

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCOLA POLITÈCNICA SUPERIOR DE GANDIA

Grado en Ing. Sist. de Telecom., Sonido e Imagen



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCOLA POLITÈCNICA
SUPERIOR DE GANDIA

**“DESARROLLO DE UN SISTEMA
ELECTRÓNICO PARA CONTROLAR EL
CRECIMIENTO DE PLANTAS DE UN
HUERTO VERTICAL HIDROPÓNICO”**

TRABAJO FINAL DE GRADO

Autor/a:

PAU GOMAR LLINARES

Tutor/a:

JOSÉ MARÍN-ROIG RAMÓN

GANDIA, 2020

Contenido

| | |
|--|----|
| Resumen..... | 3 |
| Palabras clave..... | 3 |
| Capítulo 1. Introducción..... | 4 |
| 1.1 Objetivos | 4 |
| 1.2 Metodología | 5 |
| Capítulo 2. Definición y especificaciones del proyecto..... | 6 |
| 2.1 Descripción del sistema actual | 6 |
| 2.2 Concepción del nuevo sistema: determinación de alarmas y variables de control..... | 7 |
| 2.2.1 Control sensores de agua | 7 |
| 2.2.2 Control y estado del proceso de llenado de abono..... | 8 |
| 2.2.3 Control y estado del proceso de llenado de agua..... | 8 |
| 2.3 Planteamiento del nuevo sistema automatizado | 9 |
| Capítulo 3. Herramientas de desarrollo | 11 |
| 3.1 Hardware..... | 11 |
| 3.1.1 Placa Arduino..... | 11 |
| 3.1.2 Sensores de agua | 11 |
| 3.1.3 Detectores de nivel..... | 12 |
| 3.1.4 Electroválvula | 13 |
| 3.1.5 Multiplexor | 13 |
| 3.1.6 Relé..... | 14 |
| 3.1.7 Módulo wifi ESP8266..... | 14 |
| 3.1.8 Fuente alimentación..... | 15 |
| 3.1.9 Bomba de agua, depósito y material de laboratorio diverso | 15 |
| 3.2 Software | 15 |
| Capítulo 4. Diseño del proyecto | 17 |
| 4.1 Diseño hardware..... | 17 |
| 4.1.1 Sensores de Agua..... | 17 |
| 4.1.2 Depósitos de agua y abono | 18 |
| 4.2 Diseño software | 21 |
| 4.2.1 Algoritmo del proceso de llenado..... | 21 |
| 4.2.2 Algoritmo de verificación de los detectores de agua | 21 |
| 4.2.3 Algoritmo para comprobar depósito de abono..... | 22 |
| Capítulo 5. Implementación del sistema..... | 23 |
| 5.1 Implementación Software | 23 |

| | |
|--|----|
| 5.1.1 Módulo 1: Sensores de agua y multiplexor | 24 |
| 5.1.2 Módulo 2: Electroválvulas y sensores de nivel | 25 |
| 5.1.3 Módulo 3: Sensor Abono..... | 27 |
| 5.1.4 Módulo 4: Envío del correo | 28 |
| 5.2 Implementación Hardware..... | 32 |
| 5.2.1 Implementación de los sensores de agua | 33 |
| 5.2.2 Implementación depósito de agua | 34 |
| 5.2.3 Implementación depósito de abono..... | 36 |
| Capítulo 6. Pruebas de verificación y comprobación del sistema | 37 |
| 6.1 Verificación de los sensores de agua..... | 38 |
| 6.2 Verificación de los sensores del depósito de abono | 39 |
| 6.3 Verificación de los sensores del depósito de agua | 39 |
| Capítulo 7. Conclusiones y futuras líneas de trabajo | 41 |
| Tabla de ilustraciones..... | 43 |
| Referencias | 44 |

Resumen

En el siguiente proyecto, se pretende realizar el diseño e implementación de un sistema de automatización de riego y abonado para un huerto hidropónico en el Campus de la UPV de Gandía, hasta ahora gestionado periódicamente de forma manual. Por lo tanto, el motivo que impulsa la ejecución de este proyecto es automatizar dicho huerto para que pueda ser controlado y gestionado de manera automática a través de microcontroladores, intentando que sea lo más eficiente posible y que tenga la capacidad de gestión de todos los dispositivos electrónicos que lo conforman. Así pues, por una parte, se generará el código, utilizando la placa Arduino, que permita el control de todos los dispositivos. Por otra parte, se generará el diseño electrónico y se procederá a la implementación de todos los dispositivos y materiales necesarios para alcanzar dicho propósito. Por último, se implementará un sistema para el envío de correos de control con el propósito de ver el estado de todos los sensores del sistema.

Abstract

In the following project, it is intended to carry out the design and implementation of an irrigation and fertilizer automation system for a hydroponic garden on the UPV Campus in Gandía, until now managed frequently manually. Therefore, the motive behind the execution of this project is the said garden is automated so that it can be automatically controlled and managed through microcontrollers, trying to make it as efficient as possible and have the capacity to manage all the devices. electronic that conform it. So, on the one hand, it will generate the code, use the Arduino board, allow control of all devices. On the other hand, it will generate the electronic design and will proceed to the implementation of all the devices and materials necessary to achieve said purpose. Finally, a system will be implemented for sending control emails with the purpose of seeing the status of all the sensors in the system.

Palabras clave

Arduino, huerto, hidroponía, riego, automatización.

Key Words

Arduino, vegetable patch, hydroponics, watering, automation.

Capítulo 1. Introducción

Básicamente, el fenómeno y la tendencia de crear huertos urbanos, comenzó a finales del siglo XX a raíz de una necesidad social. Es de gran interés ya no solo para el medio ambiente sino también para las grandes ciudades el uso de este tipo de huertos, que favorecen los cultivos ecológicos, aportan más valor a lo que comemos e impulsan los productos hortícolas frescos. Cubriendo las ciudades de verde, rompemos el impacto que supone el cemento y los ladrillos mediante este tipo de huertos revitalizando así los barrios de la ciudad.

El principal objetivo de este tipo de huertos de un tiempo a esta parte es crear y cultivar alimentos que se puedan transformar en una fuente importante de recursos para las familias, suministrando todo tipo de alimentos orgánicos, frescos, sanos y de alta calidad quedando todos ellos libres de sustancias químicas altamente perjudiciales para la salud y el consumo humano.

Como se puede observar, el campo está cada día más presente en la ciudad y en estos tiempos de contaminación y consumismo, un gran número de ciudadanos opta por volver a poner las manos en la tierra. Esta clase de proyectos se expande poco a poco a las ciudades y hay una serie de dificultades que se deben superar desde la instalación de los huertos en terrazas o tejados comunitarios hasta la implementación en balcones particulares.

Los tipos de huerto de autoconsumo como el que planteamos en este proyecto de final de grado no necesitan grandes espacios para llevarse a cabo, ya que se pueden explotar en lugares reducidos adaptando el cultivo a las condiciones de cada hogar.

Por eso, con el fin de mimetizarse con la sociedad y estar al frente de las exigencias y demandas de esta, se va a utilizar la instalación del huerto vertical que la Cátedra de Innovación del Campus de la UPV de Gandía hizo instalar en 2018 a través de la Startup Optimus Garden, con la que se pretende automatizar esta área del Campus de Gandía y convertirla en un trozo de zona verde comprometida con la horticultura y el medio ambiente.

1.1 Objetivos

En este proyecto se pretende poner en funcionamiento un jardín hidropónico programado con Arduino, que sea eficaz, autosuficiente y capaz de garantizar el perfecto funcionamiento del jardín de la manera más segura y viable posible. Por lo tanto, las metas que se quieren conseguir son las siguientes:

- El objetivo principal del proyecto es el diseño y montaje completo de un sistema de automatización de riego hidropónico. La finalidad de automatizar dicho huerto es tener la información del estado de los dispositivos que se van a utilizar, es decir, tener controlados el sistema de llenado de agua, el sistema de abonado, y el sistema de detección de agua en cada una de las tuberías verticales. Así pues, para llevar a cabo lo que se ha mencionado anteriormente, se pretende mejorar las condiciones del huerto actual que posteriormente se explicarán.

Como parte de los objetivos secundarios del proyecto, encontramos los que se detallan a continuación:

- Diseño de un sistema electrónico para automatizar el huerto, así como su implementación, es decir, añadir los sensores y componentes pertinentes para hacerlo autosuficiente y que pueda ser controlado de manera electrónica y de la forma más cómoda posible.
- Diseño e implementación del sistema software mediante Arduino para el control de todos los sensores y dispositivos electrónicos que conformarán el proyecto.

1.2 Metodología

Este proyecto, al igual que todos aquellos relacionados con la ingeniería, sufrirá diversas modificaciones a lo largo de su diseño e implementación para ir así ajustándose a la demanda y los objetivos previamente establecidos. Así pues, la metodología seguida se divide en cuatro fases:

- En la primera fase del proyecto, se intentará crear e idear cual será el proyecto que se quiere desarrollar teniendo en cuenta los materiales de partida y los objetivos, es decir se intentará concebir la idea general del huerto y que es lo que se pretende conseguir junto con José Marín-Roig Ramón.
- La segunda fase, será el diseño de todo el proyecto, tanto la parte software como hardware. Se establecerá el diseño y el funcionamiento del proyecto, es decir, cual es la situación actual del proyecto y cuáles son las mejoras y modificaciones que se quieren realizar. Así pues, se pasará a estudiar cual será la función que tiene que realizar cada uno de los dispositivos y por tanto se establecerá que especificaciones se precisan para cada uno de ellos, además de realizar el diseño software para poder implementar el código en la fase siguiente. En definitiva, se diseñará el código que controlará todo el sistema, y se diseñará el esquema hardware del proyecto.
- Una vez diseñado el proyecto completo, se pasará a la implementación de este. En primer lugar, se implementará el código software utilizando Arduino. En segundo lugar, se pasará a la implementación del sistema hardware, por tanto, la dinámica de trabajo consistirá más en horas de laboratorio para la implementación de la placa y poder soldar todos los componentes. Para esta fase del proyecto se accederá al laboratorio de la UPV de Gandía y se contará con la ayuda del Técnico de Laboratorio Antoni Bou. Antes de soldar la placa final, se realizará un pequeño prototipo con una placa board y leds para simular el comportamiento de todos los dispositivos y comprobar que el código funciona perfectamente.
- Por último, el proyecto pasará por una última fase de pruebas de verificación en el laboratorio para comprobar que todo funciona de acuerdo con lo establecido y pueda ser montado posteriormente en la estructura metálica de la UPV.

Capítulo 2. Definición y especificaciones del proyecto

En este capítulo se explicará cual es la base del proyecto y cuáles son los materiales iniciales de los que se parte para la realización del huerto que se plantea. Así pues, se explicarán las mejoras que se pretenden realizar junto con un esquema del proyecto introduciendo los materiales utilizados y el rol de estos dentro del proyecto. En definitiva, se fijarán las especificaciones para el diseño e implementación del proyecto.

2.1 Descripción del sistema actual

Se parte de una estructura de metal con siete tiras verticales donde se ubicarán nueve plantas en cada una de estas tiras, que son abastecidas por un sistema de riego alimentado por una bomba de agua y un depósito de agua que actualmente son controlados de manera visual. Hasta el momento, el llenado del depósito de agua se realiza de manera periódica y manual desde una toma de agua cercana al huerto.

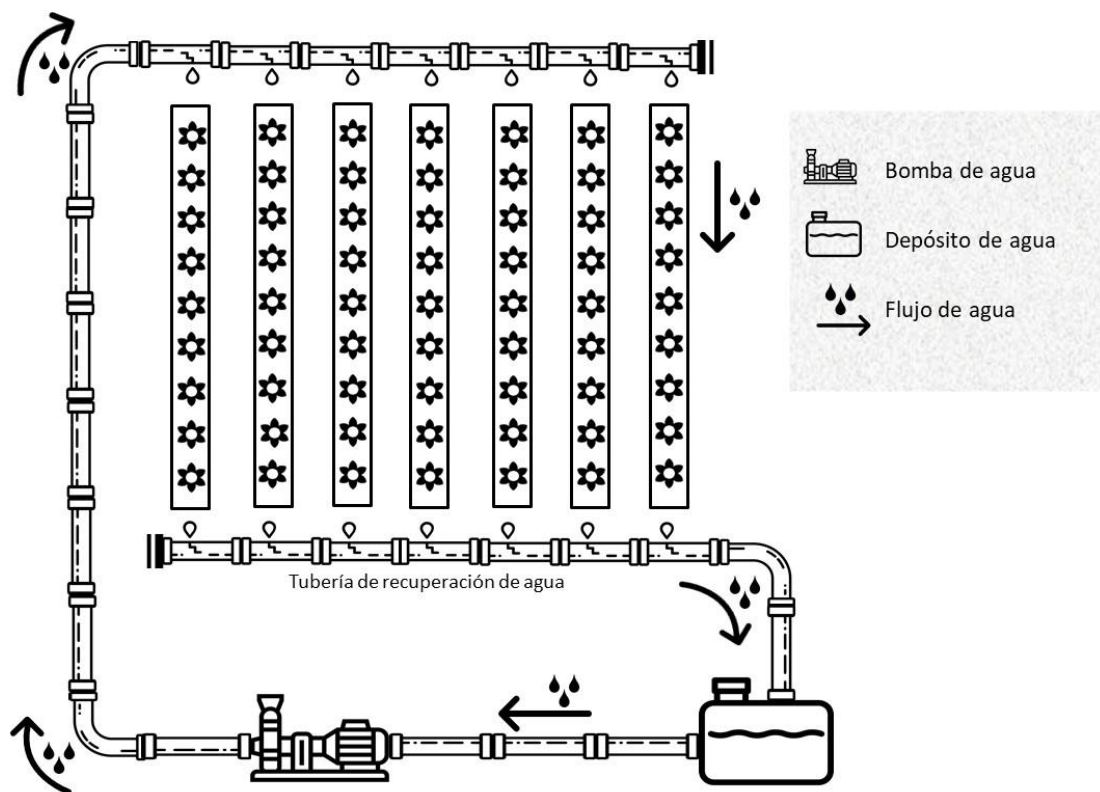


Ilustración 1. Estructura de la que se parte para realizar el proyecto.

2.2 Concepción del nuevo sistema: determinación de alarmas y variables de control

Para poder gestionar el sistema de manera eficiente, surge la necesidad de decidir que queremos controlar, y cuál es la información que queremos obtener para poder verificar y comprobar que todo funciona de manera correcta y adecuada. Se necesitará programar el sistema para que envíe correos periódicamente con toda la información, en el que se describirán el estado de cada uno de los sensores, así como el estado de todos los procesos.

Puesto que se va a diseñar el sistema para que ayude al usuario, se tendrá que definir qué tipo de alarmas son necesarias, por tanto, para controlar todo el huerto se generará un sistema de alarmas mediante correos de texto.

Para controlar el sistema se ha determinado que es necesario generar este tipo de alarmas:

- control de los sensores de agua
- control del nivel de abono
- control de los procesos de llenado de agua

2.2.1 Control sensores de agua

Para comprobar que en las tiras verticales del huerto haya flujo de agua en todo momento, se utilizarán sensores de agua para su detección. Se enviarán cada cierto tiempo correos periódicos con el estado de los sensores. Así pues, las siete primeras líneas del correo mostrarán el estado de cada uno de los siete sensores de agua, este estado puede ser:

- **“OK”** Indicará que el sensor detecta agua y por lo tanto todo funciona de manera correcta.
- **“Sin agua”** Indicará que el sensor no está detectando flujo de agua y por tanto habrá que revisarlo de manera visual. Se podrían dar los siguientes casos:
 - Si al recibir el mensaje solo hay un sensor que marque “Sin agua”, indicará que o bien algo está obstruyendo el agujero por el que se deja caer el agua a esta tira vertical en concreto, o que el fallo se encuentra en el sensor de agua que no funciona como es debido.
 - Si por el contrario hay más de un sensor o bien todos marcan “Sin agua”, esto indicará que el problema puede estar en la goma principal que transporta el agua hasta la parte superior del huerto o que la bomba de agua no funciona adecuadamente.

2.2.2 Control y estado del proceso de llenado de abono.

En este apartado, se detallarán los mensajes de alarma correspondientes al proceso de llenado de abono.

Después de los siete mensajes correspondientes al estado de los sensores de agua, el siguiente mensaje hará referencia al estado del nivel de abono. El sensor de nivel del depósito de abono podrá generar los mensajes siguientes:

- **“OK”** Indicará que el depósito tiene abono líquido.
- **“Rellenar”** Indicará que el sensor de nivel detecta un nivel de abono bajo y por lo tanto habrá que rellenar el depósito de manera manual.

2.2.3 Control y estado del proceso de llenado de agua.

Por último, después de los mensajes de control de los sensores de agua y del nivel de abono, las dos líneas finales del mensaje enviado informarán del estado del depósito de agua y del estado del proceso de llenado en caso de que este esté en marcha. Por consiguiente, se podrá encontrar los siguientes mensajes:

- El primer mensaje indicará si el depósito está lleno, o si por el contrario el sensor ha detectado que el nivel de agua es bajo y el proceso de llenado del depósito está en marcha.
 - Mientras que el sensor inferior del depósito no detecte que el nivel de agua es bajo, en cada mensaje recibido se podrá leer el siguiente mensaje: **“Tanque lleno”**.
 - Una vez que el sensor inferior detecte que el nivel de agua es bajo, comienza el proceso de llenado del depósito. Entonces, se enviará un correo comunicando que el proceso de llenado está en marcha y se podrá leer el siguiente mensaje: **“Ongoing”**.
- La última instrucción del mensaje informará si el proceso de llenado se ha efectuado de manera correcta:
 - Al igual que el mensaje anterior, si el nivel de agua no es bajo se podrá ver que el tanque de agua está lleno: **“Tanque lleno”**.
 - Una vez que el llenado de agua se haya completado y además se haya vertido el abono líquido al depósito de agua, se enviará otro mensaje para informar que el proceso de llenado tanto de agua como de abono se ha realizado con éxito y aparecerá el siguiente mensaje: **“Completado”**.

2.3 Planteamiento del nuevo sistema automatizado

Para cumplir con las exigencias demandadas en el apartado anterior (2.2), en el siguiente apartado se explicará el funcionamiento de cada uno de los dispositivos, es decir, como va a funcionar el sistema completo. Así pues, para lograr que el huerto sea autosuficiente y pueda ser controlado de manera electrónica mediante Arduino, se añadirán sensores de agua en la parte superior de cada una de las tiras verticales de la estructura metálica para asegurar que ninguna de estas se quede sin agua. En caso de que la conexión por goteo se atasque, para evitar que se sequen las plantas, se utilizará una conexión Wifi, mediante el dispositivo ESP8266, para enviar un mensaje por correo electrónico que alerte de un posible fallo, bien en los sensores de agua, en la propia goma de goteo o en la bomba de agua.

Como se ha comentado anteriormente, el llenado del depósito se realiza de manera manual y lo que se pretende es automatizar el sistema de riego. Para llevar a cabo este proceso, se añadirán dos sensores de nivel en el depósito de agua, uno en la parte superior del depósito (SN1) y otro en la parte inferior del mismo (SN2). El sensor de la parte inferior del depósito indicará cuando el nivel de agua es bajo, y por tanto se procederá al llenado de agua de manera automática mediante la activación de una electroválvula (EV1). Por el contrario, el sensor superior del depósito indicará que el nivel de agua ha llegado a su nivel máximo y por consiguiente se finalizará este proceso de llenado con el apagado de la electroválvula (EV1).

Por otro lado, dispondremos de un depósito adicional con abono líquido controlado por otro sensor de nivel (SN3), que mediante el Arduino enviará una señal de alarma cuando el nivel de abono líquido sea bajo para proceder con el rellenado del depósito del abono, este sí, de manera manual. Además, se dispondrá de otra electroválvula (EV2) para verter el abono en el depósito de agua, así pues, cuando el depósito de agua esté completamente lleno una vez finalizado el proceso de llenado de agua, la electroválvula (EV2) que controla el abono se activará durante un tiempo determinado para verter el abono correspondiente a la cantidad de agua introducida en el tanque de agua.

Para la realización del sistema explicado anteriormente se necesitarán los siguientes materiales (que en el capítulo tres serán descritos de manera detallada) dispuestos en la manera en la que se explican en el siguiente esquema:

- Sistema electrónico basado en Arduino
- Sistema electrónico ESP8266
- Dos depósitos (uno para el agua y otro para el abono)
- Tres sensores de nivel de líquido (SN1, SN2 y SN3)
- Siete detectores de agua
- Dos electroválvulas (controladas por dos relés)
- Una bomba de agua

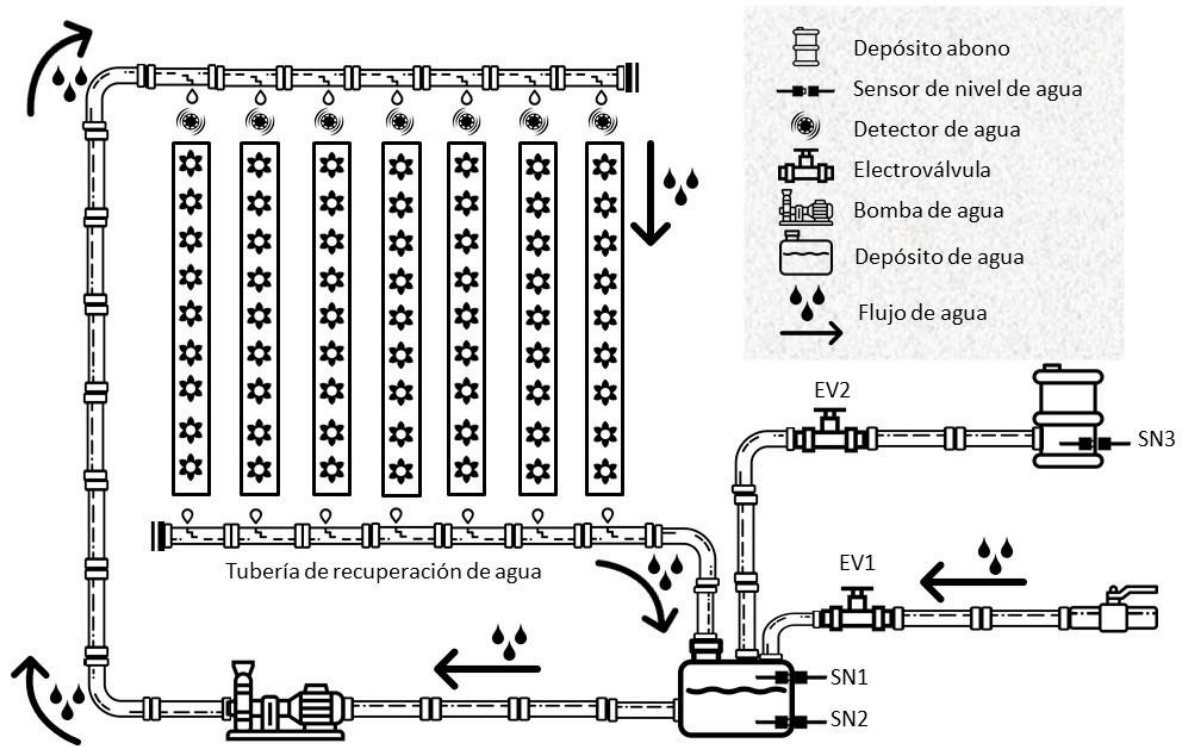


Ilustración 2. Esquema del proyecto con todos los componentes.

Capítulo 3. Herramientas de desarrollo

3.1 Hardware

En este apartado, se detallan las herramientas hardware utilizadas para la implementación de este proyecto, justificando la elección de estos y mostrando las características básicas de cada uno de los componentes de los que vamos a disponer.

3.1.1 Placa Arduino

Para la programación del código, utilizaremos la placa Arduino UNO que es una placa electrónica basada en un microcontrolador. Cuenta con:

- 14 entradas/salidas digitales, de las cuales 6 se pueden utilizar como salidas PWM (Modulación por ancho de pulsos) y otras 6 son entradas analógicas.
- Incluye todo lo necesario para que el microcontrolador haga su trabajo, basta conectarla a un ordenador con un cable USB o a la corriente eléctrica a través de un transformador.

A través de esta placa se puede trabajar de manera práctica y cómoda, se escoge trabajar con Arduino porque tiene muchas aplicaciones y usos, por su bajo coste, y como es de código abierto, facilita su combinación con otras plataformas, cosa que nos permite una alta flexibilidad y gran variedad tanto de placas como de componentes.



Ilustración 3. Placa Arduino UNO.

3.1.2 Sensores de agua

Con el propósito de detectar el agua en las columnas verticales del huerto, se utilizan sensores de agua muy usados en Arduino. El componente principal es un circuito amplificador que está formado principalmente por un transistor y unas líneas metálicas en el PCB. La resistencia tiene el valor de seguimiento alta hasta que una gota de agua corta la huella del sensor con conexión a tierra.

Las características principales del dispositivo y los motivos por los que se decide trabajar con este sensor son:

- Puede ser ampliamente utilizado en la detección de la precipitación, nivel del agua, incluso fugas.
- Son sencillos de utilizar, poseen una gran fiabilidad de uso.

- Su coste es muy asequible, además, este sensor es de bajo consumo de energía
- Alta sensibilidad.



Ilustración 4. Sensor de agua.

3.1.3 Detectores de nivel

Para la detección de nivel de agua y abono en ambos depósitos, se utilizará el interruptor de nivel de líquido LS309-31. Cuando el nivel del líquido llega a la altura del interruptor, el flotador cambia de posición. Un cierto ángulo y el contacto Reed invierten su estado debido a la acción del campo magnético del imán, que va alojado en el interior del flotador. El contacto Reed vuelve a su estado inicial cuando el flotador deja de estar a su altura. Se elige trabajar con este dispositivo por los siguientes motivos:

- Fácil de instalar, de larga durabilidad y alta fiabilidad.
- Adecuado para la detección de una amplia gama de líquidos.
- Este modelo tiene la característica que el contacto Normalmente Abierto que incorpora, si se le invierte la posición del mini sensor 180° se obtiene un contacto Normalmente Cerrado.
- Está dotado de un rango de temperaturas muy amplio (-30° hasta 130°) que se adapta perfectamente a las necesidades de nuestro proyecto.



Ilustración 5. Sensor de nivel.

3.1.4 Electroválvula

Con el fin de llenar los dos depósitos, se utilizarán dos electroválvulas (Seed Studio G1&2 Electric Solenoid Valve 713-111990004) de 12 voltios. Estas electroválvulas serán alimentadas por un transformador y las activaremos (ON/OFF) mediante un relé programado desde el Arduino. Se destaca su:

- Alta durabilidad.
- Amplio rango de temperaturas.

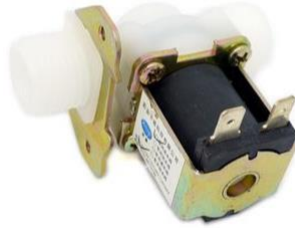


Ilustración 6. Electroválvula.

3.1.5 Multiplexor

Un multiplexor es un circuito combinatorial que permite que, teniendo varias entradas, podamos a través de un selector transmitir en la salida el dato que queramos. Hay distintos tipos de multiplexores de 2, 4, 8, 16...entradas, de paralelo a serie y sus aplicaciones son muy variadas. Se utilizará este multiplexor 8:1 (74HC4051) para leer las salidas analógicas de los sensores de agua.



Ilustración 7. Multiplexor 74HC4051.

3.1.6 Relé

El principal funcionamiento de un relé es el de activar, poner en marcha o encender algo mediante una señal eléctrica con una intensidad mucho menor que la intensidad que va a consumir el aparato o receptor que queremos encender o poner en marcha. Una característica importante de los relés es que la parte que emite la señal para activarlo está aislada de la parte del relé que pone en marcha o enciende el receptor.

Utilizaremos dos relés para la activación de las dos electroválvulas.



Ilustración 8. Relé T010010.

3.1.7 Módulo wifi ESP8266

Con el fin de transferir los datos procedentes del Arduino, se utilizará el dispositivo ESP8266 que es un chip integrado con conexión Wifi y compatible con el protocolo TCP/IP. El objetivo principal es dar acceso a cualquier microcontrolador a una red. El voltaje de trabajo oscila entre 3V y 3,6V y su consumo dependerá de diferentes factores como el modo en el que esté trabajando, de los protocolos que estemos utilizando, o de la calidad de la señal Wifi.



Ilustración 9. Modulo ESP8266.

3.1.8 Fuente alimentación

Para la alimentación de todos los dispositivos del proyecto, se necesitará una fuente de alimentación AC/DC (230V/12V) para las electroválvulas, y otra fuente de alimentación DC/DC (12V/5V) para alimentar el Arduino y el dispositivo ESP8266. Para evitar la compra de ambas fuentes de alimentación y evitar costes, se utilizará la fuente de alimentación de un PC proporcionada por el laboratorio de la UPV, ya que se puede utilizar las alimentaciones de 3.3, 5 y 12 voltios.

3.1.9 Bomba de agua, depósito y material de laboratorio diverso

Con el fin de abastecer el sistema de riego del huerto, utilizaremos dos depósitos, uno con mayor capacidad para el agua y otro de tamaño más reducido para el abono. Para extraer el agua del depósito y poder bombearla por todo el circuito cerrado del huerto utilizaremos la bomba de agua actual.

Además, se utilizará material diverso proporcionado por UPV como:

- Cables
- Tuberías de goma
- Conectores
- Placa perforada
- Soldador

3.2 Software

El lenguaje de programación de Arduino está basado en C++, lenguaje de programación orientado a objetos cuya base es el lenguaje C. C++ fue un lenguaje diseñado a mediados de los años 80 por Bjarne Stroustrup cuyo propósito fue extender el lenguaje de programación C con herramientas que permiten la manipulación de objetos. Se suele decir que C++ es un lenguaje híbrido o multi programa ya que se juntaron facilidades de programación como programación estructurada y la programación orientada a objetos.

Actualmente, C++ es un lenguaje versátil, potente y general. Su éxito entre los programadores profesionales le ha llevado a ocupar el primer puesto como herramienta de desarrollo de aplicaciones. El lenguaje C++ mantiene las ventajas del C en cuanto a riqueza de operaciones, expresiones, flexibilidad, concisión y eficiencia. Además, ha eliminado algunas de las dificultades y limitaciones del C original. La evolución de C++ ha continuado con la aparición de Java, un lenguaje creado simplificando algunas cosas de C++ y añadiendo otras, que se utiliza para realizar aplicaciones en Internet.

Con Arduino, podemos hacer uso de las librerías, que son trozos de código predeterminado que usamos en nuestro sketch. Esto nos facilita mucho la programación y hace que nuestro programa sea más sencillo de hacer y luego de entender. Las librerías son colecciones de código que facilitan la interconexión de sensores, pantallas, módulos electrónicos, etc.

La estructura básica de programación de Arduino es bastante simple y divide la ejecución en dos partes: setup y loop. Setup() constituye la preparación del programa y loop() es la ejecución. En la función Setup() se incluye la declaración de variables y se trata de la primera función que se ejecuta en el programa. Esta función se ejecuta una

única vez y es empleada para configurar el pinMode e inicializar la comunicación serie. La función loop() incluye el código a ser ejecutado continuamente (leyendo las entradas de la placa, salidas, etc.). En los distintos bloques de código cada instrucción acaba con ";" y los comentarios se indican con "//". Al igual que en C se pueden introducir bloques de comentarios con /* ... */.

Entre las distintas funciones de Arduino, algunas de las que vamos a utilizar son las siguientes:

pinMode(): configura un pin específico como entrada o salida. Sintaxis, pinMode(pin, mode)

digitalRead(): Lee el valor de un pin digital específico. Sintaxis, digitalRead(pin)

digitalWrite(): Escribe el valor HIGH o LOW en un pin digital

analogRead(): Lee el valor desde el pin analógico especificado. Esta función solo funciona en los pines analógicos, analogRead(pin).

analogWrite(,): Escribe un valor pseudo-analógico usando modulación por ancho de pulso (PWM) en un pin de salida, analogWrite(pin, value).

If: El comando "if" comprueba una condición y la ejecuta siempre y cuando esta condición será verdadera (true).

```
if (condición) {  
//acciones a ejecutar  
}
```

Capítulo 4. Diseño del proyecto

Una vez definidas las especificaciones del proyecto y los materiales que se van a utilizar, en este apartado del proyecto se explicará el diseño tanto software como hardware de este.

4.1 Diseño hardware

En primer lugar, se va a especificar la función de cada uno de los dispositivos utilizados, los sensores, la conexión de cada uno de ellos y se definirán las conexiones de los pines que se van a utilizar.

4.1.1 Sensores de Agua

Los siete sensores de agua irán colocados en la parte superior de cada una de las tiras verticales del huerto. Dado que los sensores de agua utilizados envían la señal de manera analógica y no tenemos suficientes pines analógicos en la placa de Arduino, se decide utilizar un multiplexor 8:1 para leer la salida de los sensores. De este modo, se utilizarán tan solo tres pines digitales. Estos siete sensores estarán conectados al multiplexor (entradas Y0 hasta Y6 del multiplexor), y a su vez, este estará conectado al Arduino utilizando tres pines digitales y uno analógico que nos permitirá leer el valor de los sensores de agua. Mediante las salidas digitales del Arduino D2, D3 y D4 se controlará cual es el sensor de agua del que se quiere obtener su valor. Estas salidas digitales se conectarán a las entradas S0, S1 y S2 del multiplexor y el valor obtenido de cada uno de los sensores será enviado del pin SIG del multiplexor a la entrada A0 del Arduino. Una vez obtenido el valor de los sensores, se comparará con un valor preestablecido que permita asegurar que la cantidad de agua detectada por el sensor es correcta, garantizando que las plantas no se queden sin agua. Por último, el pin Gnd del multiplexor estará conectado a masa y Vcc al pin de 5 voltios de la placa. En la siguiente figura se puede ver la conexión de los pines:

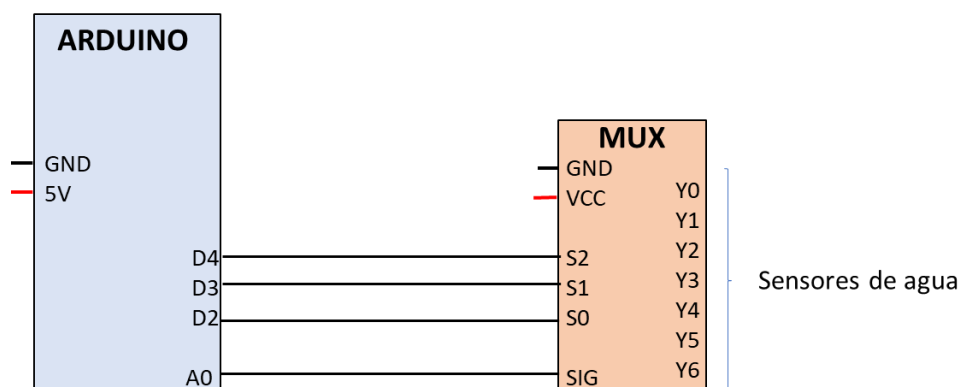


Ilustración 10. Conexión de los sensores de agua con el multiplexor y el Arduino.

Así pues, cada uno de los siete sensores de agua tendrá un pin conectado a Vcc, otro a Gnd y, por último, el pin "S" irá conectado a las entradas del multiplexor:



Ilustración 11. Pines del sensor de agua.

Para controlar que todos los sensores funcionan de manera óptima, se programará el sistema para que envíe correos automáticos cada cierto tiempo tal y como se ha detallado en el capítulo dos, por lo tanto, en caso de que algún sensor detectase un valor de agua inferior al valor establecido, cuando se reciba el correo con toda la información de todos los sensores se podrá leer el siguiente mensaje:

Sensor de agua X: Sin agua.

Por el contrario, si el valor detectado por el sensor es el adecuado aparecerá como:

Sensor de agua X: OK.

4.1.2 Depósitos de agua y abono

Se procede ahora a definir todos los pines de los sensores de nivel y de las electroválvulas, el funcionamiento de estas y los sensores y pines que se utilizarán para controlarlas.

Como se ha mencionado anteriormente, se utilizarán dos depósitos: uno para el agua y otro para el abono líquido. Para controlar el nivel de líquido de cada uno de ellos, se utilizarán tres sensores de nivel, dos para el depósito de agua (SN1 y SN2) y uno para el nivel de abono (SN3) como se puede apreciar en la siguiente figura:

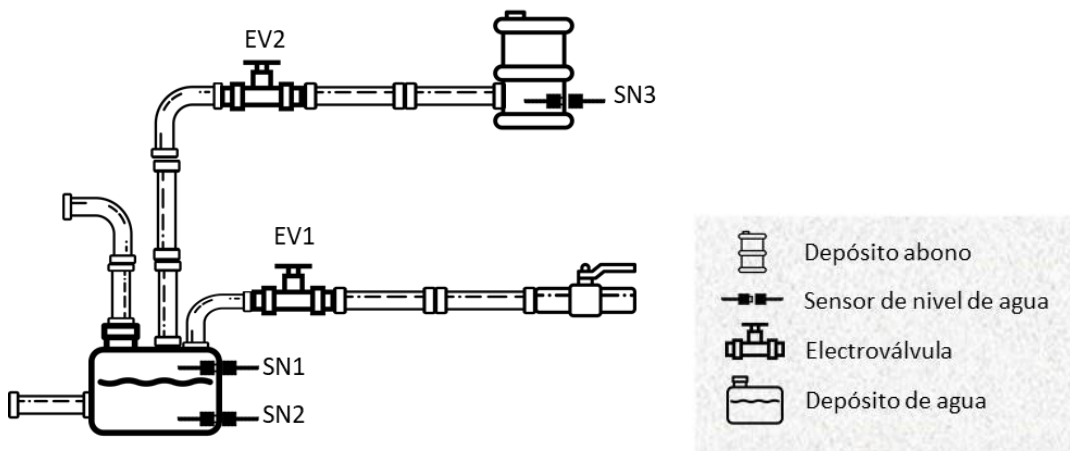


Ilustración 12. Depósitos de agua y abono.

Se incorporarán dos detectores de nivel en el depósito de agua tal y como se puede ver en la imagen anterior. Uno de ellos, colocado en la parte inferior del mismo (SN2), detectará si el nivel del agua es bajo y activará el pin que controla la electroválvula (EV1) para proceder con el llenado del depósito. Otro detector de nivel (SN1) colocado en la parte superior del depósito indicará que el agua a llegado a su nivel máximo y por tanto, apagará la electroválvula (EV1) y finalizará así el proceso de llenado de agua.

Una vez finalizado el proceso de llenado de agua en el depósito principal, se procede a verter la cantidad de abono líquido correspondiente a la capacidad de agua del depósito. Para ello, se activará el pin 12 del Arduino para que se accione la electroválvula (EV2) que controla el flujo de abono líquido para proceder con el llenado de abono. Esta electroválvula se mantendrá activada durante un cierto tiempo predefinido para verter la cantidad necesaria de abono líquido. El vertido de abono al depósito de agua se realizará de manera automática después del llenado de agua.

Para evitar que el depósito de abono no se quede sin fertilizante líquido, añadiremos un tercer sensor de nivel (SN3) conectado al pin 8 del Arduino que se activará cuando detecte que el nivel de abono es bajo.

Cuando el nivel de abono sea bajo y haya que rellenar el depósito de manera manual, se recibirá un correo alertando de la falta de abono y se podrá leer el siguiente mensaje:

Nivel de abono: Rellenar.

En este esquema se muestran los pines utilizados para los sensores de nivel y para las electroválvulas. En definitiva, se muestran todas las entradas y salidas para controlar tanto el depósito de agua como de abono:

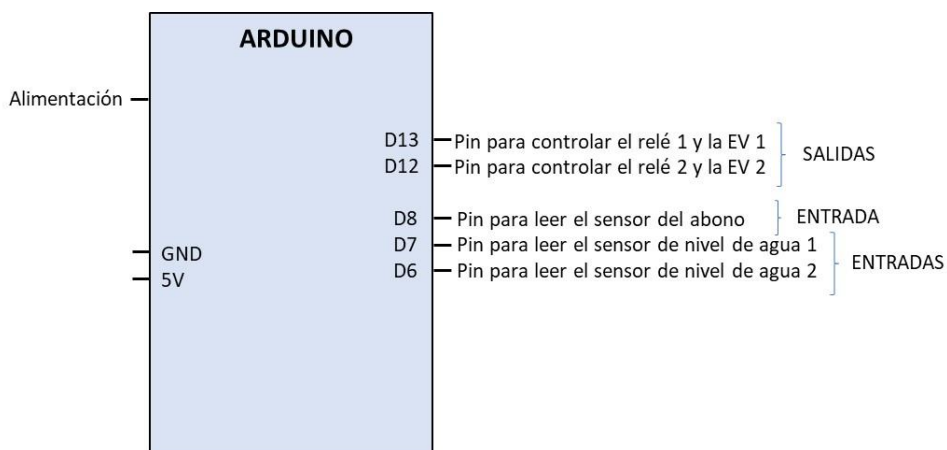


Ilustración 13. Entradas y salidas de control de ambos depósitos.

Como se ha visto, para leer la información de los sensores de agua sobre el estado del depósito de agua utilizaremos los siguientes pines:

- **Pin 6:** Detectará cuando el nivel de agua es bajo para proceder con el llenado del depósito.
- **Pin 7:** Detectará cuando el nivel de agua a llegado a su máximo para finalizar el proceso de llenado.
- **Pin 8:** Detectará cuando el nivel de abono es bajo y alertará para proceder con su llenado.

Así pues, como también se puede apreciar en la imagen 13, para poder activar las electroválvulas desde Arduino utilizaremos dos relés que estarán controlados por los pines 12 y 13:

- **Pin12:** Activara el relé 2 que controla la EV2 para verter el abono liquido al depósito de agua.
- **Pin 13:** Activará el relé 1 que controla la EV1 para comenzar con el proceso de llenado del depósito de agua.

En la siguiente imagen adjunta, podemos ver todos los pines que se conectarán al Arduino, tanto entradas como salidas con el fin de controlar los distintos dispositivos:

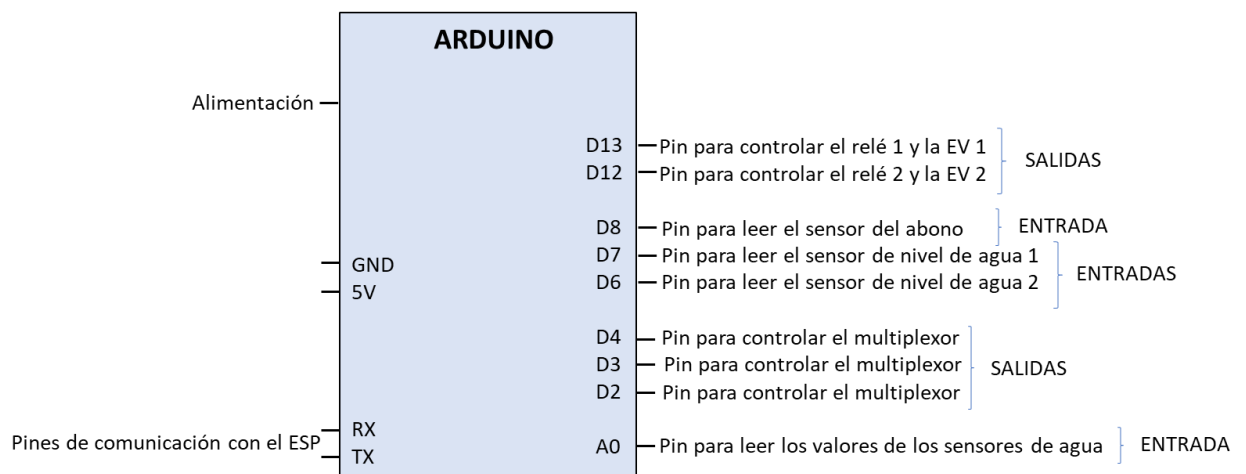


Ilustración 14. Conexión de las Entradas y salidas totales del Arduino.

4.2 Diseño software

Al igual que en el apartado anterior, antes de realizar la implementación del sistema software se procede al diseño del algoritmo, que será la secuencia o el conjunto de instrucciones que se realizarán en un orden concreto para obtener la información deseada y dar solución al problema que se plantea.

4.2.1 Algoritmo del proceso de llenado

Para el proceso de llenado se tendrá en cuenta tanto los sensores de agua del interior del depósito como los relés para la activación de las electroválvulas. Las instrucciones del proceso serían las siguientes:

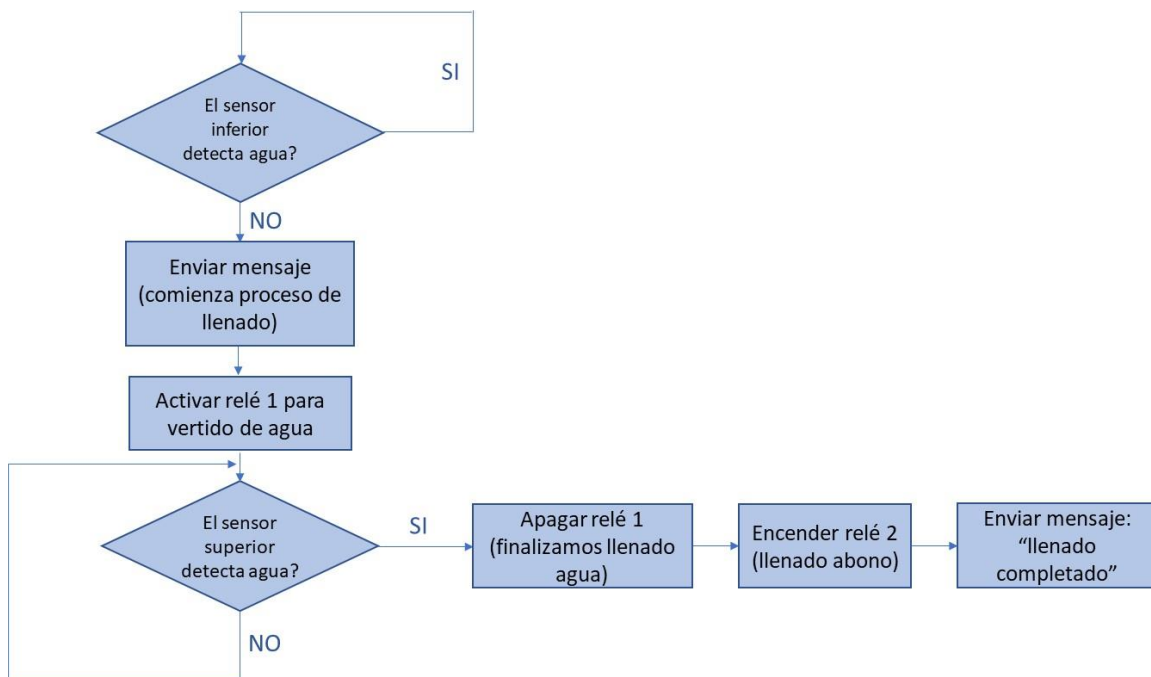


Ilustración 15. Algoritmo del proceso de llenado de agua.

4.2.2 Algoritmo de verificación de los detectores de agua

Para verificar que los siete sensores ubicados en la parte superior de las tiras verticales detectan agua, se creará un array donde se almacenarán el estado de estos sensores. Así pues, se pasará el valor de cada uno de los sensores por un comparador para comprobar si se detecta agua o no. El algoritmo será el siguiente:

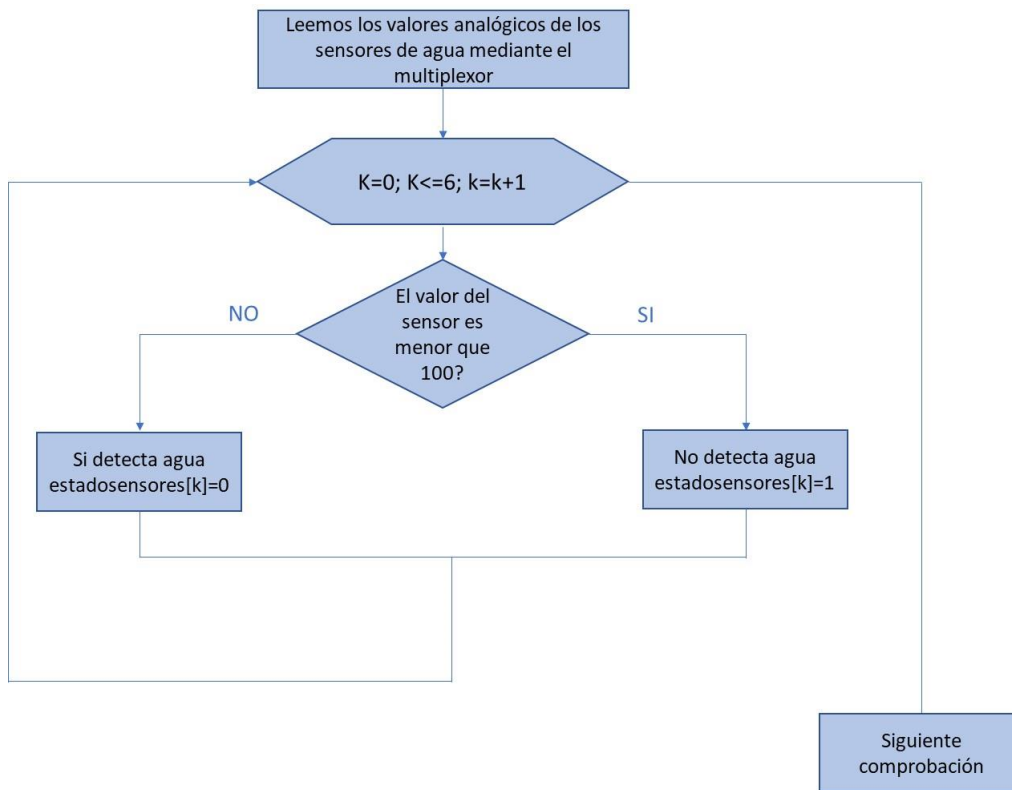


Ilustración 16. Algoritmo de comprobación de los sensores de agua.

4.2.3 Algoritmo para comprobar depósito de abono

Para comprobar si el depósito de abono está lleno o no, solo hará falta saber si el sensor de nivel del depósito detecta abono líquido. Para ello, se estará preguntando todo el tiempo por el estado del sensor:

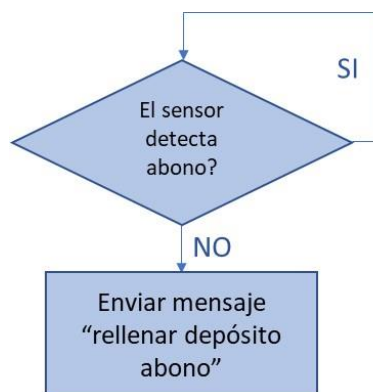


Ilustración 17. Algoritmo de comprobación del depósito de abono.

Capítulo 5. Implementación del sistema

5.1 Implementación Software

En este apartado, se procede al desarrollo software del proyecto, explicando la implementación del código.

Para que el código sea más sencillo a la hora de interpretarlo, se han creado cuatro módulos principales con dos funciones cada uno para el *setup* y para *el loop*:

En el primer módulo está implementado todo aquello referente a los sensores de agua y el multiplexor que se va a utilizar para leer los valores de cada uno de ellos.

Setup_Multiplexor ();

Comprobar_Sensores_Multiplexor ();

El segundo módulo controla las electroválvulas y los diferentes estados del depósito.

Setup_Electrovalvulas ();

Comprobar_Deposito ();

En el tercer módulo se controla el sensor y el depósito de abono.

Setup_Sensor_Abono();

Comprobar_Sensor_Abono ();

Por último, el cuarto módulo nos permitirá enviar el correo con la información de cada uno de los sensores.

Enviar_Correo ();

Para la recogida de los datos proporcionados por los sensores, se ha creado un array (*estadosensores [10]*) en el que se volcará el estado de cada uno de los sensores. Este array será enviado del dispositivo Arduino al ESP8266, el cual analizará la información y será el encargado de enviar el correo, vía Wifi, para mostrar si hay algún fallo en alguno de dichos sensores.

Array *estadosensores[10]*:

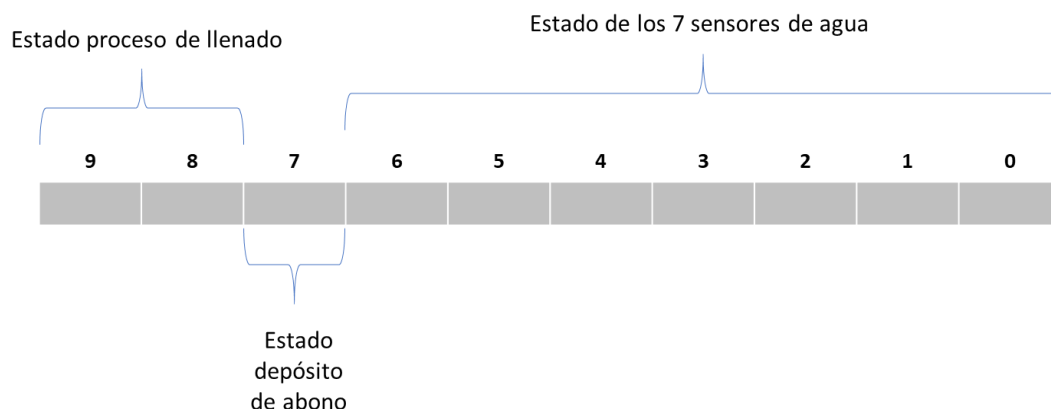


Ilustración 18. Array *estadosensores[10]*.

5.1.1 Módulo 1: Sensores de agua y multiplexor.

Los 7 sensores de agua estarán conectados al multiplexor mediante los pines YX, y este a su vez estará conectado al Arduino tal y cómo muestra en la siguiente figura:

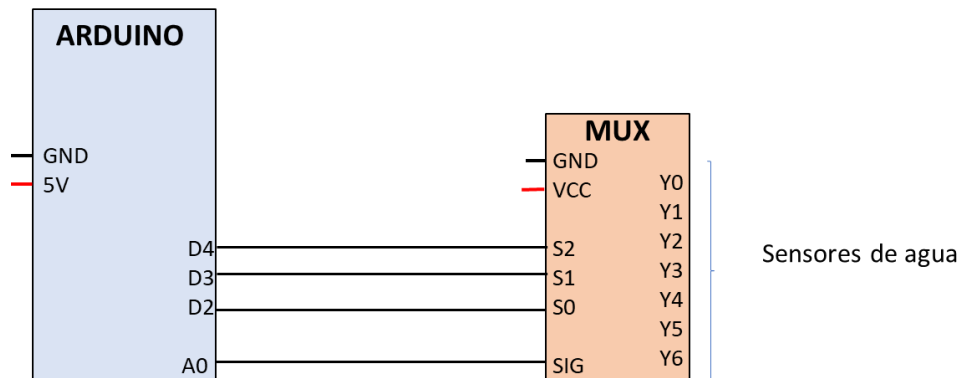


Ilustración 19. Conexiones del Arduino con el multiplexor.

Por una parte, los pines digitales (2, 3 y 4) del Arduino estarán conectados a los pines S0, S1 y S2 del multiplexor (muxS0, muxS1 y muxS2 respectivamente) y mediante la función `SetMuxChannel()` seleccionaremos cual es el canal (YX) que queremos leer.

| Sensor (YX) | S0 | S1 | S2 |
|-------------|----|----|----|
| Y0 | 0 | 0 | 0 |
| Y1 | 0 | 0 | 1 |
| Y2 | 0 | 1 | 0 |
| Y3 | 0 | 1 | 1 |
| Y4 | 1 | 0 | 0 |
| Y5 | 1 | 0 | 1 |
| Y6 | 1 | 1 | 0 |

```
void SetMuxChannel(byte channel)
{
  // Selección de la entrada de control del multiplexor.

  digitalWrite(muxS0, bitRead(channel, 0));
  digitalWrite(muxS1, bitRead(channel, 1));
  digitalWrite(muxS2, bitRead(channel, 2));
}
```

Ilustración 20. Pines multiplexor.

Este valor de cada uno de los sensores de agua será leído por el pin analógico del Arduino A0 (muxSIG). Si el valor obtenido es mayor que 100, el nivel de agua detectado por el sensor será el adecuado y se asignará un 0 a la casilla correspondiente a dicho sensor en el array (`estadosensores [10]`). En caso de que el valor obtenido sea inferior a 100, se asignará un 1 a la casilla mencionada. Este 1 indicará que hay algún problema de detección de agua en ese sensor.

```

for (byte i = 0; i <=6; i++)
{
  SetMuxChannel(i);
  byte muxValue = analogRead(muxSIG);
  if (muxValue<100)
  {
    estadosensores [i]=1;
  }
  else
  {
    estadosensores [i]=0;
  }
}

```

5.1.2 Módulo 2: Electroválvulas y sensores de nivel

Pines asociados

Como hemos visto en el capítulo anterior, los pines asociados a los sensores de nivel del depósito de agua son el 6 y 7, los asociados a las electroválvulas serán los pines 12 y 13.

```

// Pines asociados a las electroválvulas 1 y 2.
const int pinEV1=13;
const int pinEV2=12;

// Pines asociados a los sensores de nivel.
const int sensorNivel=7;
const int sensorNivel2=6;

```

Entradas y salidas

En el *setup*, estableceremos los pines 6 y 7 como entradas ya que aportarán información del estado del depósito; por el contrario, los pines 12 y 13 serán las salidas que utilizaremos para la activación de las electroválvulas a través de los relés:

```

// Modo de los pines asociados a las
electroválvulas.
pinMode (pinEV1,OUTPUT);
pinMode (pinEV2, OUTPUT);

// Modo de los pines asociados a los
sensores de nivel.
pinMode (sensorNivel, INPUT);
pinMode (sensorNivel2, INPUT);

```

Estado inicial de las electroválvulas

En primer lugar, asignamos un nivel bajo a los pines que controlan las electroválvulas para que estén apagadas puesto que se partirá con los dos depósitos llenos, tanto el de agua como el de abono:

```
// Apagado de las electroválvulas.
digitalWrite (pinEV1, LOW);
digitalWrite (pinEV2, LOW);
```

En “*Comprobar_Deposito()*” se preguntará por el estado del sensor de nivel 2, que es el sensor inferior del depósito de agua. Cuando este sensor detecte que el nivel de agua del depósito es bajo, cambiará el estado de la posición 8 del array “*estadosensores[]*” y se enviará un correo informando de que el proceso de llenado del depósito de agua ha comenzado:

```
// Lectura del sensor de nivel inferior.
valorSN2=digitalRead(sensorNivel2);

// Si el sensor infeior del depósito detecta que no hay
agua,
// su salida será 1 y empieza el proceso de llenado.
if ((valorSN2==HIGH))
{
    estadosensores [8]= 1;

    // Enviar correo de comienzo del proceso de llenado.
    Enviar_Correo ();
```

Una vez el proceso de llenado haya comenzado, se prestará atención al sensor superior del depósito y mientras este sea 0 (es decir, que no detecta agua) el proceso de llenado estará en marcha. Tan pronto como este sensor cambie a nivel alto, significará que el nivel de agua ha llegado hasta el sensor superior del depósito y se pondrá el pin de la electroválvula a nivel bajo:

```
while (valorSN==0)
{
    digitalWrite (pinEV1, HIGH);
    valorSN=digitalRead(sensorNivel);
}

// Apagado de la EV1.
digitalWrite (pinEV1, LOW);
```

Seguidamente, tal y como se ha explicado en el capítulo 2, se activará el pin (12), durante un cierto tiempo, que controla el relé y la electroválvula dos para proceder con el vertido de abono líquido en el depósito de agua:

```
digitalWrite (pinEV2,HIGH);
delay (5000);
digitalWrite (pinEV2,LOW);
```

Al comienzo del proceso de llenado se cambiará el estado de la casilla nº 8 del array para que se envíe un correo para informar del comienzo del proceso de llenado. Una vez el depósito de agua está completamente lleno y se ha vertido la cantidad de abono necesaria, esta casilla del array pasará a estar a cero como que el proceso de llenado no está en marcha. Por otro lado, para informar que el proceso de llenado se ha realizado con éxito, pondremos la casilla nº 9 del array a 1 y se enviará otro correo para notificar que el proceso ha terminado:

```
estadosensores [8]= 0;
estadosensores [9]= 1;

// Enviar correo de finalización del proceso de llenado.
Enviar_Correo ();
```

5.1.3 Módulo 3: Sensor Abono

Como se ha visto, el sensor del abono servirá para detectar cuando el nivel de abono del depósito es bajo para proceder con el llenado de manera manual, por consiguiente, lo único que se hace es leer la entrada del sensor, en este caso se realizará mediante el pin 8 (INPUT) del Arduino.

En el *setup* declaramos la variable y asignamos el pin:

```
const int sensorAbono=8;

// Valores del sensor de abono.
int valorSN3;

void Setup_Sensor_Abono()
{
// Configuración inicial del sensor de abono.

pinMode (sensorAbono, INPUT);
}
```

Por otro lado, en el *loop* se estará leyendo continuamente esta entrada. Si el valor del sensor de nivel es igual a 0 significará que está detectando líquido, se asignará un 0 a la casilla del array indicando que no hay que hacer nada. En el caso que la entrada del pin 8 sea 1, es decir, el sensor no detecta líquido, añadiremos un 1 a la casilla 7 (estado sensor abono) y cuando mandemos el correo se enviará que hay que rellenar el depósito.

```
void Comprobar_Sensor_Abono()
{
    // Comprueba el estado del sensor del abono.

    // Lectura del sensor de abono.
    valorSN3=digitalRead(sensorAbono);

    // Actualizar el array de estado de los sensores según el valor
    // del sensor de abono.
    if (valorSN3==LOW)// detecta abono
    {
        estadosensores [7]= 0;//por tanto la casilla 7 del array estará a
0 (abono OK)
    }
    else
    {
        estadosensores [7]= 1; //(Si el sensor de abono no detecta agua,
pondremos la casilla del array 7 a 1 (Hace falta llenar el depósito de
abono))
    }
}
```

5.1.4 Módulo 4: Envío del correo

Como comentado anteriormente, habrá dos tipos de mensajes: mensajes de control y mensajes de alerta.

Mensajes de control

Los mensajes de control se enviarán cada 8 horas, es decir, tres mensajes al día y en este correo se enviará el estado de todos los sensores. Así pues, tras realizar todas las comprobaciones se enviará un correo mediante la función *Enviar_Correo ()*:

```
void loop()
{
    // Revisión de los sensores asociados al multiplexor.
    Comprobar_Sensores_Multiplexor ();

    // Revisión y llenado del depósito.
    Comprobar_Deposito ();

    // Revisión del sensor de abono.
    Comprobar_Sensor_Abono ();

    // Enviar correo
    Enviar_Correo ();
}
```

Mensajes de alerta

Los mensajes de alerta surgen si durante las comprobaciones nuestro código detecta que no hay agua en el depósito y se tiene que dar comienzo al llenado del depósito. Así pues, al final dicho proceso se enviará otro mensaje de alerta informando de que el proceso se ha completado.

Envío del correo

Para el envío del correo con la información de los sensores y el estado de los depósitos se ha implementado el código tanto en la placa Arduino como en el ESP8266.

En primer lugar, en el Arduino se ha creado la función “*Enviar_Correo()*”, que se utilizará para recorrer el array que contiene toda la información de todos los sensores y de crear el mensaje mediante la llamada “*String*”.

```
String mensaje= "";

    for (int i= 0; i <= 9; i++)
    {
        if (estadosensores [i] == 1) mensaje= mensaje +
"1";
        else mensaje= mensaje + "0";
    }

    if (estadosensores [9] == 1)
    {
        // Anotar en el correo que el proceso de llenado ha
finalizado
        // y desactivar el sensor 9.

        estadosensores [9]= 0;
    }
```

Como se puede ver en la imagen anterior, antes de enviar el mensaje al ESP8266, se comprueba el estado de la casilla número 9 del array. Como se ha visto, en esta casilla se guarda la información de si el proceso de llenado ha sido completado o no. En el caso que se haya completado, ya se habrá enviado un correo informando de la finalización de dicho proceso; por tanto, en este punto el proceso habrá finalizado y deberemos colocar esta casilla del array a cero.

Una vez se ha conformado el mensaje recorriendo todo el array, se enviará el mensaje a través del pinTX del Arduino al RX del ESP8266.

Por otro lado, en el dispositivo ESP8266 se añadirá el código para leer la información proporcionada por la placa Arduino y generar el mensaje para ser enviado por email.

Conexión del ESP8266

En primer lugar, se procede a la identificación de la red para tener acceso a internet y poder enviar los mensajes. Así pues, asignaremos el servidor de la UPV y el puerto indicado en la siguiente imagen:

```
// REDES WIFI

const char* SSID_CASA = "MXXXXXXXXXXXXX";
const char* PWD_CASA = "XXXXXXXXXXXXXXXXXXXX";

// SERVIDORES SMTP

const char* SERVIDOR_UPV= "smtp.upv.es";
const uint16_t PUERTO_UPV= 25;
```

En segundo lugar, se estará continuamente comprobando si se ha recibido la información del estado de los sensores desde el dispositivo Arduino mediante el siguiente código:

```
// Mensaje recibido desde el módulo arduino.

String mensaje;

void loop()
{
  // Comprobación permanente de recepción de datos del módulo
  arduino.

  if (Serial.available()>0)
  {
    mensaje= Serial.readString ();
    Serial.println (mensaje);
    sendEmail ();
  }

  delay (10000);
}
```

Por último, para la construcción del mensaje se recorrerá todo el array para extraer la información que se ha recibido del Arduino. A continuación, se muestran los mensajes para:

- Los siete sensores de agua:

```
if (mensaje [0] == '0')
{
    espClient.println(F("Sensor de agua 1: OK\n"));
    Serial.println(F("Sensor de agua 1: OK\n"));
}
else
{
    espClient.println(F("Sensor de agua 1: Sin agua\n"));
    Serial.println(F("Sensor de agua 1: Sin agua\n"));
}
```

- El sensor de abono:

```
if (mensaje [7] == '0')
{
    espClient.println(F("Nivel de abono: OK\n"));
    Serial.println(F("Nivel de abono: OK\n"));
}
else
{
    espClient.println(F("Nivel de abono: Rellenar\n"));
    Serial.println(F("Nivel de abono: Rellenar\n"));
}
```

- Vertido de agua:

```
if (mensaje [8] == '1')
{
    espClient.println(F("Vertido de agua: Ongoing\n"));
    Serial.println(F("Vertido de agua: Ongoing\n"));
}
else
{
    espClient.println(F("Vertido de agua: Tanque lleno\n"));
    Serial.println(F("Vertido de agua: Tanque lleno\n"));
}
```


- Sensor del proceso de llenado:

```

if (mensaje [9] == '1')
{
  espClient.println(F("Proceso de llenado: Completado\n"));
  Serial.println(F("Proceso de llenado: Completado\n"));
}
else
{
  espClient.println(F("Proceso de llenado: Tanque lleno\n"));
  Serial.println(F("Proceso de llenado: Tanque lleno\n"));
}

```

5.2 Implementación Hardware

En este apartado, se procede al desarrollo hardware del proyecto, explicando la conexión y el montaje de todos los componentes.

El primer paso para la puesta en marcha del sistema es soldar los dispositivos principales a la placa. En la figura 19 se pueden ver los siguientes dispositivos:

- Placa Arduino: que recogerá el estado de todos los sensores que controlan el huerto. Estos datos serán enviados al ESP8266.
- ESP8266: recibirá los datos del Arduino y generará y enviará el mensaje.
- Dos relés: activarán las dos electroválvulas.
- Multiplexor: se utilizará para entregar la información de los sensores de agua al dispositivo Arduino.

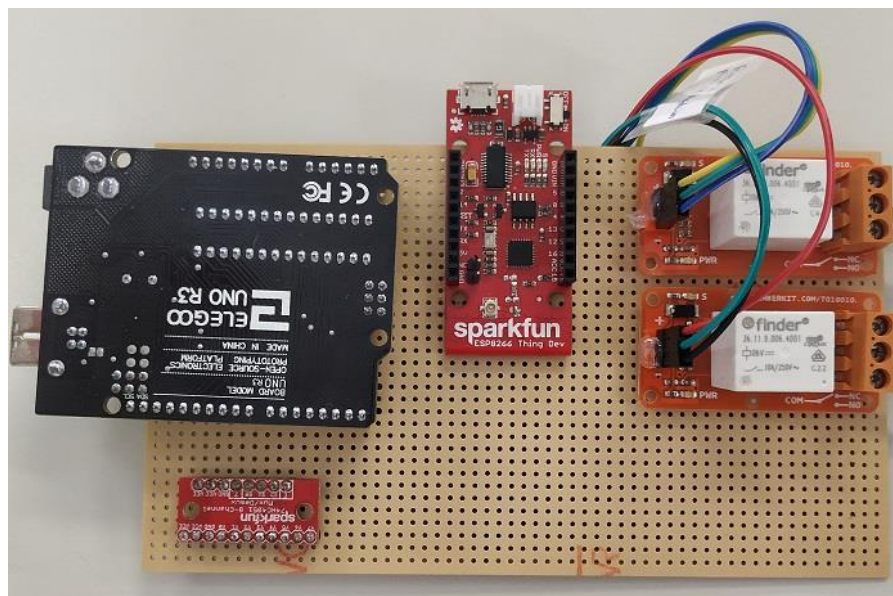


Ilustración 21. Arduino, ESP8266, relés y multiplexor.

Respecto a la fuente de alimentación del sistema completo, se necesitan dos transformadores: un transformador AC/DC (230V/12V) para los dos relés que activarán las electroválvulas y otro transformador DC/DC (12V/5V) para alimentar los dispositivos restantes. Para ello, utilizaremos una fuente de alimentación de un PC que nos proporciona 5V, 12V, 3.3V y GND. De este modo podremos alimentar todo nuestro sistema con una única fuente de alimentación tal y como se puede ver en la figura adjunta:



Ilustración 22. Fuente de alimentación.

5.2.1 Implementación de los sensores de agua

En este punto se explicará el montaje de los 7 sensores de agua que irán ubicados en las tiras verticales del huerto. Para ello, por una parte, se soldará el pin “S” de cada uno de los sensores de agua a las entradas “Y0-Y6” del multiplexor. Por otra parte, se alimentarán todos los sensores con 5V y llevaremos el pin GND de cada uno de ellos a la placa. En la siguiente figura se puede ver el montaje de dichos sensores que se ha realizado para hacer las pruebas de comprobación:

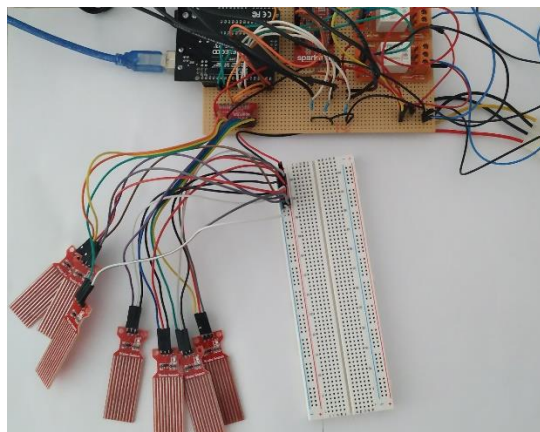


Ilustración 23. Montaje de los sensores de agua.

Como se puede ver, debido a que no se ha podido montar el sistema en el huerto de la UPV, se ha hecho una pequeña simulación para el montaje de estos sensores con la intención de poder verificar su correcto funcionamiento.

5.2.2 Implementación depósito de agua

Como se ha comentado en el capítulo 4, para la implementación del depósito de agua se utilizarán dos sensores de nivel conectados a los pines 6 y 7 del Arduino. Con estos dos sensores, sabremos el estado del depósito de agua y ambos gestionarán el encendido y apagado de los relés que controlan las electroválvulas (EV1 y EV2).

Para el montaje de los detectores de nivel, se conectarán a los pines correspondientes al Arduino (6 y 7) y se alimentarán con 5V. Como se ha visto anteriormente en las especificaciones de los detectores, estos se pueden rotar 180°. EL sensor de arriba del depósito de agua estará la mayor parte de su tiempo inactivo, ya que solo se activará durante el llenado del depósito y tan pronto se consuma agua, este cambiará su posición y dejará de consumir. Por tanto, la posición del sensor superior del depósito será la que se puede ver en la siguiente figura:



Ilustración 24. Posición del sensor superior (SN) del depósito de agua.

Por el contrario, el sensor inferior estará siempre en contacto con el agua y si se coloca en la misma posición que el detector de arriba, estará consumiendo en todo momento. Así pues, rotaremos el sensor 180° para que solo se active en el momento que el nivel de agua sea bajo. La posición de este sensor inferior será la siguiente:



Ilustración 25. Posición del sensor inferior (SN2) del depósito de agua.

En la siguiente imagen podemos ver el montaje de todos los componentes que rigen el funcionamiento del llenado del depósito de agua:

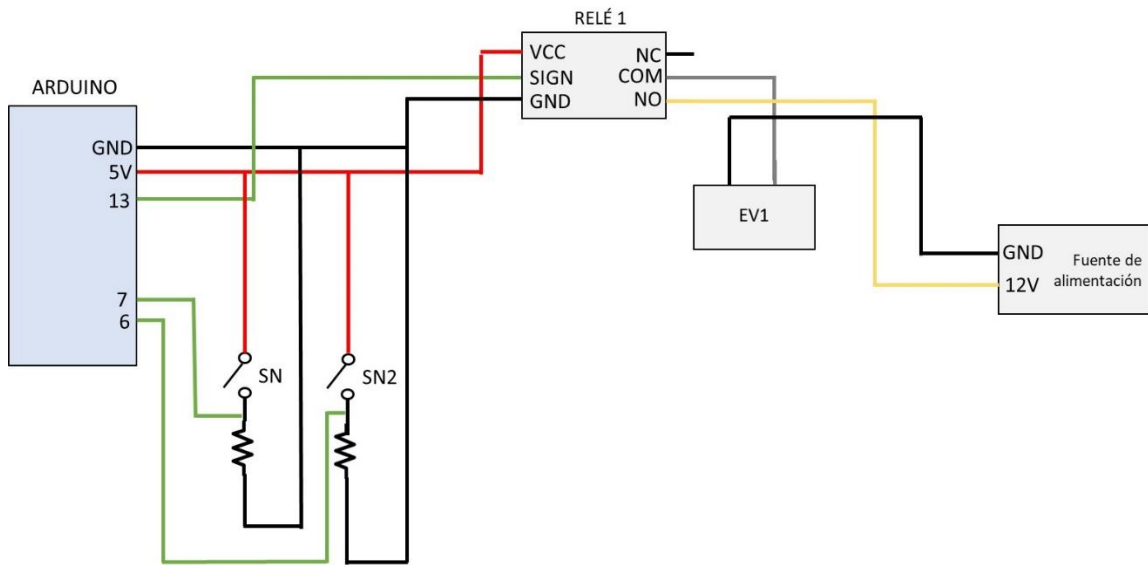


Ilustración 26. Conexiones de los sensores de nivel (SN y SN2) con el relé y EV1.

Para evitar que la electroválvula tenga siempre corriente, conectaremos el pin de 12V de la fuente alimentación al relé y el pin de GND a la EV1.

Como se puede ver, se añadirá una resistencia entre el sensor de nivel y tierra para evitar que el cable quede directamente conectado a GND y pueda dar fallos en la detección:

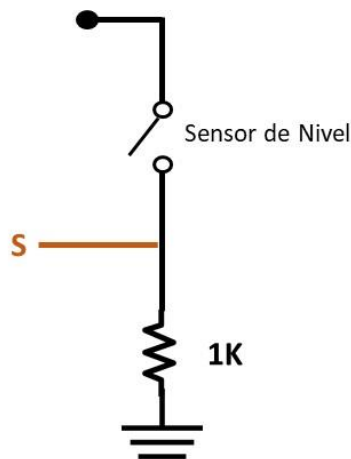


Ilustración 27. Conexión sensor de nivel.

5.2.3 Implementación depósito de abono

Para la implementación de los sensores y dispositivos que corresponden al funcionamiento del depósito de abono será muy similar al visto anteriormente. En este caso, el sensor de nivel irá conectado al pin 8 del Arduino y el pin que controlará el relé para la activación de la electroválvula será el pin 12 como se puede ver en la siguiente figura:

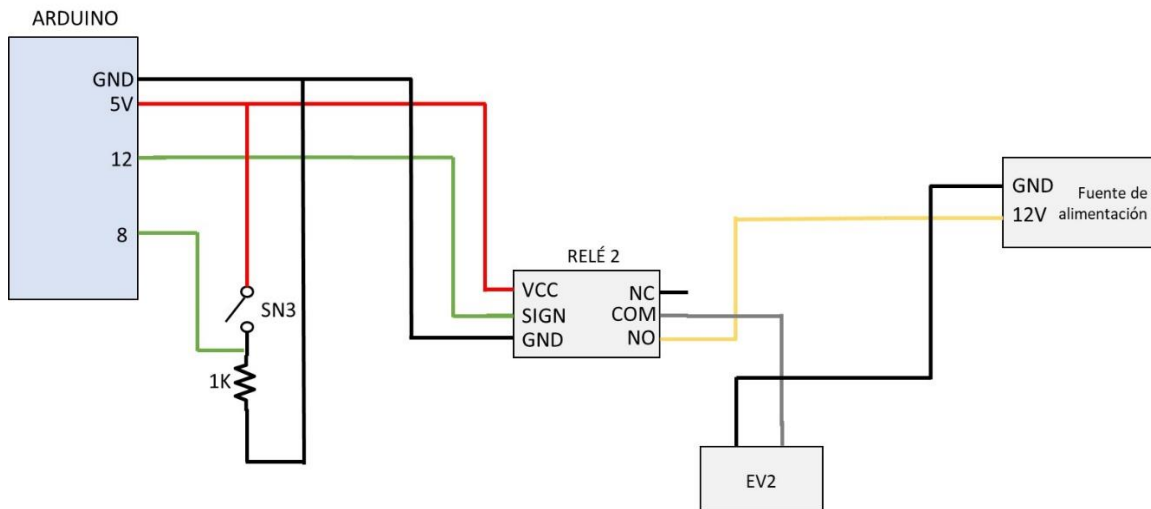


Ilustración 28. Conexión de SN3 con EV2.

En cuanto a la posición del detector de nivel del depósito de abono, se posicionará igual que el sensor inferior del depósito de agua ya que si no lo rotamos 180° estará consumiendo en su mayor parte del tiempo.

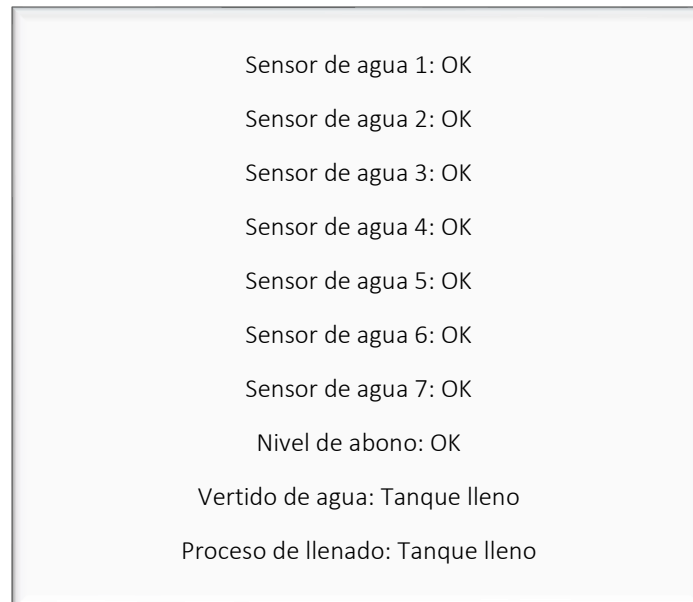


Ilustración 29. Posición del sensor inferior (SN3) del depósito de abono.

Capítulo 6. Pruebas de verificación y comprobación del sistema

Una vez que todo el sistema ha sido montado, se van a realizar una serie de comprobaciones para verificar que todos los sensores y el sistema de correo funciona de manera correcta. Para ello, se simularán fallos en el sistema en el laboratorio y se comprobará cual es la información que se recibe en los correos, de esta manera se evitará que en un futuro puedan aparecer distintos modos de fallo y que no puedan ser detectados. Para la realización del testeado del sistema, este estará montado en el laboratorio y se producirán de manera manual.

Este es el cuerpo del mensaje completo que se ha recibido al encender el sistema estando los depósitos de agua y abono llenos y todos los sensores en contacto con el agua.



```
Sensor de agua 1: OK
Sensor de agua 2: OK
Sensor de agua 3: OK
Sensor de agua 4: OK
Sensor de agua 5: OK
Sensor de agua 6: OK
Sensor de agua 7: OK
Nivel de abono: OK
Vertido de agua: Tanque lleno
Proceso de llenado: Tanque lleno
```

Así pues, se procede al testeado de todos los sensores y para ello se dividirá este apartado en los siguientes módulos:

- Verificación de los sensores de agua.
- Verificación de los sensores del depósito de abono.
- Verificación de los sensores del depósito de agua.

6.1 Verificación de los sensores de agua

Las 7 primeras líneas del mensaje informarán del estado de los 7 sensores de agua, indicando si se detecta agua o no. Al inicio del encendido del sistema y con todos los sensores en contacto con el agua se recibe el siguiente mensaje:

```
Sensor de agua 1: OK
Sensor de agua 2: OK
Sensor de agua 3: OK
Sensor de agua 4: OK
Sensor de agua 5: OK
Sensor de agua 6: OK
Sensor de agua 7: OK
```

Para comprobar que los sensores de agua funcionan correctamente, en primer lugar, retiraremos del agua y secaremos el sensor número 1 para comprobar que en el próximo correo que se reciba se puede ver que el sensor número 1 no detecta agua "Sin agua":

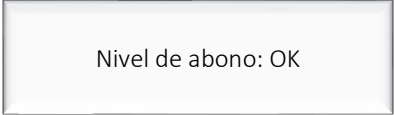
```
Sensor de agua 1: Sin agua
Sensor de agua 2: OK
Sensor de agua 3: OK
Sensor de agua 4: OK
Sensor de agua 5: OK
Sensor de agua 6: OK
Sensor de agua 7: OK
```

Una vez recibido el correo anterior, se realizará el mismo procedimiento para cada uno de los seis sensores restantes. De este modo y una vez que se ha comprobado a través del correo que cada uno de los sensores funcionan correctamente, si se recibe algún correo con alguno de los sensores "Sin agua" se podrá detectar los modos de fallo explicados en el capítulo 2.

6.2 Verificación de los sensores del depósito de abono

Después de la comprobación de los 7 sensores de agua, la siguiente línea de información, corresponde al estado del depósito de abono. Para comprobar si el detector de pico de pato (sensor de nivel) funciona de manera adecuada en el laboratorio, se realizará de manera manual y sin agua, ya que los dos modos de detección del sensor se pueden reproducir de manera manual cambiando la posición de la parte móvil del mismo.

Así pues, se observa que en el correo recibido (en el arranque del sistema y estando el depósito de abono lleno) el sensor funciona correctamente.



Nivel de abono: OK

Acto seguido, cambiamos la posición del sensor simulando que no queda abono líquido en el depósito y se comprueba que en el siguiente mensaje recibido ha cambiado el estado del sensor indicando que hay que rellenar de manera manual el depósito de abono.

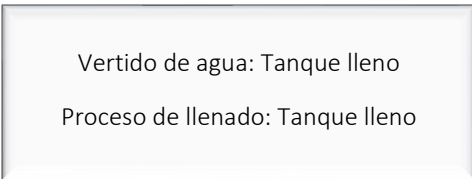


Nivel de abono: Rellenar

6.3 Verificación de los sensores del depósito de agua

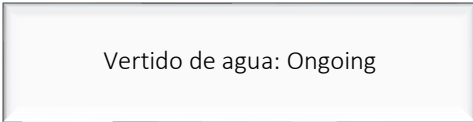
Por último, las dos líneas finales del mensaje enviado informarán del estado del depósito y del estado del proceso de llenado en caso de que este esté en marcha. El procedimiento para la comprobación del sistema será el mismo que se ha seguido para verificar la detección de abono ya que el funcionamiento de los sensores es el mismo.

Como se ha comentado en el capítulo 2, el primer mensaje indicará si el depósito de agua está lleno, o si por el contrario el sensor ha detectado que el nivel de agua es bajo y el proceso de llenado del depósito está en marcha. Se colocarán los detectores de agua simulando que el tanque está completamente lleno, y el correo que se recibe muestra que la detección es correcta.



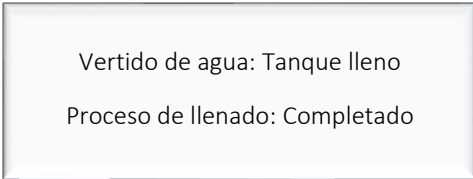
Vertido de agua: Tanque lleno
Proceso de llenado: Tanque lleno

Acto seguido, se cambia la posición del sensor superior, simulando así que el agua se está consumiendo, pero no se recibe ningún correo de alerta hasta que se cambia de posición el sensor inferior, indicando pues que el nivel de agua es bajo. Al cambiar el estado del detector inferior comienza el proceso de llenado. Se comprueba que la electroválvula 1 se ha encendido y se comprueba que se ha recibido un correo comunicando que el proceso de llenado está en marcha tal y como muestra el cuerpo del siguiente mensaje:



Vertido de agua: Ongoing

Por último, se recolocará el sensor inferior a su posición inicial simulando que el depósito se está llenando. Cuando se coloca el sensor superior del depósito en su posición inicial (horizontalmente, simulando que el nivel de agua a llegado a su máximo), se observa que la EV1 se desactiva. Acto seguido, se comprueba que la EV2 se activa durante el periodo de tiempo establecido, que posteriormente se recibe el siguiente mensaje indicando que el tanque de agua está lleno y que el proceso de llenado ha finalizado con éxito:



Vertido de agua: Tanque lleno
Proceso de llenado: Completado

Capítulo 7. Conclusiones y futuras líneas de trabajo

En el capítulo 1 del proyecto, se ha hablado del propósito del proyecto y se han presentado cuales eran los objetivos y las metas que se querían conseguir explicando la metodología que se iba a seguir. El propósito de este último capítulo es valorar las conclusiones a las que se han llegado, comprobar que el proyecto ha sido desarrollado según lo previsto y comprobar si se ha cumplido con los objetivos establecidos. También se expondrán una serie de ideas como futuras líneas de trabajo para poder mejorar este proyecto y hacer que la información del huerto llegue de manera más cómoda al usuario final.

En primer lugar, hay que comentar que los resultados del proyecto muestran que se puede tener un huerto autosuficiente y gestionado electrónicamente de manera sencilla, eficaz y a un coste relativamente bajo. En segundo lugar, hay que añadir que después de concebir la idea general del proyecto, el diseño y realizar la implementación tanto software y hardware, el desarrollo del proyecto se ha llevado a cabo de manera satisfactoria siguiendo los pasos y cumpliendo con las premisas establecidas en el capítulo 1.

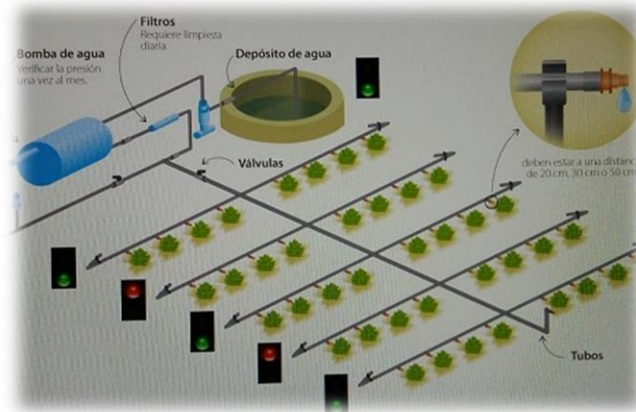
Como conclusión, se puede decir que el diseño de todo el sistema y la implementación de la placa proyecto se ha realizado con éxito y se ha desarrollado según lo previsto. Tras la realización de las pruebas de verificación del sistema completo en el laboratorio, se puede afirmar que se ha cumplido con los objetivos fijados. Cabe destacar que no se ha podido implementar el proyecto en huerto de la universidad debido a la pandemia del Covid-19, que ha impedido el acceso a la universidad durante los meses de la realización de este.

En cuanto a las futuras líneas de trabajo, el objetivo del proyecto era desarrollar un sistema autosuficiente y evitar al máximo la intervención de la mano del hombre para que el huerto funcione de manera correcta y eficaz. Al igual que en cualquier aplicación móvil o página web siempre se necesitan modificaciones para actualizar el sistema y hacerlo más visible, accesible al usuario, así como incrementar su comodidad y sus prestaciones. A lo largo del trabajo y debido a ciertas dificultades y contratiempos, este proyecto ha sufrido varios cambios tanto en el software como en el hardware, aun así, las futuras variaciones que puede sufrir el proyecto pueden ser varias.

A continuación, se expondrán algunas de las posibles mejoras o ampliaciones que se podrían realizar en un futuro próximo para renovar y actualizar el huerto:

1. Una de las prioridades de hoy en día, es hacer que cualquier sistema sea lo más eficiente posible de cara al medio ambiente. Por eso se propone realizar un estudio del consumo de agua de las plantas a largo plazo, para así comprobar si es necesario que en estaciones como el invierno las plantas necesitan que la bomba de agua esté funcionando las 24 horas del día. De este modo, si se comprobase que es las estaciones más húmedas las plantas no precisan que el agua circule por la noche, se podría ahorrar tanto agua como luz, siendo así más respetuoso con el medio ambiente.
2. También se podría realizar un estudio de la humedad relativa media de la zona donde se encuentra el huerto y añadir sensores de humedad en las plantas para un cuidado óptimo de las mismas. Estos sensores serían controlados por el Arduino al igual que los sensores actuales y para regular la humedad se podrían utilizar por ejemplo pequeños aspersores de agua.

3. Hasta ahora, el sistema envía un email al correo con toda la información del estado de los sensores, pero sería interesante poder verificar el estado de los sensores, del depósito y el estado de las plantas mediante una pequeña página web para que sea más visual. A modo de ejemplo, podría ser algo parecido a la siguiente imagen:



4. Otra de las modificaciones más interesantes que podría sufrir este proyecto es la creación de una aplicación móvil para no tener que estar constantemente revisando el correo o visitando la página web para obtener la información deseada. Además, un cambio sugerente sería poder controlar dispositivos desde la aplicación móvil como por ejemplo la bomba de agua.
5. Como se ha detallado anteriormente, se ha utilizado una placa perforada para el montaje de los distintos dispositivos soldando los cables para establecer las conexiones. Para cambiar esta placa perforada, se podría realizar una placa impresa (PCB) para, por ejemplo, evitar errores de cableado o reducir el tiempo de montaje. También se minimizaría así el espacio ocupado por la actual placa.
6. Para la detección de agua en las tiras verticales del huerto se han usado los sensores de agua por su fácil utilización y su bajo coste. Se debería comprobar la fiabilidad/durabilidad de estos a largo plazo al igual que la de todos los otros sensores utilizados y en caso de no satisfacer las necesidades del proyecto, se optaría por reemplazarlos por otros sensores más perdurables.
7. A lo largo del proyecto, se ha explicado que el sistema manda un correo con el estado de todos los sensores del sistema utilizando el dispositivo ESP8266. En un primer momento, no se iba a mandar este correo para comprobar dicha información porque iba a hacerse mediante un display y por tanto solo se iba a utilizar el Arduino. Después de la modificación, se introdujo el ESP8266 al proyecto, pero sin suprimir el Arduino, ya que todo el diseño del proyecto se había realizado solo con Arduino. Como futura modificación del proyecto, se podrían añadir pines al ESP8266 y unificar todo el código, de manera que se pueda retirar el Arduino del proyecto y que el ESP8266 u otro dispositivo que incorpore Wifi realice todas las funciones.

Tabla de ilustraciones

| | |
|--|----|
| Ilustración 1. Estructura de la que se parte para realizar el proyecto..... | 6 |
| Ilustración 2. Esquema del proyecto con todos los componentes..... | 10 |
| Ilustración 3. Placa Arduino UNO. | 11 |
| Ilustración 4. Sensor de agua..... | 12 |
| Ilustración 5. Sensor de nivel. | 12 |
| Ilustración 6. Electroválvula. | 13 |
| Ilustración 7. Multiplexor 74HC4051. | 13 |
| Ilustración 8. Relé T010010..... | 14 |
| Ilustración 9. Modulo ESP8266..... | 14 |
| Ilustración 10. Conexión de los sensores de agua con el multiplexor y el Arduino. | 17 |
| Ilustración 11. Pines del sensor de agua..... | 18 |
| Ilustración 12. Depósitos de agua y abono. | 18 |
| Ilustración 13. Entradas y salidas de control de ambos depósitos..... | 19 |
| Ilustración 14. Conexión de las Entradas y salidas totales del Arduino. | 20 |
| Ilustración 15. Algoritmo del proceso de llenado de agua. | 21 |
| Ilustración 16. Algoritmo de comprobación de los sensores de agua..... | 22 |
| Ilustración 17. Algoritmo de comprobación del depósito de abono..... | 22 |
| Ilustración 18. Array estadosensores[10]. | 23 |
| Ilustración 19. Conexiones del Arduino con el multiplexor. | 24 |
| Ilustración 20. Pines multiplexor. | 24 |
| Ilustración 21. Arduino, ESP8266, relés y multiplexor..... | 32 |
| Ilustración 22. Fuente de alimentación. | 33 |
| Ilustración 23. Montaje de los sensores de agua..... | 33 |
| Ilustración 24. Posición del sensor superior (SN) del depósito de agua. | 34 |
| Ilustración 25. Posición del sensor inferior (SN2) del depósito de agua. | 34 |
| Ilustración 26. Conexiones de los sensores de nivel (SN y SN2) con el relé y EV1. | 35 |
| Ilustración 27. Conexión sensor de nivel..... | 35 |
| Ilustración 28. Conexión de SN3 con EV2. | 36 |
| Ilustración 29. Posición del sensor inferior (SN3) del depósito de abono. | 36 |

Referencias

Todas las referencias han sido comprobadas el día 17/04/2020.

Aprendiendo Arduino, Aprendiendo a manejar Arduino en profundidad. Recuperado de: <https://aprendiendoarduino.wordpress.com/2015/03/24/librerias/>

David Blanchard, 6 mayo, 2013. Introducción a C++. Recuperado de: <https://blanchardspace.wordpress.com/2013/05/06/introduccion-a-c-que-es/>

Digi-Key electronics, SparkFun Electronics. Recuperado de: <https://www.digikey.es/product-detail/es/sparkfun-electronics/BOB-13906/1568-1423-ND/6193600>

Electronicshub. How to Send an Email using ESP8266 WiFi Module? Recuperado de: <https://www.electronicshub.org/send-an-email-using-esp8266/#Code>

JADIAZ, 21 enero, 2016 Placa Arduino Uno. Recuperado de: <http://www.iescamp.es/miarduino/2016/01/21/placa-arduino-uno/>

Luis del Valle Hernández, Texto en movimiento en un LCD con Arduino. Recuperado de: <https://programarfacil.com/tutoriales/fragmentos/arduino/texto-en-movimiento-en-un-lcd-con-arduino/>

Luis Llamas, 19 noviembre, 2016 Ingeniería, informática y diseño. Más salidas/entradas en Arduino con multiplexor CD74HC4067. Recuperado de: <https://www.luisllamas.es/mas-salidas-y-entradas-en-arduino-con-multiplexor-cd74hc4067/>

Manual de programación de Arduino: Recuperado de: <http://dfists.ua.es/~jpomares/arduino/index.htm>

Prometec, Arduino y el sensor de agua. Recuperado de: <https://www.prometec.net/sensor-agua/>

Seeed, G1&2 Electric Solenoid Valve (Normally Closed). Recuperado de: <https://www.seeedstudio.com/G1-2-Electric-Solenoid-Valve-Normally-Close-p-636.html>