Development of transducer management module for irrigation in precision agriculture based on internet of things

TIAGO DA SILVA ALMEIDA, MURILLO VINÍCIUS DOURADO BASTOS
WARLEY GRAMACHO DA SILVA, RAFAEL LIMA DE CARVALHO
Universidade Federal do Tocantins
Department of Computer Science
Avenida NS-15, ALCNO-14, Palmas / TO
BRAZIL
tiagoalmeida@uft.edu.br, murillobastos@uft.edu.br
wgramacho@uft.edu.br, rafael.lima@uft.edu.br

Abstract: The objective of this paper is the creation of a microcontroller system for management of transducers on watering for precision agriculture. The system sends the watering data to a web server for processing, making it an internet of things project. So there is the junction of two expanding areas, the internet of things and precision agriculture. The agriculture is an important area considering the state of Tocantins that have the economy centered in agro-business. We focus on the development of a field transducers control system. In our tests, it was verified the correct behavior of the system. However, its necessary change some sensor to robustest ones. Thus, it was chosen cheap components together with a way to implement the system.

Key–Words: Internet of Things, Precision Agriculture, Wireless Networks, Monitoring and Control

1 Introduction

Over the years, with population growth, water consumption has increased mainly due to its use in agriculture, which in a poorly planned way can lead to excessive spending, wasting much of this resource due to evaporation and even damaging the soil when leaching occurs.

Modern agriculture is currently facing a new era with increasing production needs and increasingly scarce resources. The increase in population brings with it the need of a greater amount of production, the ability to control resources accurately allows greater control of expenses/profits and helps to curb the depletion of natural resources such as rivers and soil nutrients. Precision agriculture allows monitoring of environmental variables and in order to have a precise response and detailed visualization of plantation behavior to solve specific problems in each case.

As in other areas, agriculture is already implementing new technologies, but there is still a wide range of possibilities to be explored to develop new methods of land cultivation that allow better use of resources, maximizing harvesting results.

In this work we discuss the creation of a microcontroller irrigation system that captures data from the environment in which the plantation is located, processes and determines the ideal moment and quantity of irrigation in the plantation, providing wireless data from sensors to the producers that help to have greater control of the monitored area.

1.1 Internet of Things

Currently, we are experiencing a phase of the popularization of projects involving some kind of electronic automation. This is possible due to the low integration costs of a coupled circuits (or within the same encapsulation, so-called System-on-Chip or SoC only), bringing simplicity to the construction of automatic systems. Furthermore, a research area appeared, called the Internet of Things (IoT), along with the also simplified area of Web development.

The IoT is an emerging market that will generate US$14.4 trillion in value [13]. From an industry perspective, four industries make up more than half of the value. These four industries include manufacturing at 27%; retail trade at 11%; information services at 9%; and finance and insurance, also at 9%. Other industries such as wholesale, healthcare, and education lag behind in terms of value generation, with a range between 1% and 7% [13]. This data shows a big growth and popularization in IoT field.

This popularization at great speed also brings important problems that need to be considered, such as *network overhead* - due to a large number of connected devices; *data security* - it is necessary to evaluate whether the web services used allow the data to be integrated, such as the impossibility of data theft; and *update* - the speed with which these systems evolve is very large, and effective and more automated upgrade forms are necessary [12].

Another problem arising from the popularization is the lack of standardization of IoT-based systems. Evidenced fact in the papers of [12] and [2]. The lack of standardization is so important that it goes from the logical architecture of the system as well as the physical architecture. Thus, different vendor solutions can not interact with each other.

In the case of residential automation, a term that has been increasingly addressed in the media as synonymous with modernity, we have air conditioning systems that spend a lot of energy and then could have established times to be turned on and off, its power could also be adjusted according to the external temperature at the moment to increase the energy efficiency of the appliances [1].

IoT is much more than smart homes or connected applications, the highlight is in the scalability that allows its implementation in larger scenarios, such as a smart city [11]. We can count on connected semaphores, management and traffic directing through connected semaphores, unified camera system, data processing to monitor traffic of pedestrians etc [16, 19].

1.2 Precision Agriculture

Precision Agriculture automates the mechanical equipment already used in the fields, bringing technologies and new knowledge of computer science, electronics, geoprocessing among others to the field [17].

Precision farming attempts to reverse the current situation so that, by facilitating the identification of specific deficiencies in certain soil regions, it allows the correct calculation of quantities and locations for the application of agricultural inputs in specific regions of the plantation that requires extra attention [9].

The use of computers makes planning easier since expenditures can be calculated more accurately, make it possible to follow-up the crop in detail, and maximize results, using the full potential that each portion of soil can offer.

Thus, [8] develops and test a low-cost greenhouse with hydroponic crop production was developed and tested using Ubiquitous Sensor Network monitoring and edge control on the Internet of Things paradigm. While [10] presented a solution for in-pipe water quality monitoring based on Internet of Things technology. The solution is used for testing water samples and the data uploaded over the Internet are analyzed. The system also provides an alert to a remote user, when there is a deviation of water quality parameters from the pre-defined set of standard values.

Precision farming is a philosophy of agricultural management based on accurate information for making similar decisions. It represents a way to manage a productive field meter by meter considering the fact that each piece of the farm has different properties [15, 5].

With the use of both concepts, we can add functions to the existing planting and management system creating a new way of interacting with the plantation. An internet-accessible application can be created that allows the observation of data and sending commands where the farmer is, just using his smartphone, for example .

Thus, the continuous development of different tools based on the aforementioned technologies represents an opening for the improvement of the procedures normally adopted in the field in order to minimize expenses and maximize profits [3].

2 Microcontrolled Automatic Irrigation

The sensor module is responsible for collecting plantation data and making decisions regarding actuator module actuation according to profiles established by the user. It includes sensors for air temperature, soil temperature, air humidity, soil humidity, soil temperature and luminosity.

In this way, all the sensors are read in a range programmed by the user and the information obtained is sent wirelessly to be accessed by the user through an MQTT (Message Queue Telemetry Transport) broker.

By collecting the information from the sensors, it will process the data according to user-defined standards to decide the need for water and will be responsible for sending the MQTT command to the actuator module to turn irrigators on / off based on this, enabling the management and rationalization of water resources.

It is important to note that the user can also interact with an actuator module and activate/deactivate irrigation independently of the routine, a necessary measure in cases where sensor malfunctions can lead to mistaken decisions.

The actuator module has the task of waiting for irrigation commands, activating or deactivating the irrigators as requested by the sensor module or by the user.

2.1 Microcontroller

Among the possible circuits, the ESP8266 stands out because it is a SoC (System on Chip) that occupies a minimal space, has built-in Wireless 802.11 features, has a standard antenna and has a low power consumption that can be further reduced using routines and functions preprogrammed in the circuit itself.

ESP8266 is a SoC with all implemented TCP / IP layers that can load applications [7]. Produced by the Chinese Espressif Systems [6], it was first thought and developed to be embedded on mobile devices, its reduced size is due to this.

ESP8266 was designed for network applications, wearable, and mobile devices [14]. Among its characteristics [7], the most outstanding are:

- Support for the IEEE std. 802.11 b / g / n;
- 32-bis RISC CPU: Tensilica Xtensa L106 running at 80 MHz (or 160 MHz);
- 64 KB of instruction RAM, 96 KB of data RAM;
- Integrated TCP / IP protocol layers;
- Supports a wide variety of antennas;
- WiFi 2.4 GHz, with support for WPA / WPA2;
- Supports STA (station) / AP (Access Point) / STA + AP;
- Supports Smart Link function for both Android and iOS;
- SDIO 2.0, (H) SPI, UART, I2C, I2S, IR remote control, PWM, GPIO;
- Deep sleep mode spend of $< 10\mu$A, with leakage energy of $< 5\mu$A;
- Returns from deep sleep and starts transmitting in <2ms;
- Standby power consumption ¡1.0mW (DTIM3);
- Output power of +20 dBm in 802.11b mode;
- Operating temperature -40C ˜125C;
- Certification FCC, CE, TELEC, WiFi Alliance, and SRRC.

MIPS processing, 20 % are already used in the implementation of the set of Wi-Fi layers and their subroutines, everything else can be used for the development of applications.

This SoC integrates the memory controller, as well as SRAM and ROM memory units. All memory units can be accessed through requests, a memory arbiter decides the access order according to the queue of requests generated.

2.2 MQTT Protocoll

The MQTT protocol is designed for low network bandwidth consumption and extremely low hardware requirements, so it is a widely used solution in IoT projects. Its flexibility makes it possible to support multiple scenarios and therefore it has been chosen to be used in exchanging messages from ESP8266 modules with each other and with the [18] client.

The MQTT protocol is defined with two types of entities in the network: a message broker and clients. The broker is a server that receives all messages from the publisher clients and forwards them to the subscribing clients. The operation is described below:

- The customer can connect to the broker and subscribe to any topic of messages in the broker signaling that he is interested in receiving the messages of this topic;
- The client can post messages on any topic, sending the message and indicating which topic in the broker it belongs to;
- The broker then forwards the message to all clients who signed up for this topic.

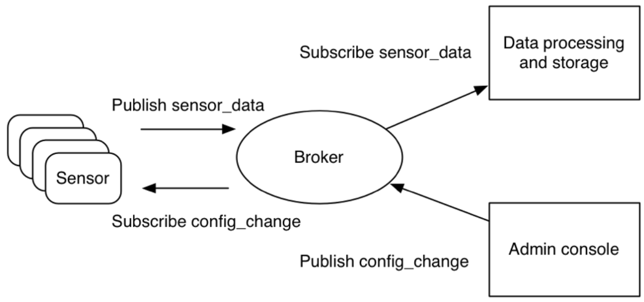

Figure 1: The MQTT publication and subscription model for IoT sensor networks.

Because MQTT messages are organized by topic, there is a great flexibility in specifying which clients can interact with certain messages (Figure 1).

2.3 Modules and Network Nodes

For this project, due to the needs, some alternatives were considered for the distribution of the modules, as can be seen in Figure 2. The separation and nomenclature were adopted in order to facilitate the organization and identification of the modules: the μWEATHER, responsible for reading the data of the environment and decision-making in relation to irrigation and μIRRIGATION that will perform the activation / deactivation of the hydraulic system allowing the precise irrigation of the area.

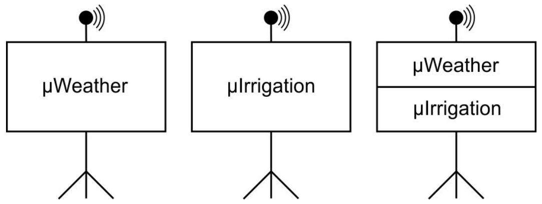

Figure 2: Logical separation between modules and possible functions for the proposed solution.

The adopted system allows the modularity of the nodes according to a section of the plantation (defined by the reach of the sensors). Further to the specific need of the area, we can associate to each node the module pertinent to require control of that region.

For the message exchanges in the network, we have the following scenario: each sensor node is responsible for detecting if a region needs irrigation and it is responsible for sending commands directly to the assigned actuator, publishing the sensor data via MQTT so to user access. Then, it has a direct hierarchy relation in which the server is at the top of the chain of commands that are propagated to the sensor modules and subsequently to the actuator modules associated with them.

The relation of sensors and their parameters to be worked are described in Table 1.

Actuators are necessary at the moment when some intervention must be done in the soil to control the soil humidity. The actuators used to execute the project are:

- Solenoid Valve – used to open / close the irrigation;
- Relays – used to control the solenoid valve.

3 Methodology

Each network node consists of one or more modules to make the system implementation more flexible. For different cultures that have distinct sizes and needs for irrigation and management. Therefore, a node can contain more than one mounted module sharing the same physical space.

The reading interval is specified by the user, the μWEATHER module regularly publishes in the topics described in the Table 2. Further, this module can be configured from the list of commands contained in the Table 3. These data are exchanged between the two module (μWEATHER and μIRRIGATION) to allow automatic control of the system. That is, according to the values of the sensors, it is possible to start the irrigation. This allows the autonomy of the system by eliminating the need for human intervention.

The values expressed in the Tables 2 and 3 between 0 and 1023 represent the reading of the analog sensors. The A / D converter in ESP8266 has a resolution of 10 bits, hence $2^{10} = 1024$ possible read values. The name of the topic is in Portuguese, but it does not difficult the understanding. The label $< namemodule >$ represents the name of an individual module in the network.

The physical implementation of the μWEATHER is illustrated in Figure 3. The implementation was done in such way that make the best possible use of the pins available on the ESP8266, freeing the D0 pin, which is used to drive the relay in the μ Irrigation module (Figure 4). This measure was taken to let the two connection diagrams to be complementary, allowing it to be merged with the μIRRIGATION making the customization easier and more flexible.

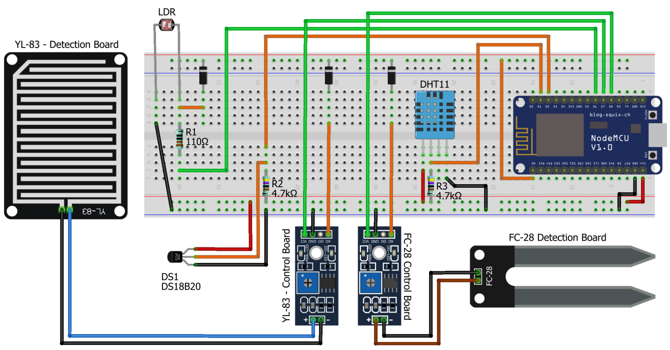

Figure 3: Diagram of connections between the components of the μWEATHER module.

Throughout the implementation, it was found that the ESP8266 has only one pin capable of making analog readings, making it necessary to use a multiplexer so that it is possible to connect several analog sensors to this analog pin, which on the board is marked A0. In the absence of CI multiplexer, a multiplexer circuit was implemented with the use of diodes. This solution is illustrated by the green wires of Figure 3.

The multiplexer circuit consists of alternating the activation of the sensors through the green connections and collecting their data through the orange connections, for this we connect the GND pins of all the sensors to the pin GND on ESP8266, we connect the VCC pin of each sensor to a different digital pin on the ESP8266 (in this case the pins are D6, D7, and D8),

Sensor Type	Detection Time	Electric Current	Voltage	Accuracy	Reading range
Air temperature (DHT11)	$< 5s$	$200\mu A$ to $500mA$	3 to 5 V	$\pm 2°C$	$0°$ to $50°C$
Air humidity (DHT11)	$< 5s$	$200\mu A$ to $500mA$	3 to 5 V	$\pm 5.0\%$ Relative Humidity (RH)	20% to 90% Relative Humidity (RH)
Soil temperature (DS18B20)	$750ms$	$< 1.5mA$	3 to 5.5 V	$\pm 0.5°C$	$-55°C$ to $+125°C$
Soil humidity (FC-28)	-	$< 0.4mA$	3.3 to 5.5 V	-	-
Luminosity (LDR)	$120ms$	$< 75mA$	3 to 5 V	-	10 to 1000 Lux
Rain (Leaf humidity - YL-83)	-	$< 100mA$	3.3 to 5 V	-	-

Table 1: Types of sensors used and operating parameters of each.

Topic	Message
iam/modules/microclima/$<modulename>$/reading/data	Timestamp
iam/modules/microclima/$<modulename>$/reading/luz	value[0-1023]
iam/modules/microclima/$<modulename>$/reading/umidadear	value[%]
iam/modules/microclima/$<modulename>$/reading/temperaturaarc	value[%]
iam/modules/microclima/$<modulename>$/reading/temperaturaterrac	value[0-1023]
iam/modules/microclima/$<modulename>$/reading/umidadesolo	value[0-1023]
iam/modules/microclima/$<modulename>$/reading/umidadefolha	value[0-1023]

Table 2: List of sensor data published by the module μWEATHER to the application server.

Topic	Message
iam/modules/microclima/$<modulename>$/ perfil/intervalReading	duration of irrigation in seconds (default: 10)
iam/modules/microclima/$<modulename>$/ perfil/intervalIrrigation	duration of irrigation in seconds (default: 10)
iam/modules/microclima/$<modulename>$/ perfil/initialTrigger	value[0-1023] (default: 800)
iam/modules/microclima/$<modulename>$/ perfil/finalTrigger	value[0-1023] (default: 1000)

Table 3: List of commands sent by module μWEATHER to module μIRRIGATION associated.

then connecting the data outputs of all the sensors to the pin A0, we can make the activation of the sensors alternately by sending HIGH signal on the digital pins according to the analog sensors connected to them.

According to the diagram in Figure 3, pins D6, D7 and D8 activate the light sensors, leaf humidity and soil humidity sensors, respectively. Finally, to avoid leakage of current in undesired directions in the circuit, diodes were used at the output of each of the sensors.

The ESP8266 can be programmed using the Lua language, which is also its native language, but despite the ease of use of the Lua language, there is the possibility to configure the Arduino SDK to compile for the ESP8266 programming in the standard Arduino language, the use of numerous existing libraries implemented for Arduino's C / C ++ language, which was used.

The Listing 1 presents the main function of the `loop()` code. Between lines 4 and 10 checks are made if irrigation is required, in the case of need to start or stop irrigation, an MQTT command is sent to the designated μIRRIGATION module.

```
1  void loop() {
2      setMQTTCallback(receivedMQTT);
3      publishSensors();
4      if(soilHumidity < initialTrigger){
5          publishMQTT("iam/modules/
               microirrigacao/$<module name
               >$/intervalIrrigation",
               intervalIrrigation);
6          publishMQTT("iam/modules/
               microirrigacao/$<module name
               >$/irrigar", on);
7      }
8      if(soilHumidity > finalTrigger){
9          publishMQTT("iam/modules/
               microirrigacao/$<module name
               >$/irrigar", off);
10     }
11     wait(intervalReading);
12 }
```

Listing 1: `loop()` function from μWEATHER.

The `loop()` function also checks if a new MQTT command has arrived and sends it to the `receivedMQTT()` function to decide what to do with the received message. The check is made based on the values of the `initialTrigger` variable and the `finalTrigger` to determine the time to start or stop watering and wait according to the value `intervalReading` to start all over again.

The Listing 2 shows function `publishSensors()` responsible for reading the sensors, stores them in the global variables and publishes them together with the date and time so that the client knows the time the reading was performed.

```
void publishSensors(){
    light = read(lightSensor);
    airHumidity = read(
        airHumiditySensor);
    airTemperature = read(
        airTemperatureSensor);
    soilTemperature = read(
        soilTemperatureSensor);
    soilHumidity = read(
        soilHumiditySensor);
    leafHumidity = read(
        leafHumiditySensor);

    publishMQTT("iam/modules/microclima
        /$<module name>$/reading/data",
        getdate());
    publishMQTT("iam/modules/microclima
        /$<module name>$/reading/luz",
        light);
    publishMQTT("iam/modules/microclima
        /$<module name>$/reading/
        umidadear", airHumidity);
    publishMQTT("iam/modules/microclima
        /$<module name>$/reading/
        temperaturaarc", airTemperature)
        ;
    publishMQTT("iam/modules/microclima
        /$<module name>$/reading/
        temperaturaterrac",
        soilTemperature);
    publishMQTT("iam/modules/microclima
        /$<module name>$/reading/
        umidadesolo", soilHumidity);
    publishMQTT("iam/modules/microclima
        /$<module name>$/reading/
        umidadefolha", leafHumidity);
}
```

Listing 2: `publishSensors()` function from μWEATHER.

The Listing 3 shows the function `receivedMQTT()` which is responsible for receiving all the configuration parameters of the module and storing them in global variables so that the whole code is executed according to the values stored in them.

```
void receivedMQTT(topic, message) {
    if(topic = "iam/modules/microclima/
        $<module name>$/perfil/
        intervalReading"){
        intervalReading = message;
    }
    if(topic = "iam/modules/microclima/
        $<module name>$/perfil/
        intervalIrrigation"){
        intervalIrrigation = message;
    }
    if(topic = "iam/modules/microclima/
        $<module name>$/perfil/
        initialTrigger"){
        initialTrigger = message;
    }
    if(topic = "iam/modules/microclima/
        $<module name>$/perfil/
        finalTrigger"){
        finalTrigger = message;
    }
}
```

Listing 3: `receivedMQTT()` function from μWEATHER.

This μIRRIGATION is designed to be the field actuator. Within the plantation, it will be linked to a μWEATHER module of a region and, within the programmed routine, will receive from that module the instruction of how to operate. Thus, when a new read is processed, the μWEATHER assigned to this module will send a power on/off command according to the region's watering need and the module μIRRIGATION will activate/deactivate the hydraulic system.

Compounded by a SoC ESP8266 connected to a relay, it connects to the MQTT broker and awaits instructions, which are detailed in Table 4.

Topic	Message
iam/modules/microirrigacao/ $<modulename>$/irrigar	on / off
iam/modules/microirrigacao/ $<modulename>$/intervalIrrigation	duration of irrigation in seconds (default: 10)

Table 4: List of commands accepted by module μIRRIGATION.

The layout of your physical assembly is illustrated in Figure 4. The pin D0, by default, is the activation pin of the relay for solenoid valve actuation. The solenoid valve has external 110/220V AC power.

In the Listings 4, 5 and 6 the code parts are complementary and in implementation they appear together in the same code. However, the codes were separated for better understanding. Thus, the code parts from the Listings 4, 5 and 6 will only be active if they are running in a μIRRIGATION module as well as the codes presented will only be active when they are running in a μWEATHER module.

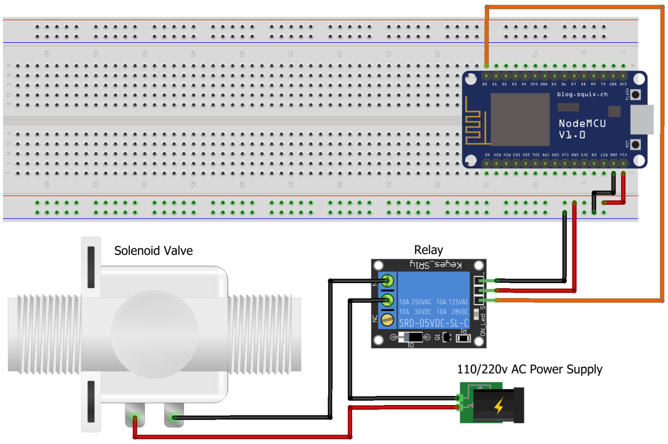

Figure 4: Diagram of connections μIRRIGATION module.

The Listings 4 presents the declaration of the global variable `hourOffIrrigation` that has the capacity to store the time and instant data to identify the time when irrigation should be shut down. In the `loop()` is checked if the time constant described in `hourOffIrrigation` reached, if so, the `stopIrrigation()` is triggered immediately. A trigger is also set up which calls the `receivedMQTT()` function as soon as some message is received from the MQTT server.

```
time hourOffIrrigation = now()+10;
void loop(){
    setMQTTCallback(receivedMQTT);
    if(hourOffIrrigation == now()){
        stopIrrigation();
    }
}
```

Listing 4: Declaration of global variables and `loop()` function from μIRRIGATION.

Listing 5 displays the function `receivedMQTT()`. It manages the command messages for irrigation so that upon receipt of an interval value followed by the command "on", it triggers the `startIrrigation()` function and passes the irrigation duration interval to the parameter. If the message is "off", it triggers the function `stopIrrigation()`.

```
void receivedMQTT(topic, message) {
    int interval = 10;
    if(topic = "iam/modules/
        microirrigacao/$<module name>$/
        intervalIrrigation"){
        interval = message
        }
    if(topic="iam/modules/
        microirrigacao/$<module name>$/
        irrigar"){
        if(message = on){
            startIrrigation(interval);
        }
        if(message = off){
            stopIrrigation();
        }
    }
}
```

Listing 5: `receivedMQTT()` function from μIRRIGATION.

The function `startIrrigation()` (Listing 6) receives a range as a parameter, then the relay pin is activated and the variable `hourOffIrrigation` receives the current hours, minutes and seconds plus the irrigation duration interval that will later be checked by the `loop()` function in order to know when the `stopIrrigation()` function is called for the irrigation cycle to complete.

```
void startIrrigation(interval) {
    on(pinRelay);
    hourOffIrrigation = now()+interval;
}
```

Listing 6: `iniciaIrrigação()` function from μIRRIGATION.

Listing 7 shows the function `stopIrrigation()`. It terminates the irrigation cycle by deactivating the relay pin and finally sending a return message to the server indicating that the irrigation is off.

```
void stopIrrigation(){
    off(pinRelay);
    readMQTT("iam/modules/
        microirrigacao/$<module name>$/
        irrigar", off);
}
```

Listing 7: `stopIrrigation()` function from μIRRIGATION.

4 Results and discussions

For the simulations and tests, a vase of 44x19x14cm proportions was used with a portion of " black earth ", widely used in gardens due to its high fertility level. Then the soil was read at its lowest tolerable humidity level by the system as well as the reading of a higher level of humidity tolerated by the system so that these values are the standard for the irrigation routines of the system. It is worth remembering that the user can change these values at any time.The ESP8266 has an A/D converter of 10 bits, so the sensor reading varies from 0 to 1023, and the higher the value, the higher the soil humidity level, then the minimum soil humidity level has been set at 800 and the maximum at 1000.

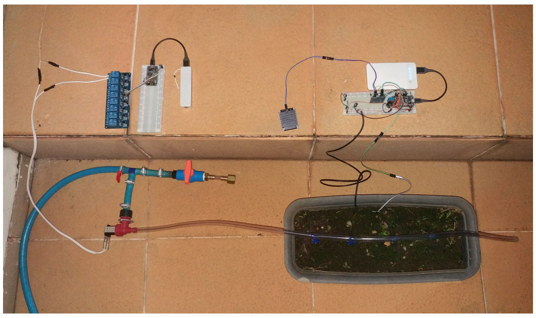

Figure 5: Modules installed in the vase and ready for a test cycle.

With the two modules, μWEATHER and μIRRIGATION mounted in the vase (as described in Figure 5), a test was started at 5:00 in November 1fst 2017, ending at 23:30 the same day to verify accuracy, noise, system reliability and to detect possible malfunctions.

The graph of Figure 6 illustrates the readings of the light sensor at the specified time. The curve indicating that sunrise occurred just after 5:30, having its highest point of incidence of sunbeams at the sensor exactly at 13:00 and returned again to its minimum level of light incidence at 18:30.

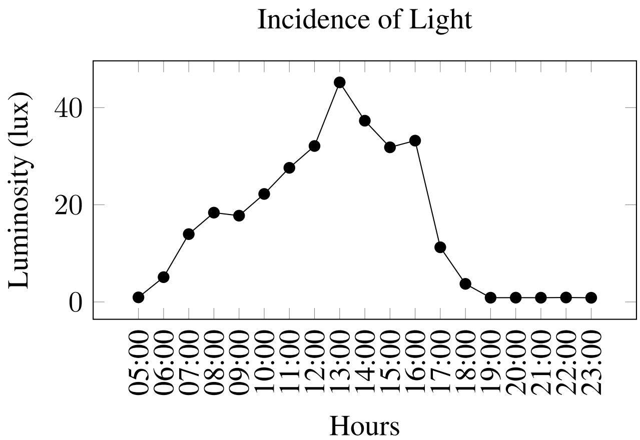

Figure 6: Data collected by the LDR analog light sensor over the course of one day.

Two interventions were required through the system to maintain the water levels in the tested soil sample, the first irrigation occurred at 5:00, as soon as the system was started and the second irrigation at around 15:00, humidity level has reached the specified minimum value.

It is important to analyze the graphs of the Figures 7 and 8 in parallel, as we thus notice that the fall of soil humidity (Figure 7) occurs faster at the moment the earth temperature (Figure 8) rises.

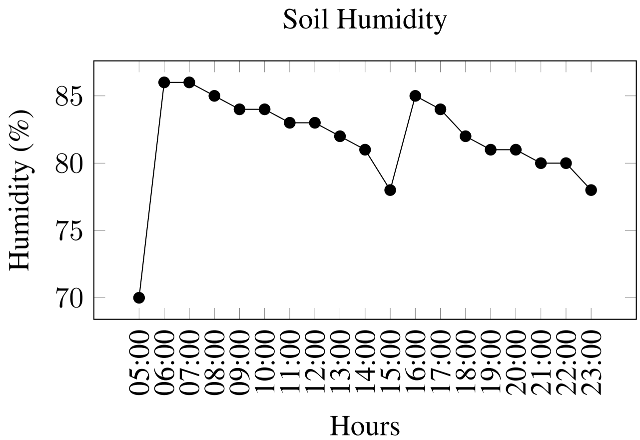

Figure 7: Soil humidity readings collected by the analog sensor FC-28 over the course of one day.

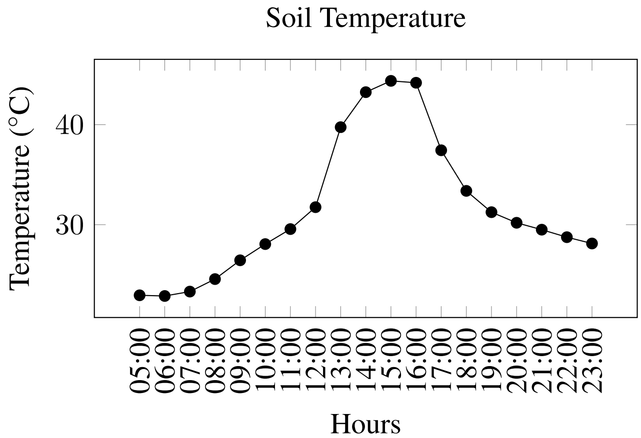

Figure 8: Soil temperature readings collected by the DS18B20 digital sensor over the course of one day.

In the same way, it is possible to note that the humidity of the leaf in Figure 9 changes in parallel to the solar incidence of the Figure 6, falling the humidity as soon as the first rays of sun begin to affect. As a function of the air temperature in Figure 10, the humidity stabilizes at the midday and the maximum temperature is reached and the humidity of the leaf drops further.

The plot of the air humidity reading, shown in Figure 11, relates sharply to the incidence of light shown in Figure 6, beginning its decline soon after the rise in incidence levels luminous and rising again very slowly after the light levels begin to fall.

The noise levels, in general, were within an acceptable limit. However, the DHT11 air temperature sensor made incorrect measurements due to the fact

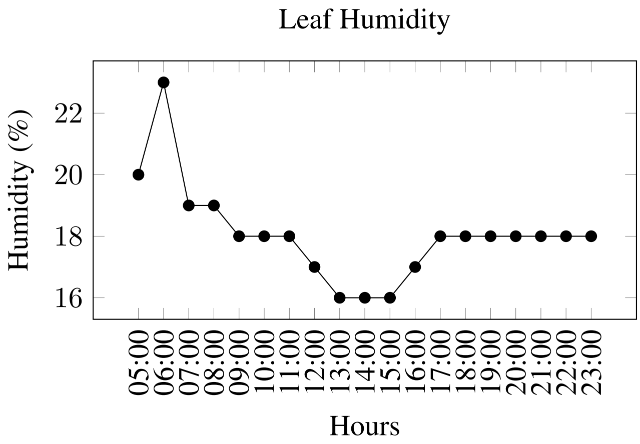

Figure 9: Rain readings or leaf humidity collected by the YL-83 analog sensor over the course of one day.

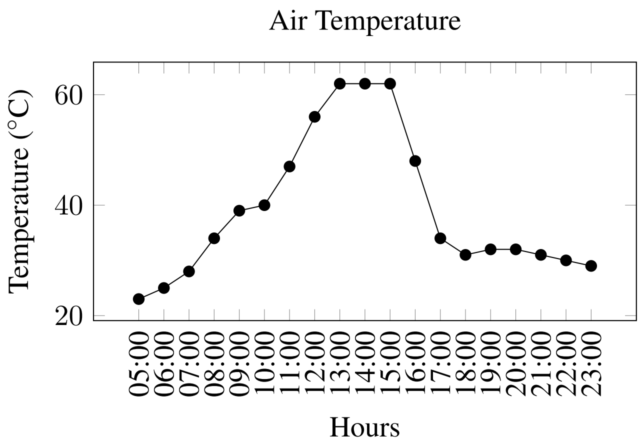

Figure 10: Air temperature readings collected by the DHT11 digital sensor over the course of one day.

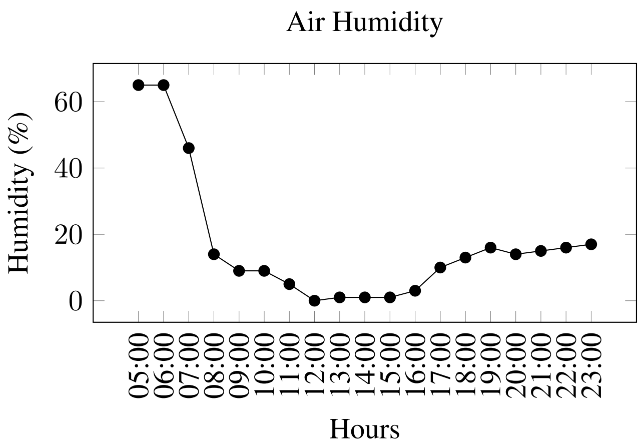

Figure 11: Air humidity readings collected by the DHT11 digital sensor over the course of one day.

that it remained at 62°C during the entire time of the highest solar incidence. This is because the sensor has exceeded the maximum reading temperature, which can be easily reached by leaving the equipment exposed to periods of direct sunlight. The operating range of DHT11 is from 0 to 50°C and the range operation in the environment is recommended ranging from 10 to 40°C [4].

Prolonged energization of the soil humidity sensor without data readings results in corrosion on the specimen above normal. Therefore, the energization of the sensor was done in short periods of time, in which the data is read, thus avoiding the process of electrolysis with the water present in the soil that results in the corrosion of one of the poles of the probe. This helps to extend the life of the probe.

5 Conclusion

The progress in the construction of embedded systems and the creation of more efficient web services allowed and leveraged the growth of the IoT. One of the main applications of IoT technology is precision farming. This area aims at the development of novel equipment for the efficient management of agribusiness, regardless of which segment it is in. It is in this focus that the present paper concentrates.

Therefore, a solution was developed for precision agriculture to the water control of a given crop. The water control used in irrigation has been chosen to be one of the most serious and fundamental problems of modern agriculture. The proposed system was divided into two modules, one for sensing and another for triggering irrigation.

In the sensing module, several types of sensors were used, which would not all be effectively needed for automatic irrigation control, however, the information generated provides more information about the wheather conditions for decision making by the producer.

At the same time, all the information is sent to the Internet, which facilitates the retrieval of the information, later, for reuse, sharing, and treatment. After sending to the broker on the internet, the data should be sent to an application server for the effective use. From the server, it is possible to develop mobile and/or hybrid applications for data manipulation.

The tests were carried out in a controlled garden environment, however, its operation was totally adequate, in the tested configurations, which allows concluding that the proposed system model can be used without configuration difficulties in different cultures and at different scales.

However, we emphasize that due to errors in the temperature and humidity sensor (DHT11), it would be prudent in future work to use more robust sensors. The change of soil humidity sensor would also be prudent, since it is a resistive sensor the oxidation of the material is unavoidable, at most delayed. Therefore, with these additions and changes, the solution costs would be higher, which initially was quite low. A major development concern was always keep the project costs low.

References:

[1] A. R. Al-Ali, I. A. Zualkernan, M. Rashid, R. Gupta, and M. Alikarar. A smart home energy management system using iot and big data analytics approach. *IEEE Transactions on Consumer Electronics*, 63(4):426–434, November 2017.

[2] S. A. Al-Qaseemi, H. A. Almulhim, M. F. Almulhim, and S. R. Chaudhry. Iot architecture challenges and issues: Lack of standardization. In *2016 Future Technologies Conference (FTC)*, pages 731–738, Dec 2016.

[3] AntonisTzounis, Nikolaos Katsoulasa, Thomas Bartzanas, and Constantinos Kittas. Internet of things in agriculture, recent advances and future challenges. *Biosystems Engineering*, 164:31 – 48, 2017.

[4] Aosong Electronics Co. Datasheet do sensor dht11, 2018. `http://img.filipeflop.com/files/download/Datasheet_DHT11.pdf`.

[5] Bill Batchelor. *Precision agriculture: introduction to precision agriculture*. University Extension, Iowa State University, 1997.

[6] Espressif Systems, 2016. `https://espressif.com`.

[7] Espressif Systems. Esp8266 datasheet, 2017. `http://espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf`.

[8] Francisco Javier Ferrández-Pastor, Juan Manuel García-Chamizo, Mario Nieto-Hidalgo, Jerónimo Mora-Pascual, and José Mora-Martínez. Developing ubiquitous sensor network platform using internet of things: Application in precision agriculture. *Sensors*, 16(7), 2016.

[9] CW FRAISSE and L Faoro. Agricultura de precisão: a tecnologia de gis/gps chega às fazendas. *Revista Fator GIS, Curitiba*, (21):28–33, 1998.

[10] S. Geetha and S. Gouthami. Internet of things enabled real time water quality monitoring system. *Smart Water*, 2(1):1, Jul 2017.

[11] N. L. L. Guang, T. Logenthiran, and K. Abidi. Application of internet of things (iot) for home energy management. In *2017 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, pages 1–6, Nov 2017.

[12] P. Kansakar and A. Munir. Selecting microarchitecture configuration of processors for internet of things (iot). *IEEE Transactions on Emerging Topics in Computing*, pages 1–1, 2018.

[13] In Lee and Kyoochun Lee. The internet of things (iot): Applications, investments, and challenges for enterprises. *Business Horizons*, 58(4):431 – 440, 2015.

[14] Never Underestimate Real Development spaces (NURDs). Esp8266 nurdspace, 2016. `https://nurdspace.nl/ESP8266`.

[15] D Roza. Novidade no campo: Geotecnologias renovam a agricultura. *Revista InfoGEO, n*, 2000.

[16] Abida Sharif, Jian Ping Li, and Muhammad Irfan Sharif. Internet of things network cognition and traffic management system. *Cluster Computing*, Jan 2018.

[17] Mauro Tschiedel and Mauro Fernando Ferreira. Introdução à agricultura de precisão: Conceitos e vantagens. *Ciência Rural*, 32(1):159–163, February 2002.

[18] Michael Yuan. Conhecendo o mqtt, 2018. `https://www.ibm.com/developerworks/br/library/iot-mqtt-why-good-for-iot/index.html`.

[19] L. Zhou and H. C. Chao. Multimedia traffic security architecture for the internet of things. *IEEE Network*, 25(3):35–40, May 2011.