World Economic Situation Prospects*. https://www.un.org/development/desa/dpad/wp-content/uploads/sites/45/WESP2020_FullReport.pdf

World Health Organization. *Urban population growth*.

(2020) https://www.who.int/gho/urbanhealth/situation_trends/urban_population_growth_text

Zhang, X., Cao, Z. and Dong, W. (2020). Overview of Edge Computing in the Agricultural Internet of Things: Key Technologies, Applications, Challenges, in *IEEE Access*, vol. 8, pp. 141748-141761.

<https://doi.org/10.1109/ACCESS.2020.3013005>.

Figures

Figure 1

Gross fixed capital formation (GFCF) in agriculture, by group of countries and region, 1990–2015 (ONU,2020).

Figure 2

Clean Architecture (Martin,2017).

Figure 3

Raspberry Pi

Figure 4

Arduino controller assembly

Figure 5

Arduino Uno (Arduino,s.f.)

Figure 6

DHT11 sensor structure (Naylamp, s.f.)

Figure 7

Mounting the DHT11 sensor (Programa fácil, s.f)

Figure 8

Air quality sensor (Naylamp tutorial, s.f.)

Figure 9

MQ135 assembly, for air quality measurement.

Figure 10

Air quality and CO2 sensor (Mouser MIKROE-1630, s.f.).

Figure 11

Air quality and CO2 sensor assembly

Figure 12

Soil moisture sensor (Datasheet Notes d'applications modules Click Board™ et Arduino, s.f)

Figure 13

Floor sensor mounting

Figure 14

UV Light Sensor (Adafruit Analog UV Light Sensor Breakout - GUVVA-S12SD, s.f.)

Figure 15

UV light sensor mounting

Image not available with this version

Figure 16

Figure 17

General architecture design for alert and real-time data capture