Research Article

Received: 18 February 2023 / Accepted: 21 April 2023 / Published: 5 May 2023

Design of a Mobile Application Used for Monitoring the Drip Irrigation System in Coffee Crop Using CDIO Methodology

Johan Julián Molina Mosquera

Juan Pablo Tierradentro Aya

Pedro Alexander Contreras Andrade

Electronic Engineering Program,
Faculty of Engineering,
Universidad Surcolombiana,
Colombia

DOI: https://doi.org/10.36941/ajis-2023-0058

Abstract

This article will develop the design and integration of an automated drip irrigation system prototype for a coffee crop of Robusta species (Coffea Canephora) with CDIO methodology. It will be implemented using Open Source elements, with an ESP32 processing card, a mobile application developed in Android Studio for monitoring conditions in real-time, including ON/OFF irrigation control, a Water Solenoid Valve that will allow irrigation and sensors for humidity and temperature.

Keywords: *Mobile Application, Measurement, Control, Automatization, Sensors, Moisture, Temperature*

1. Introduction

The conception of this project was considered because of the issue evidenced by "Hacienda El Santuario", a farm located in Tolima, Colombia, which has a coffee crop and wants to implement drip irrigation, with the limitation that there is not a person who is permanently monitoring the crop and its irrigation hours, also, does not have the capacity to determine compliance with the necessary humidity conditions for the crop previously mentioned.

After analyzing the farm's environment, it installs a manual drip irrigation system; therefore, it is necessary to open the water tap to irrigate the coffee crop. Drip irrigation is a method of slow and frequent irrigation of water drops that keeps the crop's roots moist and does not use high pressure; this method saves up to 40% of water resources (LA República, 2013).

The objective of this project was the design and implementation of a mobile application that allows real-time monitoring, automation, and control of a drip irrigation system applied to a coffee crop of Robusta species (Coffea canephora), that, according to the Technical Manual for the Production of Robusta Coffee, the conditions in which the crop must be grown are: temperature between 22°C to 30°C, ground moisture between 75% to 85% (Ing. Jaime López et al., 2016). In this way, it can resolve the problem in the farm where it is grown since this process was done manually

and under observation. At the same time, with the devices and the application to achieve the implementation, it is intended to be automatic and remote. This project comprises sensor devices for environmental conditions, an ESP32 data acquisition card in charge of the operation, and the interpretation of the values measured by the sensors used through the WIFI protocol to a database. Additionally, a mobile device with an Android operating system. The aim is to monitor the environmental conditions of the crop through a portable and easy-to-use application, which allows efficient crop irrigation with the drip irrigation technique.

2. Project Stages

2.1 Conceive

The Drip irrigation technique consists of conducting water through a pipes system containing emitters that allow water drops to fall periodically through drippers (Tec. Hidráulico Mario Liotta INTA: Resp Riego y Drenaje INTA - Esp et al., 2015).

The Internet of Things (IoT) refers to a wide range of smart objects based on sensors. It means they can connect to the network, and they stay under control by microcontrollers with the ability to process, interpret information, and store information about different environmental variables can be stored, such as temperature, pressure, light, humidity, soil moisture, PH, among others, and thanks to new technologies in communications, applications can be made in real-time (Placidi et al., 2020).

For example, in agricultural applications, a real-time monitoring system can help optimize the use of resources, such as water and fertilizers, resulting in lower costs and greater efficiency. Another advantage of the connectivity of components to a server in real-time is the ability to access data and control devices from anywhere with an internet connection, which provides greater flexibility and control in the management of processes, with all this can achieve resource efficiency in the agricultural sector, which is currently in high demand due to climate change(Saab et al., 2019).

Considering the problem described above, it is decided to develop an electronic system that allows this irrigation in an automated way, in which, through this system, the valve that allows the passage of water is opened and closed. For this purpose, the use of a water solenoid valve is employed, which has a diameter of ½ inch, and operates at 12V, that is, when this voltage is applied the valve opens allowing the passage of water, and when this voltage is turned off it closes, flows in one direction only, and the minimum flow pressure is 0.02 MPa to 0.8 MPa (e-Gizmo Mechatronix Central, 2016), the idea is that by utilizing humidity and temperature sensors, the crop is being monitored, and through certain conditions at which the registered variables are found, the valve is indicated when it has to open or close.

Figure 1: Water Solenoid Valve 1/2" 12V DC

According to the variables to control, a humidity and temperature sensor is employed to connect to the controller. Then the respective conditions processing can be implemented to operate the solenoid valve. Regarding the Technical Manual for the Production of Robusta Coffee, the conditions in which the crop must be grown are: temperature between 22°C to 30°C, ground moisture between 75% to 85% (Ing. Jaime López et al., 2016) In this case the sensor to be used is the DHT22 for the ambient temperature, it also measures the relative humidity of the environment, it works with DC power supply from 3V to 6V, measuring temperatures from -40°C to 80°C with an accuracy of ± 0.5°C and resolution of 0.1°C. In terms of humidity, it manages to capture from 0 to 100% relative humidity, with an accuracy of 2-5% and a resolution of 0.01% (Mobaraki et al., 2020).

Figure 2: DHT22 sensor

Previously the DHT22 sensor was described. In this case, it will be used for temperature measurement since for moisture, it has been considered to take it from the soil. In order to achieve this purpose, two analog capacitive sensors were implemented, the Capacitive Soil Moisture sensor v1.2. It has an operating voltage between 3.3 to 5.5V and an output voltage between 0 and 3V. The sensor has a recommended depth mark on its infrastructure (Placidi et al., 2020).

Figure 3: Soil Moisture Sensor

For the remote monitoring of the operation of this system, the decision was to develop a mobile application, using Android Studio, which allows a smartphone, to review the variables that are being monitored and check if the valve is open or closed in addition to a switch button that allows deactivating the system in case of any issue. This application was configured for an Android with API 30, and the design was made using JAVA as a programming language and XML for the graphic design (Girones & Mauri, 2020).

Finally, the ESP32 is used as a microcontroller, which is an SoC (combined chip as mentioned in the official documentation) that provides connectivity (Wi-Fi and Bluetooth, at 2.4 GHz), so that data can be sent to a database in real-time and thus enable the operation of the mobile application and the complete system, support for multiple communication interfaces, support for low power operation and hardware blocks dedicated to security in a single chip. This has a 5V supply voltage, 3V input/output voltages, 24 pins, some input only, and two 12-bit ADCs (Bertoleti, 2019).
It is expected to eventually produce a product as follows:

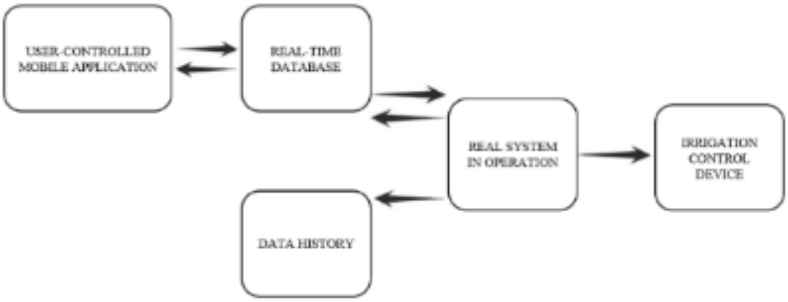


Figure 4: Client-server model in design

2.2 Design

Next, the design of software and hardware needs to be implemented. This design has been done as follows:

- BACKEND

The first step is to configure the Arduino IDE to recognize the ESP32. Moreover, add the necessary libraries to make the WIFI connections, the connection to the DHT22 sensor, the Firebase_ESP_Client library that makes the connection to Google Firebase real-time database (RTDB), NTPClient to connect to an NTP (Network Time Protocol) server to get the current time, and TimeLib to handle dates and times.

It is required to define the pins to be used, which will be the 16 pins for the control of a relay, which will be handled to make the switching to the solenoid valve, pin 36 to make the connection with the capacitive humidity sensor, which must necessarily be a pin with ADC and the pin for the temperature sensor DHT22 will be the number 4, also the WIFI credentials and the URL address for the connection with firebase.

As the code is developed in the Arduino IDE, it consists of two parts, one in the setup() function and the other in the loop() function, which works as follows:

In the setup() function, the code connects to the specified Wi-Fi network, then connects to the Firebase RTDB database with the API_KEY and database URL, and finally connects to the NTP server to get the current time. This is used to create a timestamped log of readings.

In the loop() function:

- It utilizes the update() function of NTPClient to update the time.
- Reads the humidity and temperature values from the DHT22 employing the DHT library, as

it is a digital sensor it does not need calibration.
- Reads the soil moisture value from the humidity sensor.
- Uses the Firebase RTDB function to get the "Power" value (indicating whether the system is On, or Off).
- It uses the map() function to calculate the percentage of soil moisture from the value read from the sensor, since it is an analog output, it must be calibrated so that the information can be well received by the microprocessor, for this two reference values are taken, which are the values read by the sensor in the open air and those read in water, which were 2600 and 1000 respectively, so it is conditioned that when the sensor detects the soil moisture, it shows 100% humidity, and when it detects the dry ground it shows 0% humidity.
- If the system is ON, and the soil humidity is less than 75%, it activates the relay to turn on the water solenoid valve.
- If the system is ON, and the soil moisture is greater than 85%, it turns off the relay to stop the water solenoid valve.
- If the system is OFF, it turns off the relay.
- Uses the Firebase RTDB function to update the "Water" value (indicating whether the water solenoid valve is ON, or OFF).
- Prints the humidity, temperature, time, and date values on the serial monitor.
- Uses a 500ms delay, before re-running the code in the loop.

To activate the water solenoid valve it requires to be energized from an external source, a power stage consisting of a small 5V relay module must be used, which has a 12V transformer connected to the normally open pin to power the water solenoid valve. In order to, when it needs to be activated and let the water pass, the switching is done and allows the passage of energy.

For the activation of this module, it was necessary to make an extra adjustment, considering that the output signal of the microcontroller control pin is only 3.3V, and the relay module needs 5V for its activation. This adjustment consisted of using a reference 2N3904 NPN bipolar junction transistor (BJT) in a common emitter configuration and operating in switching. To the base supply, the output pin of the ESP32 is connected, which is a Vi=3.3V when active and Vi= 0V when deactivated, generating the cutoff zone. For the Vcc, we use the 5V coming out of the power pin of the microcontroller connected to the control pin of the relay module, at the same time, is connected to the transistor collector pin, and the emitter is the common ground. Therefore, when the base voltage is activated, it enters into saturation because it has more than the voltage needed to achieve the base-emitter voltage, which is 0. 7V, and this makes contact with the emitter ground, allowing the collector current flow, which in turn energizes the relay module with the 5V Vcc and are necessary for its operation, this voltage will remain fixed at any level of base current (Boylestad & Nashelsky, 2009, pp. 140–141).

The relay control signal is conditioned to the moisture variable measured by the sensor. The control signal will be active when the moisture in the soil is less than 75% and will stop when it reaches 85%.

As regards the database, it will be done through Google Firebase, since it is a service that offers the use of real-time databases created by the user, in addition to being a resource used by the Universidad Surcolombiana in its use of Google, in this case, the Realtime Database option is used, which allows changes in hardware status to be reflected in the mobile application created in a matter of milliseconds, and the Firestore is used to make a history of the data obtained by the sensors, such as soil moisture, environment humidity, and environment temperature, in addition to the status of the solenoid valve.

- FRONTEND

Android Studio is employed for the mobile application, configured with API 30, with XML graphical interface and Java operability, making it functional for Android 11 onwards.

The API30, launched in September 2020, incorporated improvements in the user interface, access to payment cards, home automation devices, improved permissions management, and the use

of Bluetooth even with airplane mode activated (Girones & Mauri, 2020)

The API was chosen considering the technical specifications of the developers' mobile devices. An API is the Application Programming Interface. In the case of Android, it is an integer value that represents the API version of the framework of the Android version of the device, which is used by the applications to interact with the Android software, which contains attributes, elements, packages, and categories. The APIs are updated in accordance with the version of Android but preserve the compatibility between the new API level and the previous versions. (*Desarrolladores de Android | Android Developers*, n.d.)

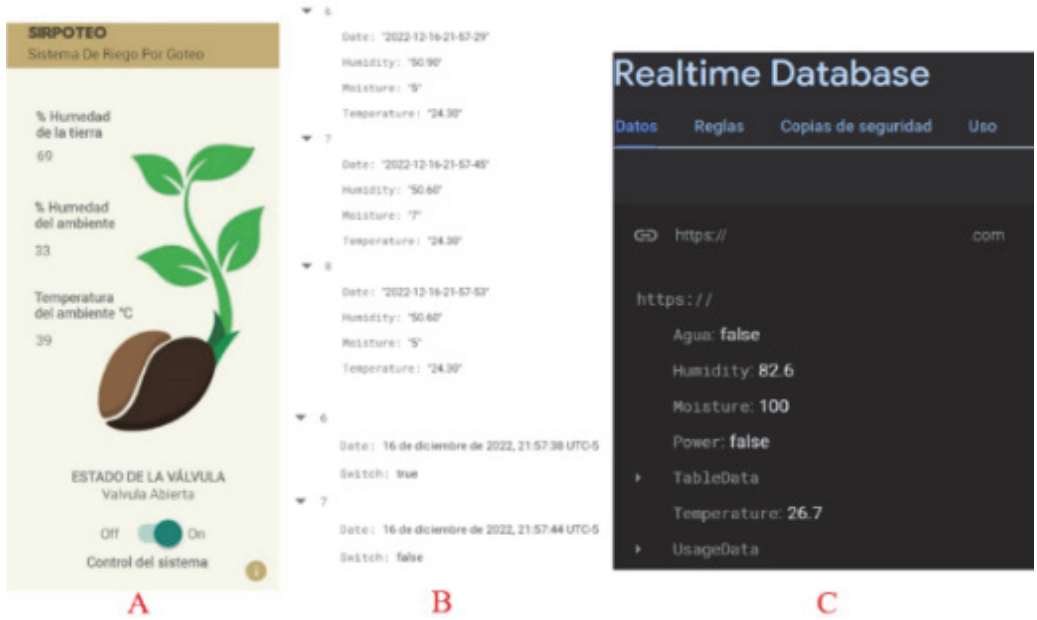


Figure 5: A) Frontend of the mobile application; B) Database recording the measured variables; C) Real-time database for status changes, and data update in the mobile application.

In the Frontend there is an interface that allows visualizing the soil moisture, environment humidity and environment temperature, it also shows the status of the valve, and a switch button which has the function of stopping or activating the drip irrigation system in case of any emergency revision in the plants or for maintenance needs. The interface, in its different states, is as follows:

The interface and communication are achieved thanks to the compatibility that Google Firebase has with Android Studio through a service in firebase called FirebaseDatabase. The data is updated using an asynchronous detector listening to the reference. This detector is activated every time the data changes. For reading data from Android Studio, the DatabaseReference instance is added, and to display them in the application, it is combined with setValue. The data can be of type String, Long, Double, Boolean, List<object>, Map<String, Object> (Google Inc., n.d.).

A printed circuit board (PCB) was designed to protect the essential parts of the system and keep the ESP32 and relay module tidy. In addition, the connection pins for the sensors were located and connected through terminal blocks to facilitate preventive and corrective maintenance without the need to solder or desolder components. In this way, the maintenance process is more straightforward since only the terminal blocks need to be loosened or adjusted.

For the design process of the Printed Circuit Board (PCB), the optimal distribution of the components was taken into account to ensure good performance, in the same way, it is ensured that

the board is easy to assemble and repair. When arranging the components on the board, it is important to take into account the space restrictions and the electrical connection, the design was made in such a way that the components are distributed so that they do not interfere with each other, thus avoiding possible short circuits or electromagnetic interference that may affect the operation of the device. In addition, the components were arranged in such a way that the inputs and outputs are located on only one side of the board, in this case on the bottom side, to facilitate the connection with other components and systems, since these in this case are connected at the bottom of the PCB, which can be useful in applications that require a complex interconnection, and the power supply was located on the right side of the Printed Circuit Board.

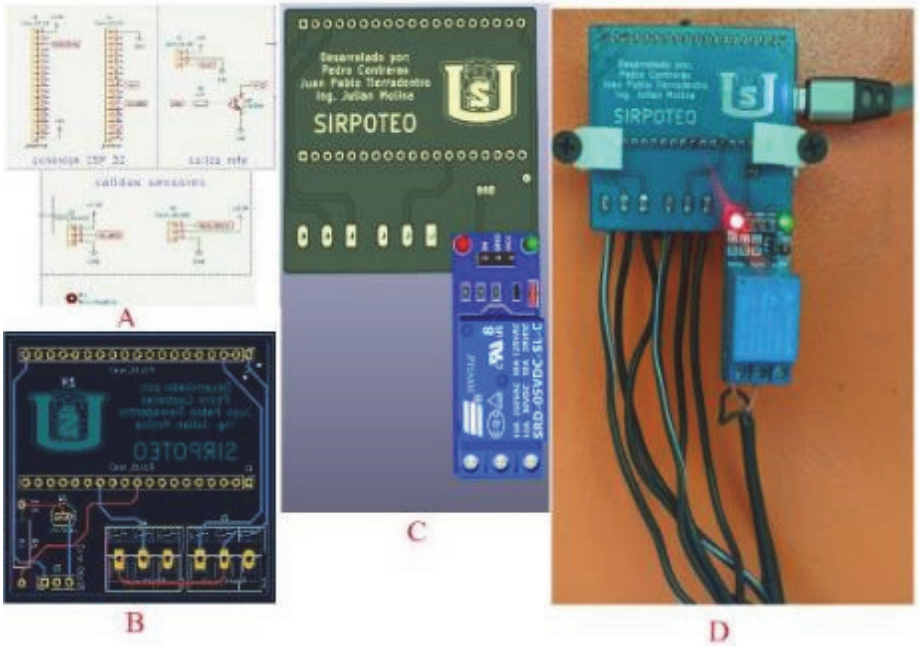


Figure 6: A) PCB design using the KiCad program; B) Finished PCB model; C) 3D model of the finished PCB; D) PCB implemented inside the system

2.3 Implement

For the implementation of the system described above, different tests were performed to verify the correct connectivity and data transmission between the mobile application, the database, and the physical system, to determine its proper integration and general operation. For these tests, the described sensors, the ESP32 board, the water solenoid valve, and the calibration and error correction processes were carried out in a controlled environment using a protoboard.

Nonetheless, the next step was to implement the previously mentioned in the natural environment. To achieve this, a printed circuit board (PCB) was designed, and the corresponding assembly of the elements was made, creating the prototype of the electronic product, hardware, and its respective mobile application.

The PCB was placed inside a metal box to protect it from environmental conditions and not leave it outdoors. The inside part was coated with expanded polystyrene to prevent overheating of the box material that could affect the devices it contains. For the different electrical connections,

subway pipes were used, which included both; the cables that go to the sensors and those that go to the outlets for the power supply of the system, in addition to preventing the wires from coming into contact with the soil of the crop, avoiding humidity and corrosion. The sensors, due to their material and working conditions, are resistant to environmental conditions. However, even so, additions of acetic silicone were made, which is used to fix objects that are outdoors without interfering with its operation. To close the box and prevent the entry of insects, a plastic mesh was used to avoid interference in communications and to prevent the access of these unwanted visitors.

The following budget was used for this project, in Colombian Pesos (COP) and U.S. Dollars (US$)[1].

Table 1: Cost Table

Materials	Costs (COP)	Costs (US$)
Printed Circuit Board (PCB)	$ 50.000	$ 10,34
Water Solenoid valve 1/2	$ 30.000	$ 6,20
Source 12V 2A	$ 16.500	$ 3,41
Source 5V 1A	$ 10.000	$ 2,07
Jumpers	$ 1.000	$ 0,21
ESP-32 Module	$ 42.000	$ 8,68
5V Relay Module	$ 6.000	$ 1,24
Capacitive Humidity Sensor	$ 50.000	$ 10,34
DHT22 Sensor	$ 28.500	$ 5,89
BJT Transistor	$ 200	$ 0,04
Metallic Box	$ 25.000	$ 5,17
Piping and Electrical Installations	$ 41.800	$ 8,64
Total	$ 301.000	$ 62,22

2.4 Operate

Once the implementation of the system was completed; hardware, software, and the respective mobile application proceed to verify the correct operation of the irrigation automation, that is, that irrigation was automatically activated according to the required conditions, which are ground moisture below 75%, and environment temperature of more than 35°C, moreover, the operation of remote activation and deactivation of the system was reviewed, since it could be controlled from more than 260 km away between the crop and the developers of the application.

The developed application works as follows:

[1]*Dollar according to the market rate in Colombia on February 28th, 2023*

Figure 7: System operation A) Application name; B) Measured variables; C) Valve status indicator; D) Irrigation on/off control; E) Author information; F) Application logo; G) Developers.

This project was implemented using the following client-server model, which is the way the system works, and the interactions between the frontend, and the backend.

Figure 8: Client-Server Model implemented.

The physical system was assembled with all the necessary adjustments for the crop, these adjustments consist of the metal box which contains the printed circuit and its accessories, piping and electrical installations for the power supply of the system, and the extra protections against the environment that could generate corrosion, as shown below:

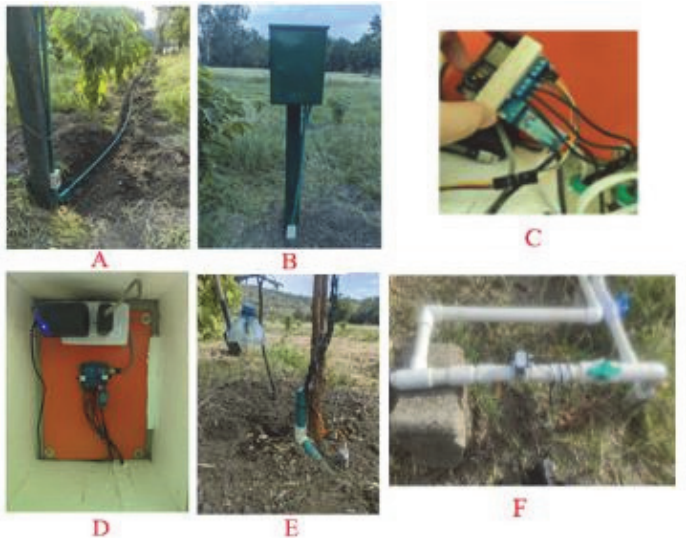


Figure 9: A) Underground electrical piping; B) Metal box which contains the circuit; C) Circuit connections; D) The interior of the metal box; E) Sensors in contact with the crop; F) Implementation of the solenoid water valve in the water pipe.

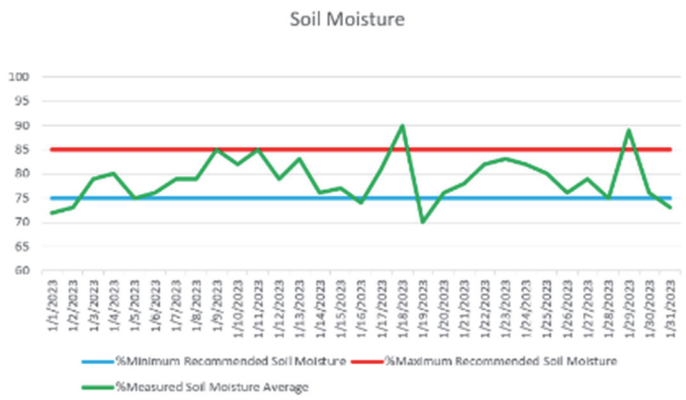
2.5 Analysis from Results

It is relevant to highlight that Robusta coffee (Coffea canephora) crop is influenced by a variety of factors, some of which have been monitored in this specific project. In particular, the environment humidity, and temperature have been measured, as well as the soil moisture, which are fundamental factors for the optimum development of the coffee crop. The results obtained in this study refer to the period of time between January 1, 2023, and January 31, 2023.

To measure the effectiveness of the drip irrigation system used in coffee cultivation, soil moisture measurements were compared with the standard values recommended for this species. These recommended values are between 75% and 85%. Additionally, it was evaluated whether the environmental temperature conditions were maintained within the recommended range of 22°C to 30°C for the coffee crop.

Measurements results are presented in this section and analyzed to determine the effectiveness of the drip irrigation system in maintaining optimal conditions for coffee crop growth. Possible implications of the results for the implementation of irrigation systems in other crops or environments are also discussed.

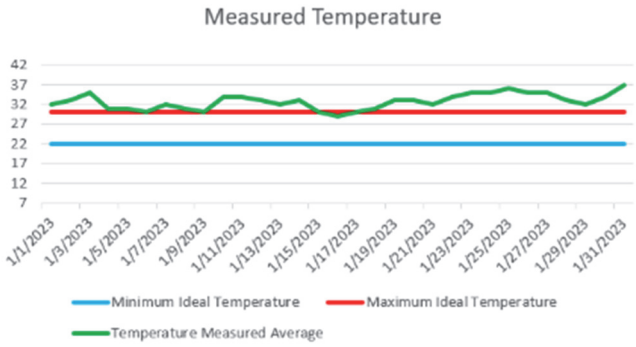
Starting by checking the soil moisture, which is the primary variable to be controlled, which is between 75% and 85%, the results are below:



Graph 1: Obtained Soil Moisture

Graph 1 shows that soil moisture conditions were adequately controlled, maintaining the average between the expected limits, according to expert recommendations. It can be observed that on some days, there were abrupt variations in the measurements. This is due to failures in the electricity supply and various changes in the weather conditions in the area of the crop.

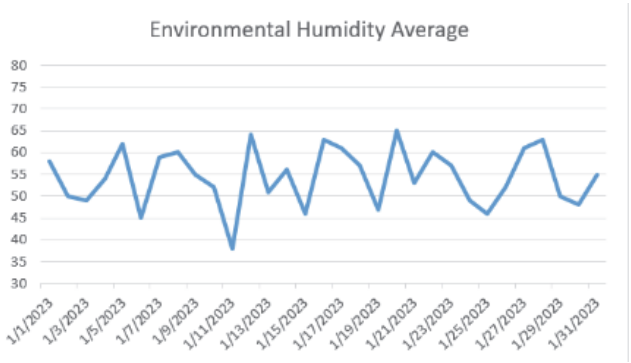
Following the procedure, temperature variations (°C) were reviewed and got obtained as follows:



Graph 2: Measured Temperature

Graph 2 shows that temperatures measured in the coffee crop were outside the ideal ranges, which is attributed to the fact that the crop is grown outdoors and is not in a controlled environment, such as a greenhouse, where the temperature could be regulated. The data recorded for this variable corresponds to the climatic conditions of the area where the crop is located.

Finally, among the variables measured was the humidity of the environment, from which the following results were obtained.



Graph 3: Environmental Humidity Average

As for this humidity in the environment, there was no scale of optimum values for the crop according to the consulted source[2], hence the data obtained from the sensor are shown, in which there is a variable behavior according to the climatic conditions of the geographical area in which the crop was planted.

Nonetheless, it was also found that this prototype can be reused in other crops. For this fact, there are several factors to consider before proceeding. First, the system will need to be adapted and calibrated to meet the specific irrigation needs of each crop. Moreover, climatic and soil conditions may vary significantly among different crops, which may require additional adjustments to the system. In summary, although it is possible to reuse the drip irrigation monitoring and control system in other crops, careful evaluation and proper adaptation is required before performing.

3. Conclusions

Implementing a drip irrigation control and monitoring system using an ESP32 with a Wi-Fi module in a coffee crop has several advantages, among them are:

Water savings: By using a drip irrigation system, the amount of water needed for the crop can be reduced as water is applied more efficiently and in precise amounts. The control and monitoring system allows adjusting the amount of water according to the needs of the crop and thus avoiding excessive water use.

Time and labor savings: The drip irrigation control and monitoring system allows irrigation to be automated, reducing the need for manual supervision.

Increased efficiency: The irrigation control and monitoring system reduces operating expenses for water, labor, and energy use since only the right amount is applied for the crop and helps improve crop quality by keeping the soil in optimal conditions, making the crop more profitable.

References

Bertoleti, P. (2019). Proyectos com ESP32 y LoRa - Pedro Bertoleti. https://books.google.es/books?hl=es&lr=&id=DoioDwAAQBAJ&oi=fnd&pg=PT4&dq=esp32+datasheet&ots=mBouAloIph&sig=9Nru6tn2UTKVowMao2Wpm2JGk40#v=onepage&q=esp32 datasheet&f=false

Boylestad, R. L., & Nashelsky, L. (2009). Electrónica : teoría de circuitos y dispositivos electrónicos (10th ed.). Pearson Educación.

[2] *Manual Técnico para la Producción de Café Robusta 2016*

Desarrolladores de Android | Android Developers. (n.d.). Retrieved January 2, 2023, from https://developer.android.com/guide/topics/manifest/uses-sdk-element?hl=es-419

e-Gizmo Mechatronix Central. (2016). ZE-4F180 12V WATER SOLENOID VALVE. https://uelectronics.com/wp-content/uploads/2019/04/ZE4F18012VWatersolenoidvalve.1387394297.pdf

Girones, J. T., & Mauri, J. L. (2020). El gran libro de Android (9th ed., Vol. 21, Issue 1). Marcombo, S.L.

Google Inc. (n.d.). Lee y escribe datos en Android | Firebase Documentation. Retrieved July 27, 2022, from https://firebase.google.com/docs/database/android/read-and-write?hl=es-419

Ing. Jaime López, Ing. Marvin Rodríguez, Ing. César Barrera, Ing. David Makepeace, & Ing. José Guzmán. (2016). Manual Técnico para la Producción de Café Robusta. Agosto. https://www.anacafe.org/uploads/file/283f6fd107ef4ce38af855880c47c49d/Manual-Cafe-Robusta.pdf

LaRepública, P. (2013, September 23). Riego por goteo permite ahorros de un 40% en el consumo de agua. https://www.larepublica.co/archivo/riego-por-goteo-permite-ahorros-de-un-40-en-el-consumo-de-agua-2063351

Mobaraki, B., Komarizadehasl, S., Javier, F., Pascual, C., Antonio, J., & Galant, L. (2020). Determination Of Environmental Parameters Based On Arduino Based Low-Cost Sensors Low-cost long-term Structural Health Monitoring of Bridges View project evaluation of a structure by visibility method and building its BIM model View project Determination Of Environmental Parameters Based On Arduino Based Low-Cost Sensors. Recent Trends in Construction Engineering and Education. https://www.researchgate.net/publication/344192844

Placidi, P., Gasperini, L., Grassi, A., Cecconi, M., & Scorzoni, A. (2020). Characterization of Low-Cost Capacitive Soil Moisture Sensors for IoT Networks. Sensors 2020, Vol. 20, Page 3585, 20(12), 3585. https://doi.org/10.3390/S20123585

Saab, M. T. A., Jomaa, I., Skaf, S., Fahed, S., & Todorovic, M. (2019). Assessment of a smartphone application for real-time irrigation scheduling in mediterranean environments. Water (Switzerland), 11(2). https://doi.org/10.3390/w11020252

Tec. Hidráulico Mario Liotta INTA: Resp Riego y Drenaje INTA - Esp, R. C. P. R., Riego, I. N. C. P. C. R. y D., & equipo de Riego INTA. (2015). Manual de Capacitación RIEGO POR GOTEO. https://repositorio.inta.gob.ar/bitstream/handle/20.500.12123/4528/INTA_EEASanJuan_Liotta_Riego_por_goteo.pdf?sequence=1