Deliverable B.2:

PILOT PLANT DESIGN OF THE INTEGRATED SYSTEM FOR WATER REUSE AND RECYCLING

Action	B.2	Pilot plant design of the integrated system for water reuse and recycling
Dissemination level[1]	PP	
Nature[2]	R	
Due delivery date	31/03/2016	
Actual delivery date	27/05/2016	
Number of pages	23	

Lead beneficiary	University of Murcia
Contributing beneficiaries	RITEC, CEBAS-CSIC

Document Version	**Date**	**Author(s)**	**Comments**
1	18/03/2016	Cristina Albaladejo Pérez, Miguel Ángel Zamora Izquierdo, Antonio Skarmeta Gómez	
2	05/2016	Cristina Albaladejo Pérez, Miguel Ángel Zamora Izquierdo, Antonio Skarmeta Gómez	

Acknowledgement:

This deliverable was produced under the co-finance of the European financial instrument for the Environment (LIFE) programme during the implementation of the project "LIFE DRAINUSE" (LIFE14 ENV/ES/000538).

[1] Dissemination level: **PU** = Public, **PP** = Restricted to other programme participants, **RE** = Restricted to a group specified by the consortium, **CO** = Confidential, only for members of the consortium.
[2] Nature of the deliverable: **R** = Report, **P** = Prototype, **D** = Demonstrator, **O** = Other.

Deliverable abstract

This document describes the control unit design necessary to build the integrated system for water reusing and recycling. Control unit design of the pilot plant proposed in the project involves the study of several parts that are detailed in this document. We have analyzed different hardware and communication architectures to select the most appropriated in this case. Two software platforms are presented, detailing in each layer its operation. And regarding to functional specification, this document describes the general functionality and the inputs and outputs necessaries (sensor database) for the operation of the system.

In section 1, an introduction to the functional description through the different modules is described. Next, the document focuses in the hardware architecture (section 2) and the communication architecture (section 3). Section 4 describes the software platform, user interface and the communication protocol. Section 5 details the control unit functionality and sensor database. Finally, a summary is included in section 6.

List of acronyms and abbreviations:

PLC – Programmable Logic Controller

M2M – Machine to Machine

SHA-2 – Secure Hash Algorithm

AES – Advanced Encryption Standard

CU – Control Unit

EC – Electrical conductivity

AI – Analog Input

DI – Digital Input

DO – Digital Output

List of Contents

List of Figures

1. Overview of greenhouse modules

In general, some parameters are important to measure in a greenhouse. These can be group together in some control modules as Figure 1 shows: irrigation module, climate module, nutrition module and crop growth module. Each module has an independent function and, moreover, they interact to perform a smart control of the complete system. The nutrition module is related directly with the crop growth module. This, in turn, depends on climate and irrigation modules, and these last are also connected.

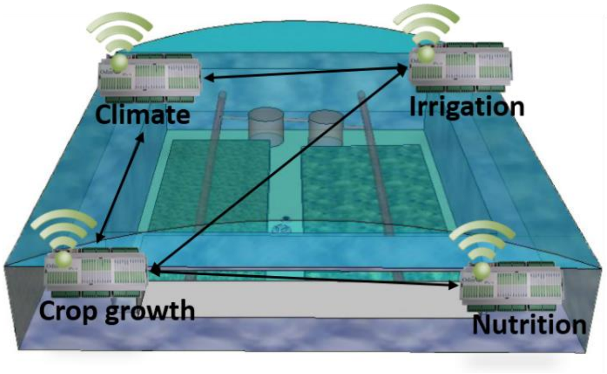


Figure 1: General control modules in a greenhouse

The **irrigation module** manages the valves and pump to activate or deactivate the irrigation in the crop. This module acts according to parameters fixed by the user and parameters measured by climate and crop growth modules.

The **climate module** measures the climate conditions inside and outside of greenhouse with the aim of controlling the system by means of, for example, opening or closing of windows and activating or deactivating of refrigeration system.

The **crop growth module** is composed of sensors to monitor the state of the plants and this way it can to make a tracking of growth. Depending on this growth, the rest of modules will adapt each operation to optimize the performance of the system.

Finally, the **nutrition module** is responsible of preparing the composition of water and nutrients dependent on real state of the plant.

This project is focused in water reuse, this way, although the hardware and communication architectures are designed for a whole control system in the greenhouse, that involves all the modules described above and other additional ones, we focused the detailed Input/output descriptions and control loops to the modules that use the water, that is, nutrition, purification and disinfection modules.

2. Hardware architecture

For achieving this aim, Figure 2 shows a general architecture that includes different communication protocols, several controllers with different types of interfaces to connect sensors and a software platform to final users.

The ideal architecture have to be flexible and scalable. The flexibility in the inputs and outputs ports have to be the maximal. The controllers that we will use in this project are based on one of the latest generation of 32bit microcontrollers and have 16 ports that can be configured independently as digital inputs, analog inputs (voltage controlled 0-10V or current controlled 4-20 mA), and also as analog outputs (0-10V) or digital outputs (for relays). And moreover, each master controller can manage 16 slaves. In this way, a controller can have up to 256 additional inputs/outputs ports.

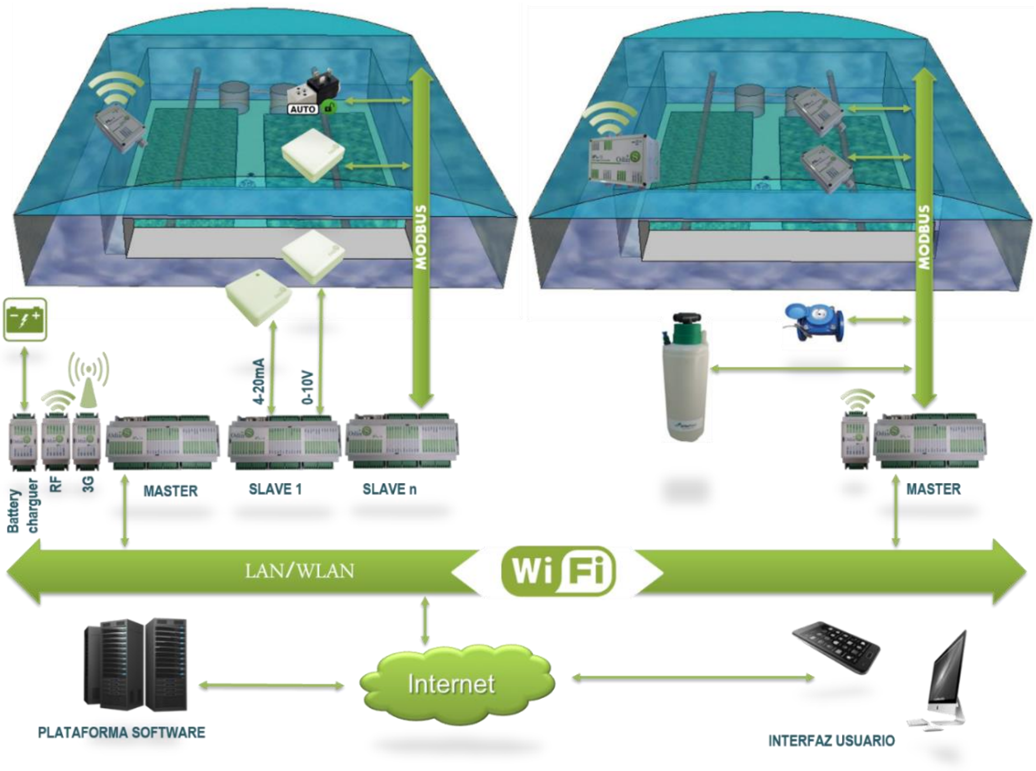


Figure 1: Architecture of a Smart Greenhouse

The controllers have communication ports based on the most popular technologies. As you can see in Figure 1, the controllers provide communication through multiple protocols that turn it into a very versatile tool, able to communicate with a wide range of sensors, devices and electronic systems.

Taking into account the flexibility of the hardware architecture that we will use in Drainuse, and depending on location of each module (inside the greenhouse or inside the irrigation hut), we can suggest different options related with the internal hardware architecture. The following topologies are possible:

a) In the first option, as you can see in Figure 2, there are a master controller with some slaves distributed inside of the irrigation hut and greenhouse. The communication between all is through CAN bus and the master controller will be responsible for connecting (through Ethernet, WIFI or cellular) with an Intranet or Internet (cloud software).

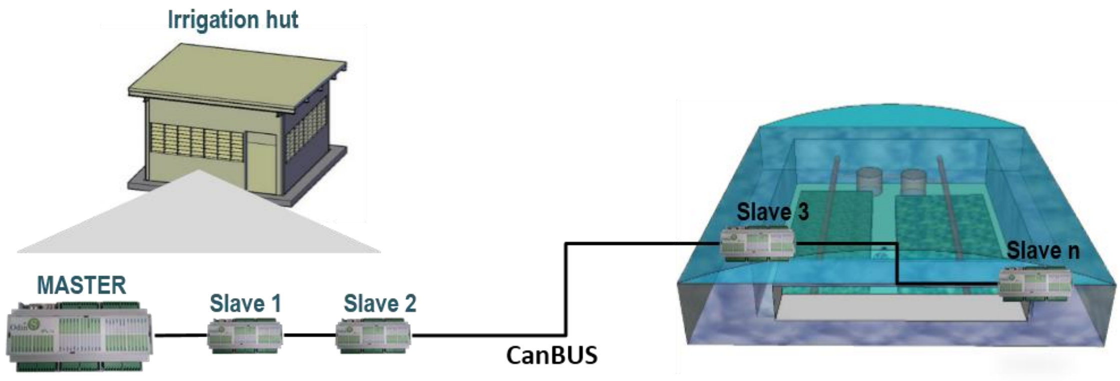


Figure 2: Internal Communication architecture with CanBUS

b) This option include two master controllers, one located inside of the irrigation hut and another inside of the greenhouse (see Figure 3). Both are able to communicate through a switch to connect with Internet. The communication between master controller and slaves is the same that in previous configuration, through CAN bus.

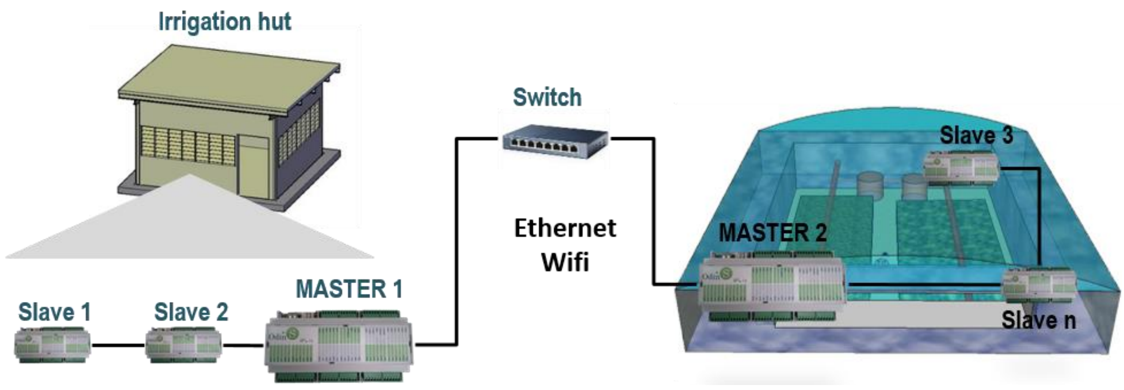


Figure 3: Internal Communication architecture using Switch

c) As Figure 4 shows, the last option is composed of a master controller with slaves connected through CAN bus inside of the irrigation hut, and several low power devices (connected to sensors/actuators) distributed in greenhouse. These devices can connect with the master controller using RF communication.

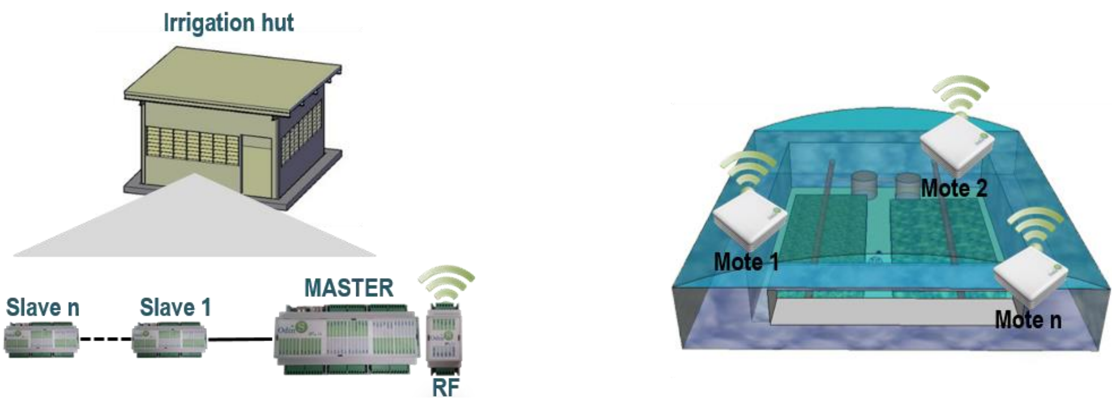


Figure 4: Internal communication architecture using RF

On the other hand, the communication between master controllers and Internet will depend on the possibilities in each situation (see Figure 5). Nevertheless, in both cases the system have to include a cellular alarm system to warn immediately the user when any fault occurs.

a) An option is to connect master controller with Internet through cable (**ADSL or fiber optic)** and moreover connect to him a **3G/GPRS module** operating in GSM mode. In this case, data are transmitted via ADSL/fiber and the alarm system would use:
   - Emails through cable ADSL/fiber to send advices to users
   - And SMS and calls through cellular GSM.

   In addition, if the ADSL/fiber connection fails, the master controller is able to commute from GSM mode to GPRS mode and so to send the advices to the users. With this topology there is a redundant alarm system.

b) The second options is to use only **3G/GPRS/GSM communication**. With this configuration the master controller sends data through 3G/GPRS and commute to GSM communication to send SMS or make calls.

Finally, we can add that the control system proposed has the possibility of using and alternative wireless communication called Sigfox. This very new standard is starting to be used for sensor monitoring in remote places were 3G/GPRS coverage is low, and follow the Internet of Things philosophy.

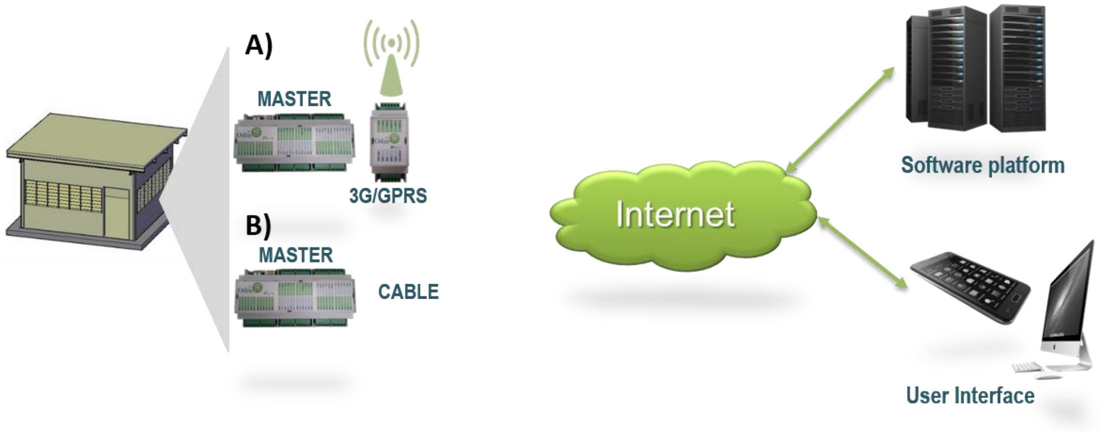


Figure 5: External communication architecture

The internal communication architecture in this project, due to the charge of inputs/outputs, a hybrid architecture between Figure 2 and Figure 3 is the most suitable for this system. Section 4 details the quantity of master controllers and slaves necessaries to connect all the sensors. About the external communication architecture, ADSL communication will be used since the greenhouse pilot have this option.

3. Communication architecture

Related to communication architecture, we presents two different alternatives of architecture. The more appropriated choice depends on requirements in each application. One alternative is focused on using full controller functionality, with high computational performance (section 3.1) similar as a PLC (Programmable Logic Controller) but with IoT interfaces and adapted to the greenhouse use case, and another possibility of architecture is more oriented to dataloggers, where intelligence is in the cloud software instead of the controller (section 3.2). Both alternatives are based on IoT paradigm.

3.1. Multilayer communication architecture

The architecture is divided in different layers, with some of them having persistence and/or logging support. The main persistence unit is a database, which stores both real-time and historic information, and acts as a communication channel between different layers.

According to

Figure 6, there are the following horizontal layers:

- **Front-end**. This is the interface used by external entities, such as users or other enterprises.
- **Back-end**. This includes drivers to communicate with the controllers, database and moreover support to REST, MQTT and CoAP protocols (IoT protocols). Back-end communicates with Front-end using HTTP protocol.
- **Hardware layer**. The controllers communicate with back-end through Web Services.

Interface	REST
Description	This interface offers RESTless support to all the information offered as public by the administrators. At the same time, it controls the access to restricted data so that not all information has to be public.

Interface	MQTT
Description	For users who want to use a publish/subscribe system, MQTT is available. In these cases, when users just want to be notified when data changes, being subscribed to a topic is an easier and more lightweight option than polling.

Interface	CoAP
Description	The main objective of the CoAP application protocol is to provide a generic Web protocol for the special requirements of constrained wireless nodes. CoAP is similar to HTTP but its goal is not to simply compress HTTP but to realize a subset of REST operations optimized for M2M interactions.

At the same time, there are also a vertical common layer named **Management**. This includes an Editor to register and set up the different master and slave controllers. The configuration of inputs and outputs of the controllers is done directly from the Graphic Editor of the Software Platform, through dialog boxes and without programming the controllers. Furthermore, with this editor and the different software intelligent modules we are able to carry out basic control actions (based on inputs/outputs states, timers, events, etc.) and also the control of a complex installation. Everything without writing any line of code.

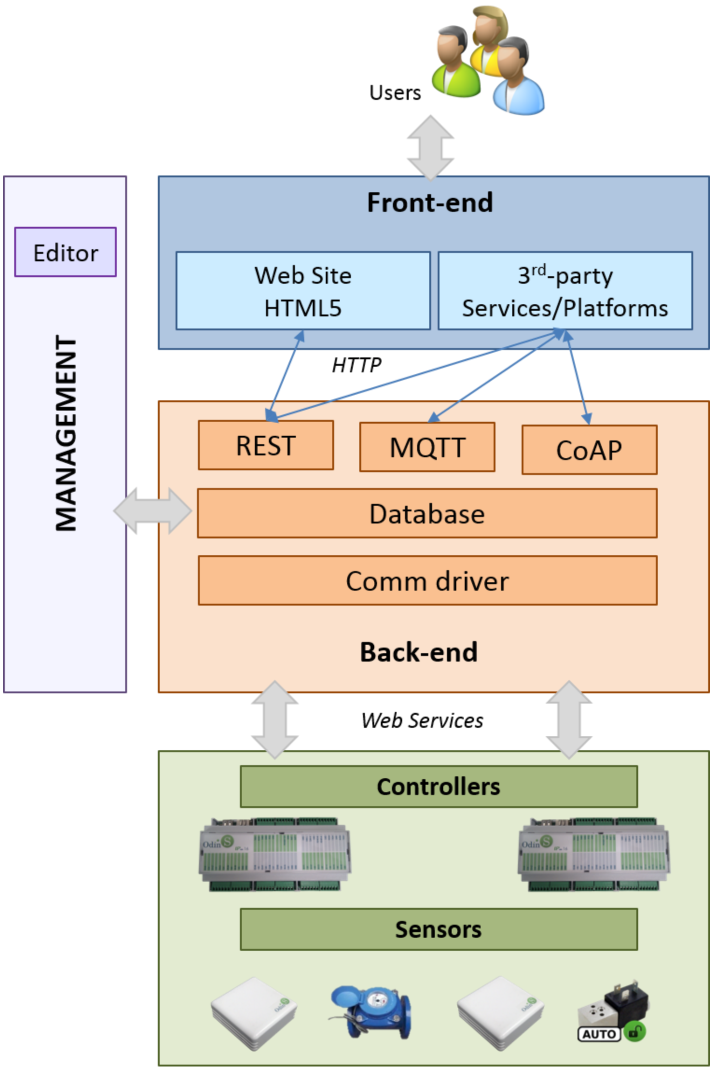


Figure 6: Multilayer communication architecture

3.2. M2M communication architecture

If the device only acts as a datalogger, an M2M (Machine to Machine) architecture is acceptable (see Figure 7). M2M allows any sensor or gateway to communicate themselves with an IoT system and automatically responding to changes in its environment.
In this case, the architecture is divided only in two different layers:

- **Application layer**. The datalogger can sending information directly to a proprietary web service or to third party.
- **Hardware layer**. The device is able to communicate using the standard protocols MQTT, CoAP and REST.

As we mentioned above, in this case the intelligence of the system is out of the hardware (controllers), and the controllers are used as gateways to monitoring the different sensors (as a datalogger) and acts over the actuators (engines, valves, etc).

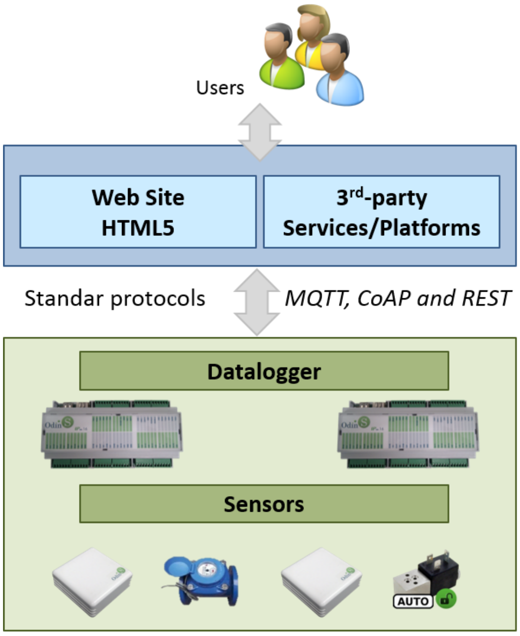


Figure 7: M2M communication architecture

The last option, commented in previous paragraph, is oriented to big quantity of sensors and controllers (big deployment as a Smart City use case), were the real control unit is in the cloud and the controllers are gateways. Nevertheless, this option is less robust for control loops and critical systems where the control has to be guaranteed even in case of loss of communications. This is the case of this project, where the greenhouse has to be a reliable system, always in operation even without internet communication, the intelligence has to be in the hardware. For this reason, the first option in communication architecture will be the implemented in this use case.

4. Software platform and user interface

4.1. Software platform

The software platform designed for this project is based on Web 2.0 (Web Site HTML5 in Figure *7*) since the users, not only are able to visualize data in graphs or maps, but also they are able to interact with the system through this software (tele-operate or command, change configuration parameters, etc).

Once the system is configured by user using the editor, the web graphical interface of the platform manages all the devices installed, shows the alarms and possible incidents of the system, and give us the possibility of visualization all kind of historic data.

The remote management and monitoring can be done through any computer or mobile device (laptop, tablet, smartphone, etc.) with a commercial web browser thanks to the HTML5 web technology.

As Figure *7* shows, the communication protocol used between Back-end and the Web Site HTML5 of front-end is based on REST standard protocol but the protocol implemented between back-end and the hardware layer is a proprietary protocol.

The proprietary protocol used is a UDP-based communication protocol that connects main components of the system over an IP network. The protocol implemented includes a set of messages which are used to flash the microcontroller code memory remotely. We have developed an application which performs this task, which also allows specialists to update the firmware of the devices by means of a local Ethernet connection or by Internet.

This protocol implements an approach based on symmetric cryptography to deal with these questions. SHA-2 (Secure Hash Algorithm) is used to calculate a hash of the packet payload and the resultant value is encrypted, jointly with the heading, with AES (Advanced Encryption Standard), by means of a symmetric key shared among the entities of the system. The message is completed with a CRC of the whole packet.

The SHA hash assures the integrity of the payload, and the AES encryption applies authenticity to the information transmitted because the synchronization between the sender and the receiver (included in the heading) is hidden. Confidentiality is not directly provided for the packet payload because encryption is supposed to be included only in desired messages. Alarm messages, on the other hand, can offer encryption by themselves. At the service level, confidentially is offered by means of a secure HTTP access.

4.2. User interface

The use interface permits to the user to introduce initial data to configure the system, and several options to visualize data on real time of the system state or data historic.

The menu bar includes the following options:

1) **Data**
    - Insert data
        - Water analysis
        - Fertilizer analysis
        - Balance
        - Drainage analysis
    - List of…
    - Configuration of the system

2) **Nutrition**
    - Fertilizer cubes
        - New cube
        - Edit cube
        - Delete cube

3) **Alarms**
    - New alarm
    - List of alarms

4) **Irrigation**
    - Irrigation cube
        - New cube
        - Edit cube
        - Delete cube
    - Irrigation scheduling

5) **Data analysis**
    - Representation graph

The first option of the menu, **Data**, allows to the user to introduce data analysis of water, fertilizers, balances or drainages. These data are used in the operation of the system. Moreover, it include options to list, modify or delete the analyses, and other data of the configuration of the system as per example the analyses selected.

The second option, **Nutrition**, includes the data for preparing the small cubes of the nutrition unit in Figure 8. The user has to select the fertilizers (principal and secondary) in each fertilizer cube.

The next option, **Alarm**, provides the possibility of register alarms in to an input and generate an action about an output. These events are notified to users through SMS, email or advice message.

In the fourth option, **Irrigation**, the user can insert data related with the irrigation of the system. For the one hand, the configuration of the irrigation cube, and for the other hand, the user can specify the parameters of a scheduled irrigation.

Finally, in the **Data Analysis** option the user can visualize data in graph and export data in Excel.

5. Control Unit

Once reviewed the general architecture, this section is focused on **Control Unit (CU)**. The objective of the CU is manage the other units detailed in the project (nutrition unit, disinfection unit and purification unit) to prepare the nutritive solution.

5.1. Main operation

Figure 8 shows the cycle to prepare the nutritive solution of irrigation.

The nomenclature of the cubes is:

- Mixture cube, named “Cube A”, is where mix is performed. UC decides the quantity of each cube that is necessary to mix. The nutrition unit is composed of fertilizer cubes and Cube A.
- Disinfected drainage cube, named “Cube B”.
- Purification cube, named “Cube C”.
- Drainage cube, named “Cube D”.

In this first experiment one irrigation cube is included in the system. The additional two irrigation cubes are used in a second experiment in 2017.

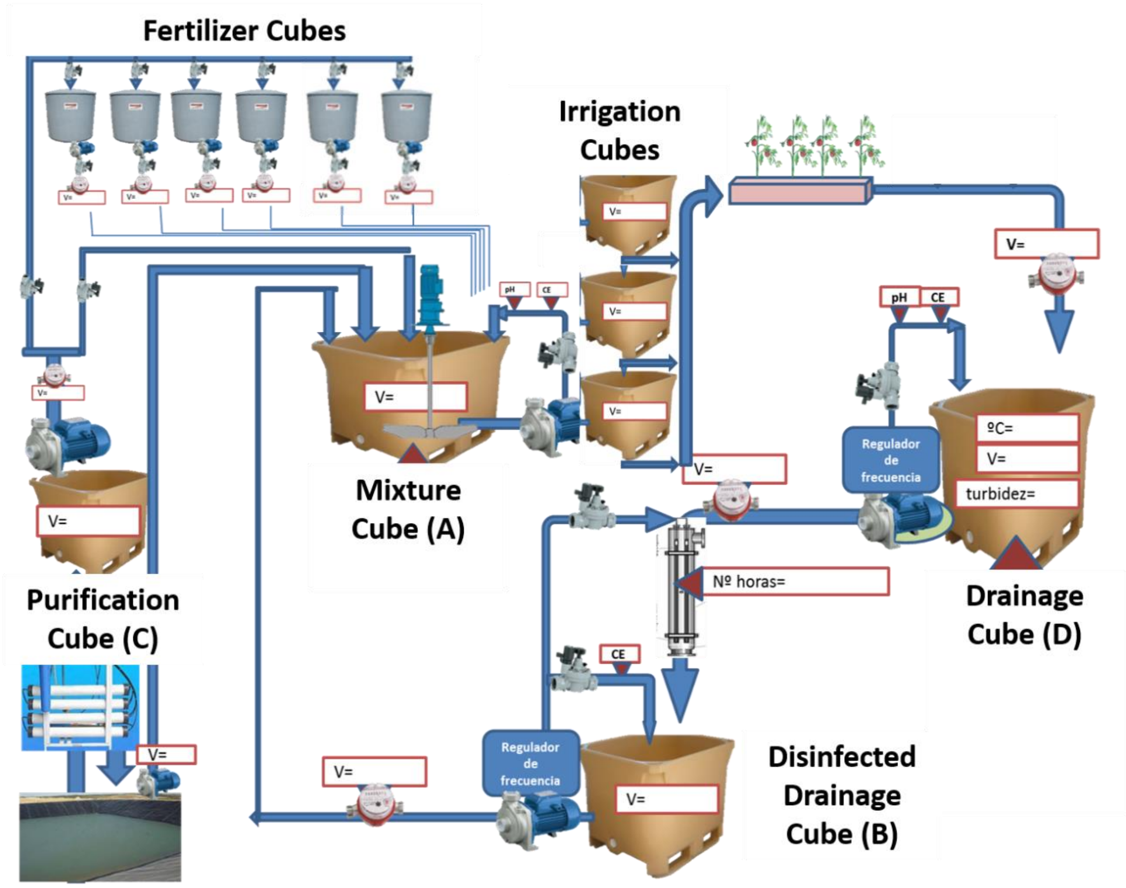


Figure 8: Cycle to prepare the nutritive solution

The operation of this system is represented in the diagram of Figure 9. The final mixture of water and nutrients will be dumped in cube A when the process finish, and then this will be available to irrigate. To prepare the mixture in cube A, first it must check if there are disinfected drainage in cube B.

On the one hand, if there are not disinfected drainage in cube B, new nutritive solution must be prepared. For making this nutritive solution two parameters are necessary (see Figure 10):

- Quantity of litres to prepare.
- The nutritive composition.

Then, it must check if there are these nutrients in all small cubes or fertilizer cube (the final prototype will include 6 fertilizer cubes). If there are not, a warning message is sent to the user to notify it the composition of the necessary nutrients. If there are enough nutrients in the small cubes to prepare the mixture in cube A, it must calculate the quantity of each one of these.

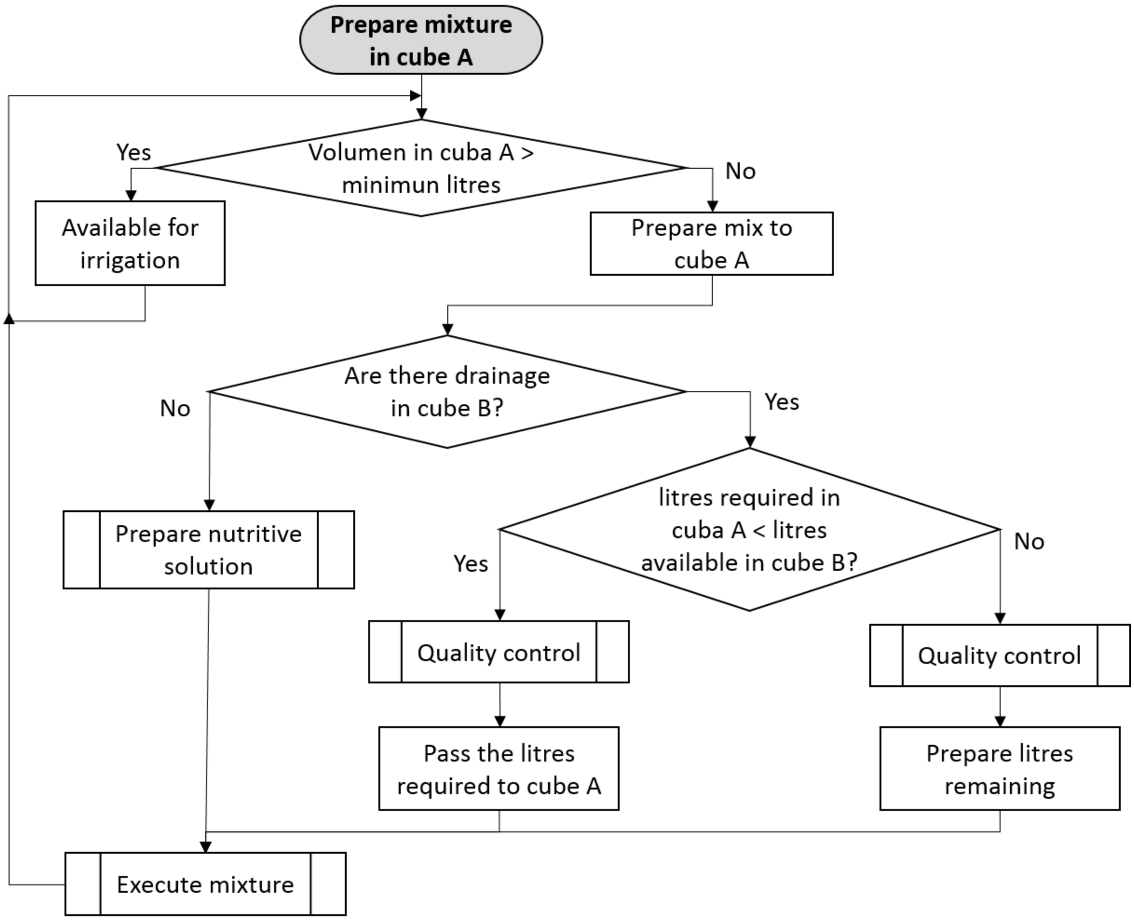


Figure 9: Main flowchart of the Control Unit.

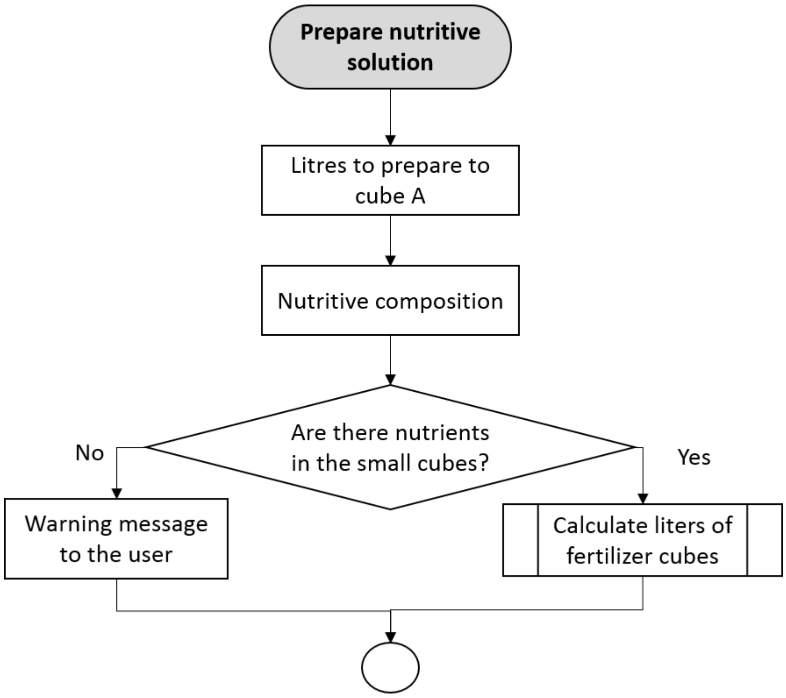


Figure 10: Flowchart of procedure “Prepare nutritive solution”

To calculate the litres of the fertilizer cubes for making mixture must consider the balance concentration selected (desired concentration in mixture) by the user. Then, it compare the nutrients that we want in the mixture (the balance concentration) with the nutrients of the fertilizers in the small cubes.

Once the concentration of each fertilizer cube is calculated, this value is converted to litres. With these data the execution of the mixture can be already performed.

On one hand (as you can see in Figure 9) if there are drainage in cube B, it must check if the quantity is bigger or smaller than the necessary one. In both cases the solution has to pass a quality control. When the control is satisfactory, if the quantity of litres required in cube A is less than the quantity of litres available in cube B, it must pour the numbers of litres necessary to cube A. On the other hand, if the quantity of litres required in cube A is bigger than the quantity of litres available in cube B, the quantity of litres remaining must be prepared. Finally, the process continue with the executing of the mixture so the nutritive solution in mixture cube is already ready to pass to one of the irrigation cubes to apply the watering.

5.2. Quality control

When the system can reuse disinfected drainage is necessary to check two parameters: the electronic conductivity (EC) and the nutrients of the disinfected drainage.

Firstly, as you can see in Figure 11, it compare the EC of the disinfected drainage in cube B with a maximum value fixed by the user. If the first is larger than the second, then an adjustment of the EC is necessary. Otherwise, directly it check the nutrients of the disinfected drainage in comparison with the balance selected by the user.

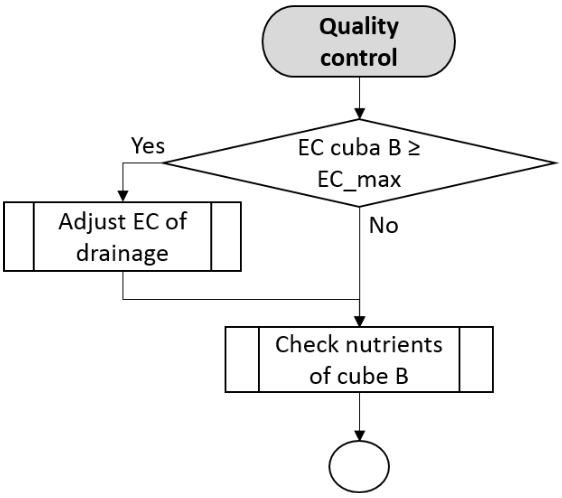


Figure 11: Flowchart of procedure "Quality control"

5.2.1. Electrical Conductivity (EC)

Several waters are mixed to adjust the EC of drainage. The main aim is maximize the use of drainage, second is use reservoir water to make up the EC, and finally, use desionized water if it is necessary.

First, we calculate the quantity of drainage and of reservoir water to get a mixture with a good EC (case 1 in Figure *12*). If this value of EC is correct, we pass to check nutrients, but if the calculated value of EC is larger than the fixed EC, then we calculate again using desionized water too (case 2). Finally, the latest estimates are made with less drainage and more quantity of desionized water.

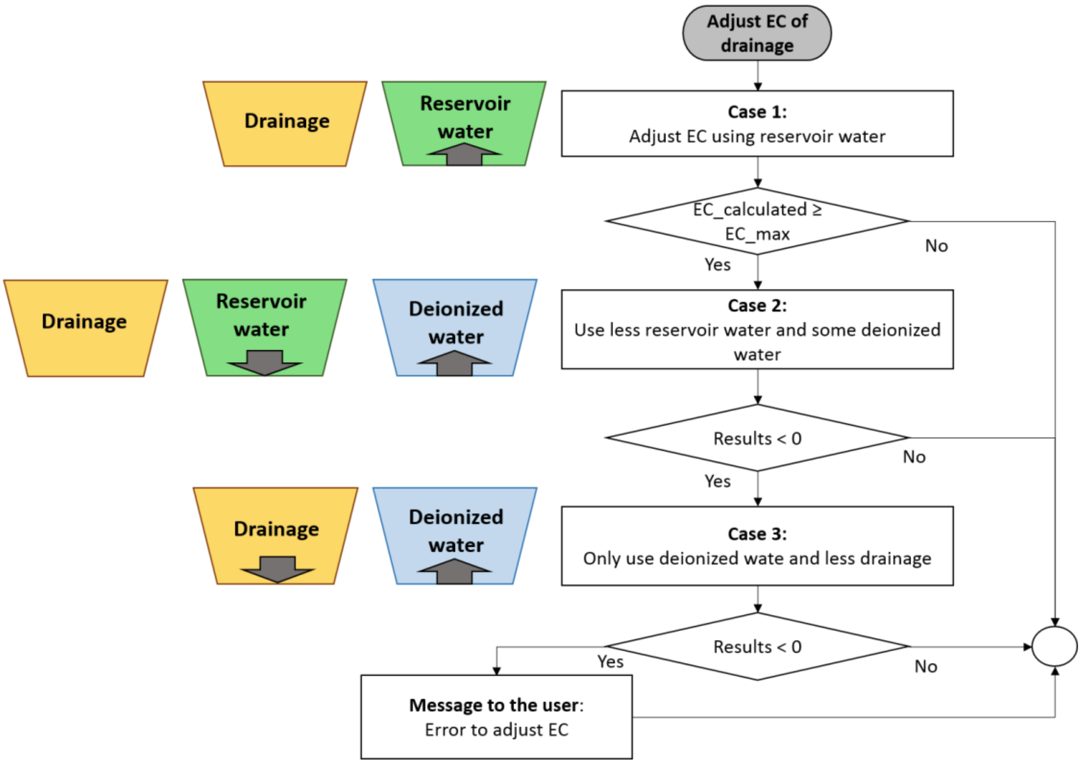


Figure 12: Flowchart of procedure “Adjust EC of drainage”

5.2.2.Nutrients

The check of nutrients consists of comparing the concentration of each nutrient with the limits fixed by the user. If the value of a nutrient is less than the fixed by the user, we calculate the new concentration providing nutrients missing. However, if the value of a nutrient is bigger than the fixed by the user, we calculate the new concentration making a dilution of the excess nutrients.

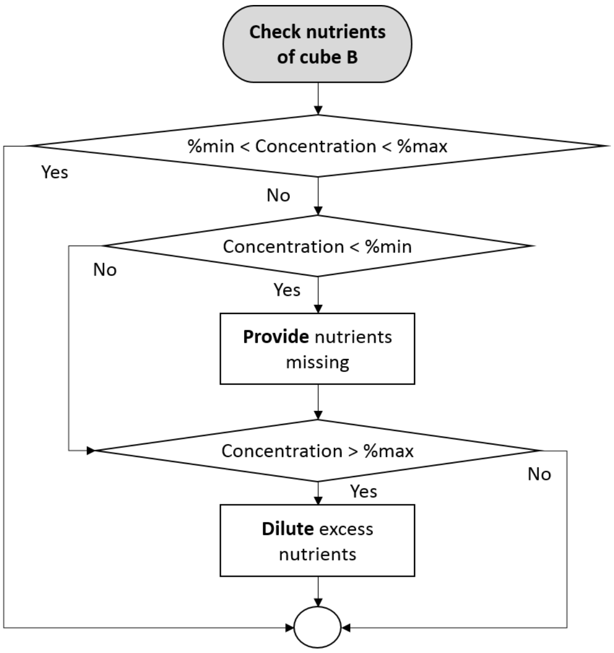


Figure 13: Flowchart of procedure “Check nutrients of cube B”

5.3. Execute mixture

Once that the nutritive solution is calculated, it can run the preparation following the steps detailed in Figure 14. When the mixture execution is ready, the nutritive solution is poured to the irrigation cube indicated by the user.

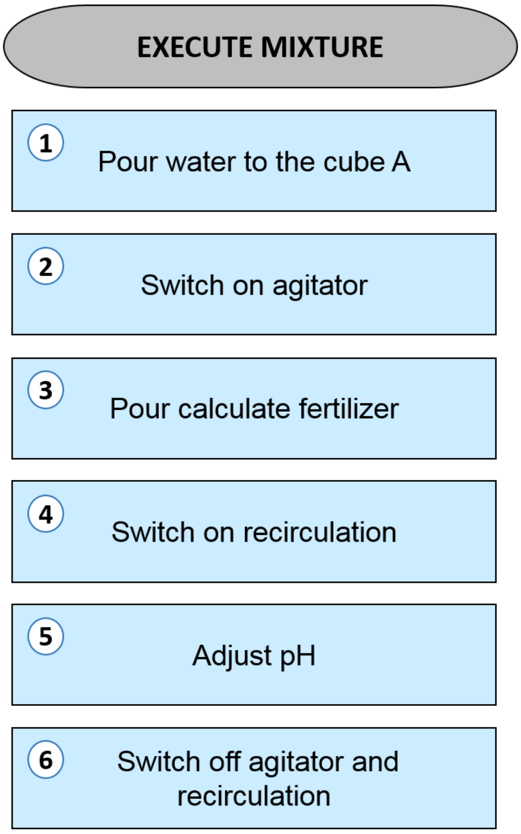


Figure 14: Steps of procedure “Execute mixture”

5.4. Sensor database

To perform this control the Control Unit is composed of several digital or analogic inputs and outputs. Next figures (Figure 15 - Figure 23) details the inputs (analogic or digital) and outputs (analogic or digital) of each one of the modules and the type of measure necessary. In all figures, the nomenclature is the following: AI (Analogic Input) marked in green color, DI (Digital Input) marked in blue color and DO (Digital Output) marked in yellow color. Moreover, in each I/O is indicated the tag used in AutoCAD schema.

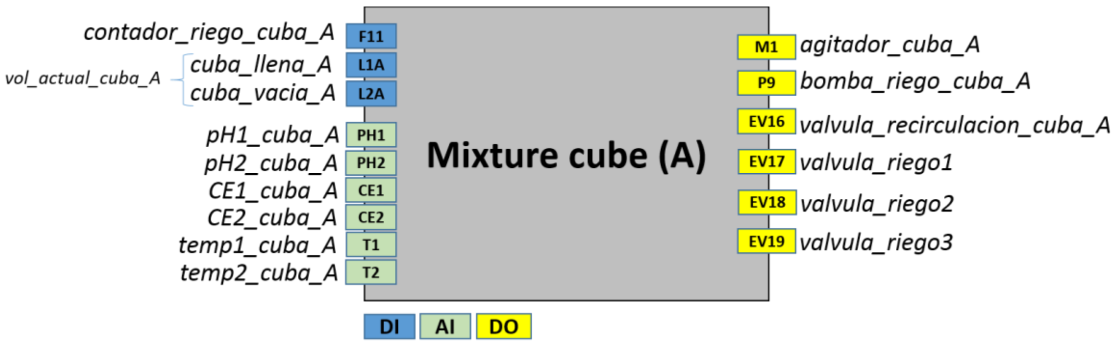


Figure 15: I/O Mixture Cube – A

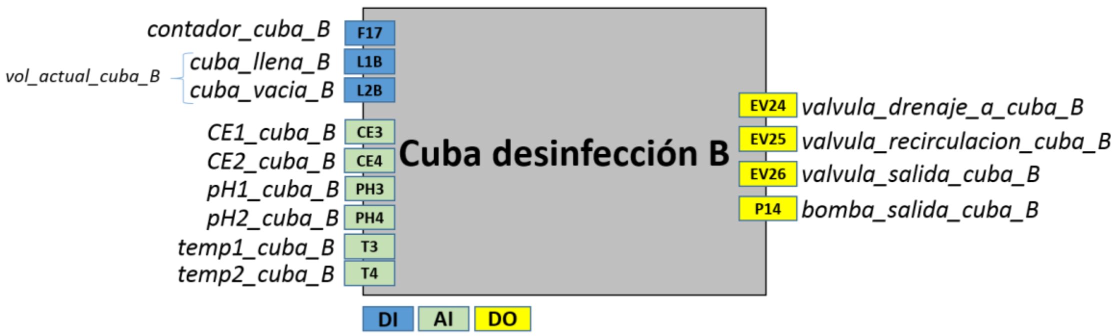


Figure 16: I/O Disinfection Cube – B

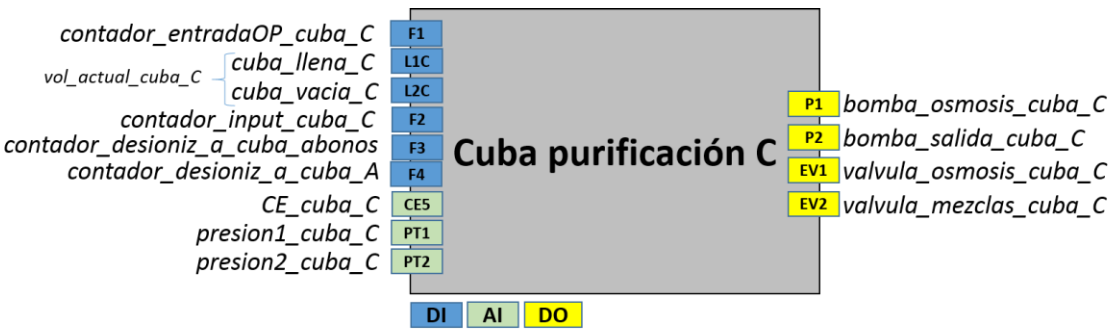


Figure 17: I/O Purification Cube

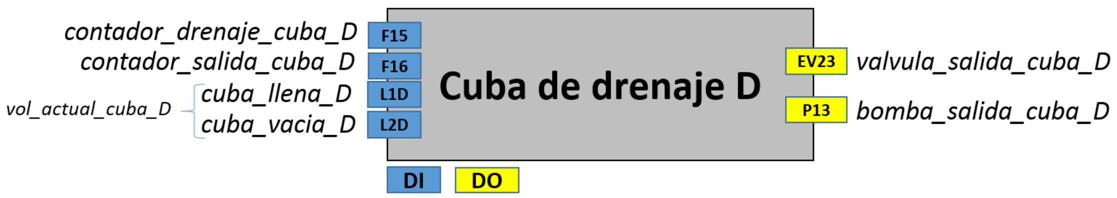


Figure 18: I/O Drainage Cube

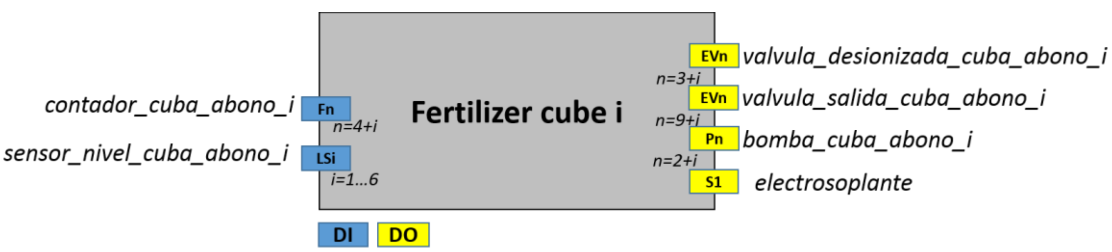


Figure 19: I/O Fertilizer Cube

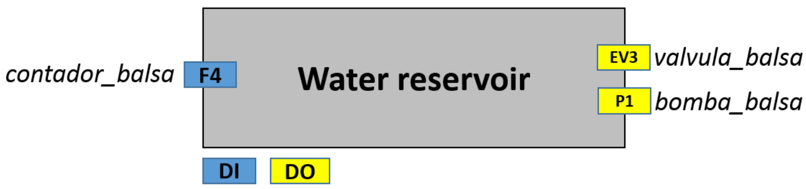


Figure 20: I/O Water reservoir

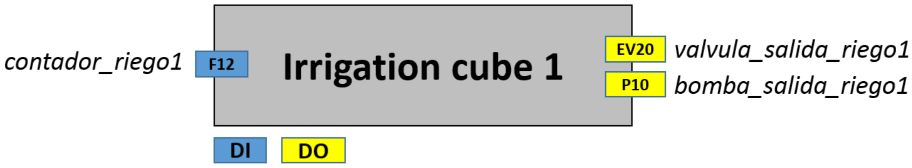


Figure 21: I/O Irrigation cube 1

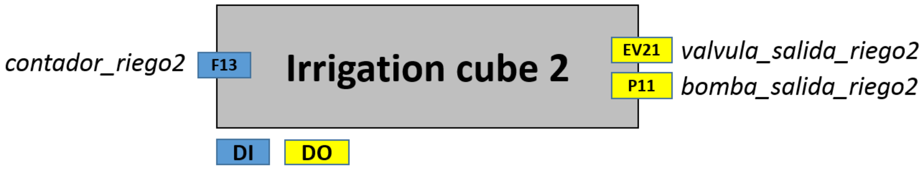


Figure 22: I/O Irrigation cube 2

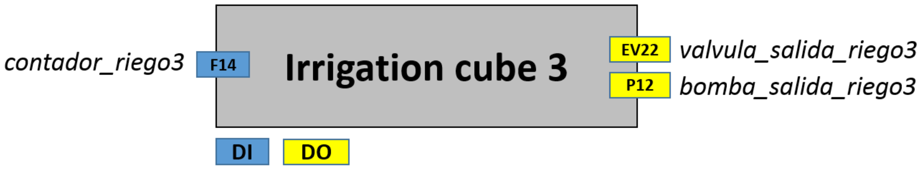


Figure 23: I/O Irrigation cube 3

Finally, Table 1 summarizes the quantity of inputs and outputs classified by type (analogic or digital). The control unit manage four big cubes (A, B, C, y D), 6 fertilizer cubes, the water reservoir and the irrigation cubes.

Table 1: I/O of all cubes

I/O	Quantity
AI	15
DI	32
DO	43
Total	**90**

Six automatas are necessary to connect all I/O of the Table *1*. Two of them will run as a masters and four as slaves. Therefore, the system will be controlled by two automatas. The operation of each one is showed in Figure 24 and Figure 25.

The 'Automata 1' (A1) (see Figure 24) manages the six fertilizer cubes, as the preparation with desionized cube as the control of the volume with the level sensors. The A1 receives the volume of each fertilizer cube to pour to desionized water mixture cube A and then activate or deactivate the necessary valves. Moreover, A1 controls the volume of each fertilizer cube. When a fertilizer cube is full, it stops the input of water and sends an alert to the user. When a fertilizer cube is empty, sends an alert to the user.

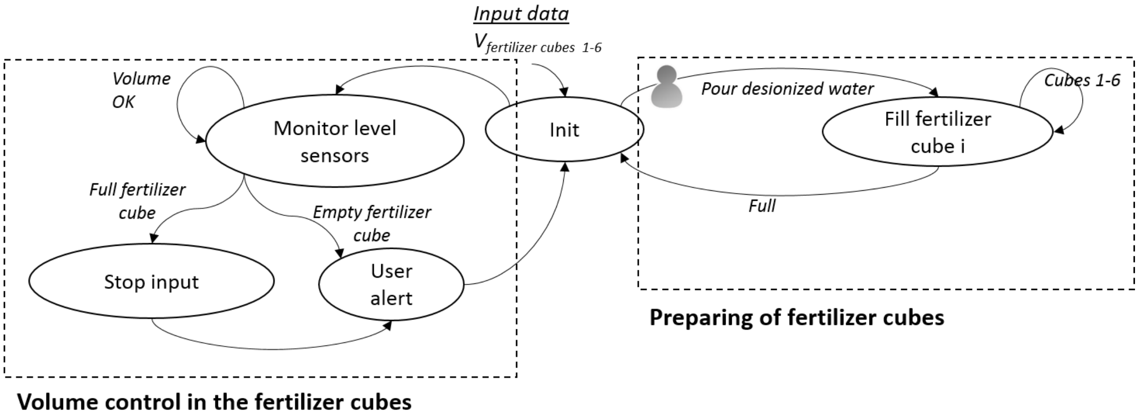
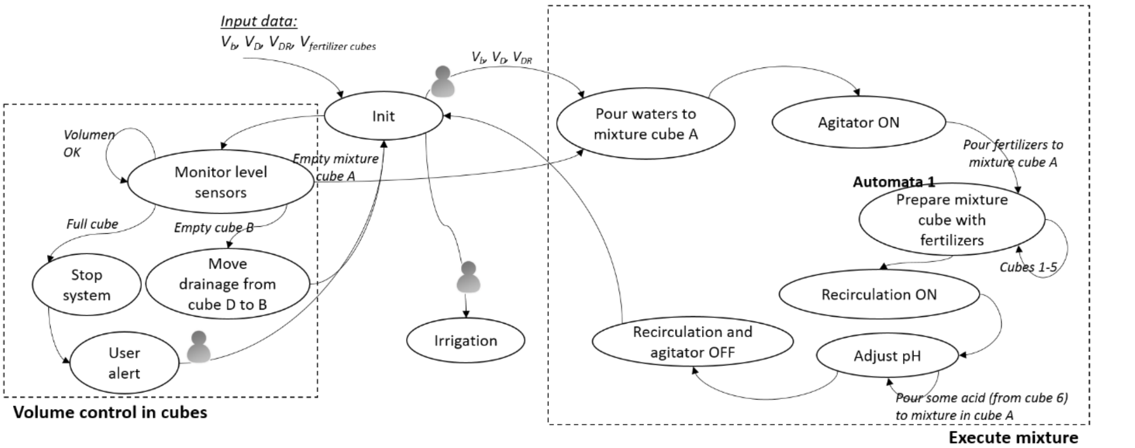


Figure 24: 'Automata 1' operation

The 'Automata 2' (A2) (see Figure 25) with four slaves manages the big cubes. The main aim is to prepare nutritive solution in mixture cube A using disinfected drainage, desionized water and reservoir water.

First, A2 receives the volume of each cube to pour in mixture cube A. With these data the A2 starts the execution of mixture (process in Figure 14). When the execution finish, the A2 manages the irrigation following the scheduled irrigation by the user.

Moreover, A2 controls the volume of each cube. When a cube is full, A2 stop the system and send alert to the user. When the mixture cube A is empty, the A2 begins a new nutritive solution. If the disinfected drainage cube is empty, A2 activates valves and pumps necessaries to move drainage from cube D to B.



V_b: reservoir water volume
V_D: desionized water volume
V_{DR}: drainage volume
$V_{fertilizer\ cubes}$: fertilizer cubes volume

Figure 25: 'Automata 2' operation

5.5. Interaction between devices and user application

An overview of system operation is detailed in Figure *26*. First, the user inserts configuration data and analysis data in the user web application. This information is used to configure the fertilizer cubes (step two in Figure *26*). When the user makes this configuration, the web application send the volumes of each fertilizer cube to the 'automata 1'. This automata active valves and pumps to pour the quantity of desionized water indicated in each small cube.

Before beginning with the preparation of nutritive solution, the user select the mode of irrigation and then, the web application sends to the 'automata 2' the configuration parameters.

Next step consist of making the calculations to prepare the solution nutritive. With the information inserted by the user, the application calculates the quantity of each cube (litres of fertilizers cubes, litres of reservoir water (Vb in Figure *26*), litres of desionized water (VD) and litres of drainage (VDR)) necessary to execute the mixture. This results are sent to the 'automata 2' and then, it runs the execution of the mixture. In this process, the 'automata 2' requests to the 'automata 1' that pour fertilizers of the small cubes in the mixture cube.

When it finish the mixture execution, the 'automata 2' inform to the user application and data are stored. In this moment there are already water to irrigation.

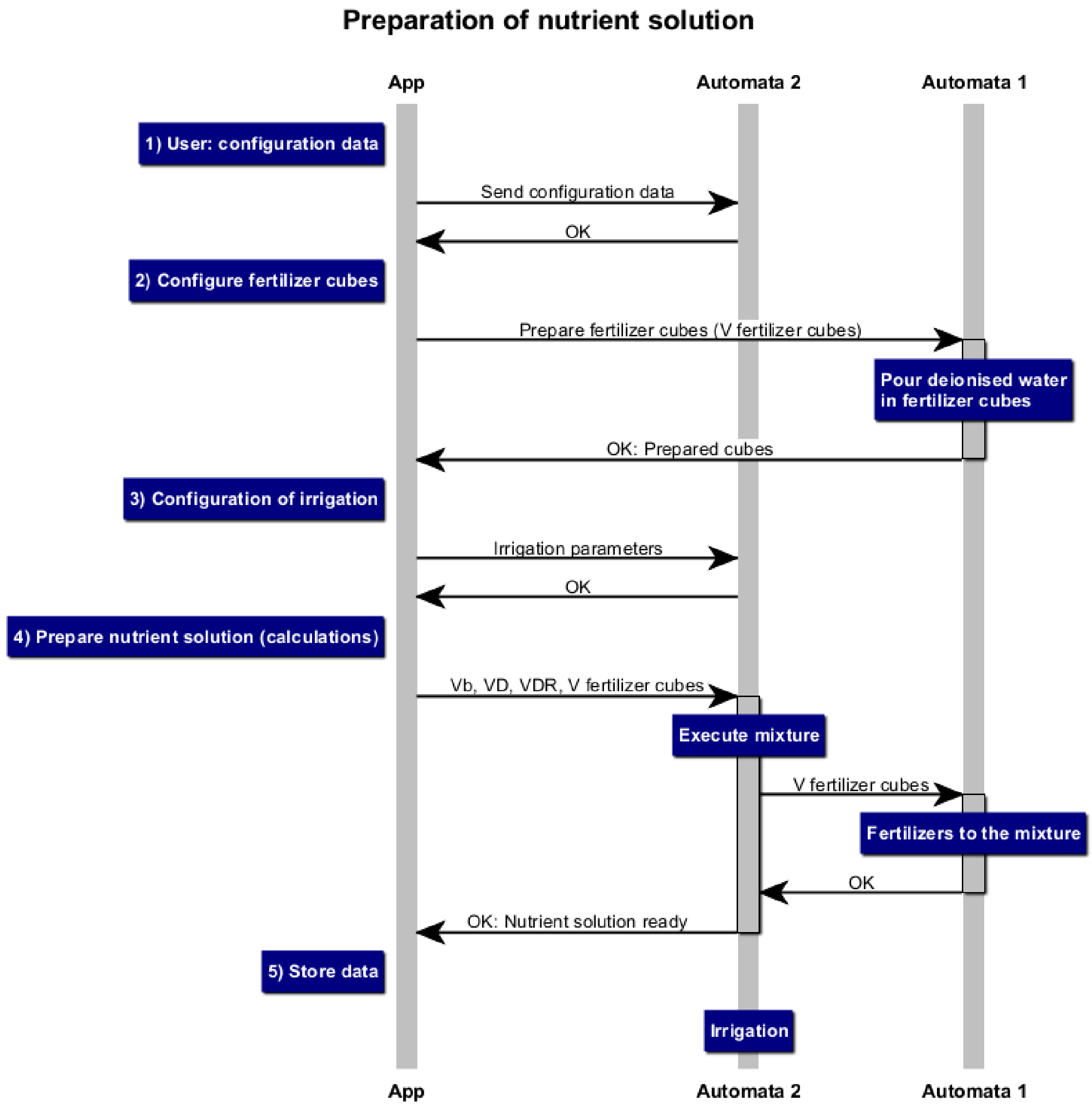


Figure 26: Interaction between automatas and user application

6. Conclusions

Summarizing, we have a greenhouse control system with **90 I/O** (sensors and actuators), with the distribution shown in the Table *1*: I/O of all cubes. The controller used in this use case is the described previously, where each unit (master or slave) has 16 I/O, which can be configured as it will be necessary in each case (analog or digital I/O).

The initial prevision is the use of two master and 4 slaves as hardware architecture, that is, six controllers. The communication between the two masters will be Ethernet or WIFI, depending on the facilities. During the setup of the system and the configuration of all the controllers will be the final decision (depending of the physical distribution of the sensors).

About the external communication architecture, ADSL communication will be used since the greenhouse pilot have this option. Cellular communications will be also used as alternative and redundant channel for SMS alarms.

Furthermore, clearly the communication architecture that will be used is the multilayer version, where the different controllers implement the main control of the system. From the cloud platform we will be able to supervise, command and change configuration and setup, but the system is reliable and is able to operate in absence of Internet communication.