



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA AUTOMÁTICO PARA LA GESTIÓN DE UN MICROHUERTO URBANO

TRABAJO FINAL DEL

Máster Universitario en Ingeniería Mecatrónica

REALIZADO POR

Sergio López Cortés

TUTORIZADO POR

Carlos Ricolfe Viala

CURSO ACADÉMICO: 2019/2020

DOCUMENTOS QUE CONFORMAN EL PROYECTO

Documento I. MEMORIA y ANEXOS

Documento II. PLIEGO DE CONDICIONES

Documento III. PRESUPUESTO

RESUMEN

Hoy en día, la corriente que apuesta por productos más naturales y sostenibles, especialmente los alimenticios, está en auge. Es por ello por lo que el movimiento de cultivar tu propia comida ha cogido fuerza en estos últimos años. ¿Pero cómo encaja esto en la sociedad actual dónde la mayor parte de la población vive en la ciudad y apenas dispone de tiempo? Aprovechando la tecnología de la que se dispone.

Cada vez es más común ver como la automatización y el IoT (*Internet of Things*) está presente en los hogares de la sociedad moderna. Esto responde a la reducción del coste de los componentes (sensores, microprocesadores, microcontroladores, etc.) haciéndolos accesibles no solo a las industrias. Este tipo de tecnologías facilitan muchas tareas y permiten ahorrar tiempo en el día a día, por lo que también pueden ser de gran ayuda en las tareas de un jardín o huerto urbano.

Así entonces, este proyecto tiene como objetivo la construcción de un prototipo de microhuerto urbano automatizado, utilizando el espacio y recursos materiales mínimos. El presente proyecto se puede descomponer en cuatro etapas principales: Diseño, selección de componentes, implementación del sistema automatizado y desarrollo de una interfaz de usuario que permita la interacción con el sistema.

La automatización del sistema se ha realizado mediante una placa de desarrollo. Son dispositivos, generalmente, con un microcontrolador reprogramable integrado que permite ejecutar programas, que a su vez controlan las diferentes entradas y salidas como sensores y actuadores. Las placas más comunes y conocidas son las de la marca *Arduino*, pioneros en el sector y desarrolladores de un software *open-source* (de dominio público) que facilitaba la programación de estas placas y las acercaba a público de diferentes sectores.

Pese a que no se ha contado con una placa de la marca, sí es compatible con el software de programación, por lo que el programa de control del sistema sería compatible con cualquier otra placa de características similares y compatible con el software de *Arduino*. Con este trabajo se pretende, a su vez, profundizar en esta área y comprobar el gran potencial de estos dispositivos, cada vez más populares y que facilitan en gran medida la creación de prototipos para la automatización de sistemas.

Palabras claves: microhuerto urbano, IoT, diseño, automatización, prototipo, placa de desarrollo, *Arduino*, open-source.

ABSTRACT

Nowadays, the trend regarding more natural and sustainable products, especially foodstuffs, is booming. This is the reason why the movement of growing your own food has gained strength in recent years. But how does this fit into today's society where most of the population lives in the city and barely has time? This can be done taking advantage of the technology currently available.

It is increasingly common to see how automation and IoT (Internet of Things) is present in modern society's homes. It responds to the reduction in costs of components (sensors, microprocessors, microcontrollers, etc.) making them accessible not only for industry. This type of technology eases people's many daily tasks and saves time in their day to day life. So, they can also be a great help in garden, or urban garden, tasks.

Thus, the aim of the project is to build an automated urban micro vegetable garden prototype, using minimal space and minimal material resources. This project can be broken down into four main stages: Design, component selection, automated system implementation and a user interface development that allows interaction with the system.

System automation has been done using a development board. They are usually devices with an integrated reprogrammable microcontroller that allows running programs, which in turn manage the inputs and outputs such as sensors and actuators. The most popular boards are those of the *Arduino* brand, industry pioneers and developers of an open-source software that facilitated board programming. Moreover, they brought these boards closer to audiences from different sectors.

Although this project has not been done with an *Arduino* board, it does support its software. So, the system control program would be supported by any other board with similar features and compatible with Arduino software. This project aims, in turn, to deepen this area and prove the great potential of these devices that facilitate creation of prototypes for automation systems so much.

Keywords: urban micro vegetable garden, IoT, design, automation, prototype, development board, *Arduino*, open-source software.

AGRADECIMIENTOS

A mi familia y en especial a mi pareja por todo el apoyo incondicional recibido durante estos meses, sin vosotros no hubiera sido posible.

Dar las gracias a todas esas personas, conocidas o encontradas en el camino, dispuestas a ayudar de corazón sin esperar nada a cambio.

Por último, agradecer a mi facultad el aporte de recursos materiales y docentes necesarios para el desarrollo de este proyecto.

Muchas gracias.

Documento I. MEMORIA y ANEXOS

ÍNDICE DE CONTENIDOS

1. INTRODUCCIÓN.....	1
1.1. Antecedentes.....	1
1.2. Objetivos.....	2
1.3. Alcance de la memoria.....	2
2. DISEÑO DEL PROTOTIPO	3
2.1. Proceso creativo y optimización del diseño.....	3
2.1.1. Estación meteorológica	7
2.2. Placa de desarrollo.....	8
2.2.1. Selección de la placa de desarrollo	8
2.3. Selección de los sensores y actuadores	9
2.3.1. Bomba de riego.....	10
2.3.2. Módulo relé KF-301	10
2.3.3. Módulo sensor de ultrasonidos HC-SR04 y Sensor de nivel XKC-Y25-V.....	11
2.3.4. Módulo higrómetro v1.0 y Módulo higrómetro v1.2	12
2.3.5. Sonda de temperatura DS18B20.....	14
2.3.6. Sensor de temperatura y humedad DHT11	15
2.3.7. Anemómetro y Sensor de luminosidad	16
2.3.8. Módulo Wifi ESP8266.....	17
3. FUNCIONAMIENTO DE LOS COMPONENTES	19
3.1. Módulo de riego.....	19
3.1.1. Calibración del módulo de ultrasonidos HC-SR04	19
3.1.2. Simulación del sistema de riego	21
3.2. Módulos higrómetro v1.0 y v1.2.....	23
3.3. Sonda DS18B20 y Sensor DHT11	25
3.4. Calibración del LDR.....	28
3.5. Calibración del anemómetro	31
4. SISTEMA DE CONTROL.....	33
4.1. Flow Chart.....	33

4.1.1.	Sample Time de la lectura de datos.....	34
4.2.	Código de Programa	36
4.2.1.	Cabecera	36
4.2.2.	Uso del timer 1	36
4.2.3.	Bucle de control.....	38
4.3.	Interfaz de Usuario	39
5.	MONTAJE DEL PROTOTIPO.....	42
5.1.	Construcción	42
5.1.1.	Bancal	42
5.1.2.	Riego.....	42
5.1.3.	Estación meteorológica	43
5.2.	Diagrama de conexiones.....	44
5.2.1.	Optoacopladores.....	45
5.2.2.	Amplificador Operacional no inversor	46
5.2.3.	Elementos resistivos.....	46
5.2.4.	Placa de conexiones.....	47
5.3.	Montaje final y puesta en marcha.....	48
6.	MEJORAS Y PRÓXIMOS PASOS	50
7.	CONCLUSIÓN.....	51
8.	BIBLIOGRAFÍA	52
9.	ANEXOS.....	53
9.1.	Anexo I: Manual de Usuario	53
9.1.1.	Configuración del acceso Wifi	53
9.1.2.	Conexión Intranet – Internet	54
9.1.3.	Interfaz de usuario.....	54
9.1.4.	Mantenimiento	55
9.2.	Anexo II: Flow Chart.....	56
9.3.	Anexo III: Esquemas eléctricos	57
9.4.	Anexo IV: Código del Programa	59

ÍNDICE DE FIGURAS

Figura 1 – Ejemplo de una mesa de cultivo	1
Figura 2 – Primera versión planteada del prototipo y detalle de los sensores y la estación meteorológica	3
Figura 3 – Segunda versión del prototipo	4
Figura 4 – Diseño final del prototipo	5
Figura 5 – Diseño de la estación meteorológica.....	7
Figura 6 – Placa de desarrollo Elegoo Mega 2560 R3	8
Figura 7 – Bomba escogida para el riego	10
Figura 8 – Módulo Relé KF-301.....	10
Figura 9.a y 9.b – Módulo HC-SR04 y sensor de nivel XKC-Y25-V	11
Figura 10.a y 10.b – Módulo higrómetro v1.0 y v1.2.....	13
Figura 11 – Sonda de temperatura DS18B20	14
Figura 12 – Sensor de temperatura y humedad DHT11.....	15
Figura 13 – Anemómetro de fabricación propia	16
Figura 14 – LDR y Esquema de montaje del sensor	17
Figura 15 – Módulo Wifi ESP8266 ESP-01	17
Figura 16 – Fuente de alimentación para protoboard	18
Figura 17 – Montaje para la prueba de calibración del módulo HC-SR04.....	19
Figura 18 – Gráfica de los resultados obtenidos en la prueba del módulo HC-SR04	20
Figura 19 – Gráfica de repetibilidad en la medición con el módulo HC-SR04.....	20
Figura 20.a, 20.b y 20.c – Imágenes del montaje para la prueba del sistema de riego y detalles de los componentes y sus conexiones	22
Figura 21 – Gráfica de los valores obtenidos en el ensayo de medición.....	23
Figura 22.a y 22.b – Gráfica de la evolución de las lecturas de los higrómetros v1.0 y v1.2	25
Figura 23.a y 23.b – Gráfica de las lecturas de temperatura (°C).....	26
Figura 24.a y 24.b – Gráfica de las lecturas de humedad relativa del aire (%).....	27
Figura 25 – Gráfica lux-resistencia del LDR.....	29
Figura 26 – Gráfica de la relación salida PWM - lux.....	30
Figura 27 – Gráfica de las lecturas de luminosidad tomadas por cada sensor	30
Figura 28.a y 28.b – Anemómetro fijado al techo y conexión con la placa de desarrollo y el ordenador	31
Figura 29 – Gráfica velocidad – voltaje del anemómetro	32
Figura 30 – Gráfica del muestreo de la humedad y la temperatura del terreno.....	35
Figura 31 – Fragmento de código de la configuración del timer 1	37
Figura 32 – Esquema de bits del registro TCCR1B.....	37

Figura 33 – Esquema de bits del registro TIMSK.....	38
Figura 34 – Fragmento de código de la rutina de interrupción.....	38
Figura 35 – Fragmento de código de la configuración del timer de riego automático	39
Figura 36 – Fragmento de código de la configuración del módulo Wifi.....	40
Figura 37 – Imagen de la interfaz de usuario	41
Figura 38 – Imagen de la estructura del bancal de cultivo	42
Figura 39 – Imagen del bancal con el sistema de riego instalado.....	43
Figura 40 – Detalle del sistema de riego y el depósito.....	43
Figura 41 – Imagen de la estación meteorológica sin todos sus componentes	44
Figura 42.a y 42.b – Esquema para una salida optoacoplada y esquema de alimentación mediante optoacoplador.....	45
Figura 43 – Circuito de un amplificador operacional no inversor.....	46
Figura 44 – Imagen de la placa de conexiones.....	47
Figura 45.a y 45.b – Despliegue de todos los dispositivos que conforman el sistema e imagen del conjunto montado	48
Figura 46 – Primeras pruebas del sistema completo	49
Figura 47 – Imagen del montaje final del prototipo	49
Figura 48 – Fragmento del código donde se introducen las credenciales Wifi	53
Figura 49 – Ejemplo de la lista de direcciones IP recibida	53
Figura 50 – Imagen del primer apartado de la interfaz de usuario	54
Figura 51 – Imagen del segundo apartado de la interfaz de usuario.....	54
Figura 52 – Imagen del tercer y último apartado de la interfaz de usuario.....	55

ÍNDICE DE TABLAS

Tabla 1 – Tabla comparativa entre los higrómetros V1.0 y V1.2.....	13
Tabla 2.a y 2.b – Tabla de resultados de la prueba del sistema de riego	22
Tabla 3 – Tabla para el cálculo del sample time	35

1. INTRODUCCIÓN

1.1. Antecedentes

Estos últimos años, se ha podido percibir como las personas apuestan por productos cada vez más naturales, saludables y sostenibles en el tiempo. Su máximo exponente se encuentra en los productos alimenticios, por ello cada vez más personas se introducen en el mundo de los cultivos, pero en un mundo tan cambiante, rápido y urbanizado, es complejo para mucha gente llevarlo a cabo. Aquí es donde aparece la tecnología y el punto de entrada a este proyecto.

La idea de los huertos urbanos hace tiempo que convive en la sociedad actual, aunque en la mayoría de los casos se ven en terrazas o patios de gran superficie. La realidad es que la mayor parte de la población vive en las urbes con espacios exteriores reducidos y, en respuesta a esto, han surgido diferentes opciones como los “macetohuertos” y las mesas de cultivo de diferentes formas y tamaños que se adaptan a estas condiciones.



Figura 1 – Ejemplo de una mesa de cultivo. [Imagen]. Recuperado de <https://www.jardinitis.com>. (2020)

Pese a esto, aún faltaría encontrar la solución que se adapte al estilo de vida moderno, es decir, poder ausentarse del hogar sin que los cultivos perezcan. Desde un dispositivo móvil ya es posible controlar varios elementos del hogar fácilmente como la temperatura, las luces o las persianas, entonces ¿por qué no hacer lo mismo con los cultivos domésticos?

1.2. Objetivos

El objetivo de este trabajo final de máster es el diseño y construcción de un prototipo de microhuerto urbano automatizado. Pretende ser una unidad funcional de cultivo que disponga de una interfaz de usuario a través de una aplicación móvil.

Este prototipo ha sido desarrollado mediante una placa de desarrollo, unos dispositivos de bajo coste que permiten actuar sobre diferentes sensores y actuadores con entradas y salidas comúnmente digitales. En apartados posteriores de la presente memoria se entrará en más detalle.

Con este proyecto, se pretende también profundizar y aprovechar el potencial que brindan dichas placas de desarrollo y como pueden ayudar en proyectos, tanto a nivel académico como profesional.

1.3. Alcance de la memoria

En esta memoria se ha redactado todo el proceso de creación del prototipo desde cero, haciendo especial énfasis en la parte de automatización, control e interfaz de usuario, siendo esta una de las más relacionadas con la titulación del máster.

Como se ha comentado, el objetivo es crear un prototipo funcional y operativo, haciendo uso de los mínimos recursos materiales y económicos, con las limitaciones que ello conlleva. En ningún caso se pretende hacer de este una aplicación comercial ya que para ello sería menester una serie de condiciones que se detallarán más adelante en esta memoria.

2. DISEÑO DEL PROTOTIPO

2.1. Proceso creativo y optimización del diseño

Durante el proceso de diseño surgieron varios planteamientos hasta llegar al diseño óptimo que cumpliera los requisitos y objetivos planteados, cabe recordar que estos eran, principalmente, crear una unidad funcional con el mínimo uso de recursos.

El primer planteamiento se observa en la siguiente imagen:

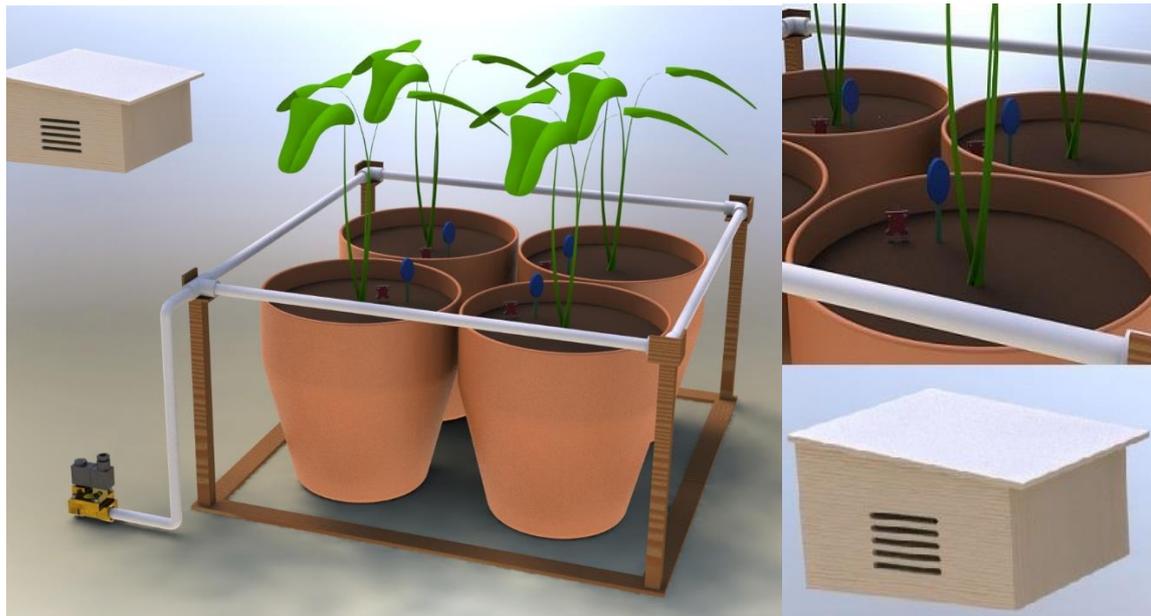


Figura 2 – Primera versión planteada del prototipo y detalle de los sensores (derecha arriba) y la estación meteorológica (derecha abajo)

La idea de este diseño es agrupar diversos cultivos mediante el uso de macetas como contenedor de estos. Fijados en el sustrato de cada maceta, tendríamos un sensor de humedad y un sensor de temperatura, parámetros que sirven para monitorear el estado del suelo y activar el riego. El riego se activa mediante una electroválvula de 12 VDC conectada, a su vez, al sistema de agua corriente del hogar. Como se puede observar, una vez activada la electroválvula, el agua circularía por un tubo perforado de PVC, regando por aspersión.

Por último, se aprecia en la figura 2, el detalle de una estación meteorológica. Esto responde a la necesidad de crear una estación que permita almacenar la placa de desarrollo y todos los componentes y conexiones. Bajo esta premisa, se decidió aprovechar este espacio para crear una estación meteorológica de la que se profundizará al final de este punto.

Ventajas de este diseño:

- Facilidad de montaje de la zona de cultivos.
- Contenedor independiente para cada cultivo.

Inconvenientes:

- Se requiere de una llave de paso cercana para alimentar el circuito de riego.
- Se debe adaptar el tubo a las conexiones de la electroválvula.
- Todas las macetas deberían ser de igual o similar medida.
- Dos sensores por cada maceta, mayor uso de recursos y coste.
- Monitoreo de cada suelo de forma independiente pero un solo sistema de riego.
- El tubo de PVC es rígido por lo que se necesitan varios accesorios y complica el montaje del sistema de riego.
- Desaprovechamiento del espacio.

Viendo los inconvenientes encontrados en el primer diseño se realizó una segunda versión:



Figura 3 – Segunda versión del prototipo

En esta segunda versión se cambian las macetas convencionales por cajas de plástico, usadas comúnmente para el transporte de frutas y verduras. Estas se revisten con una lona geotextil, que evita la pérdida de sustrato por los orificios a la vez que permite el correcto drenaje del sobrante de agua de riego o lluvia. Aprovechando la estructura de las cajas se sustituye el tubo de PVC por un tubo de riego estándar, facilitando en gran medida el montaje del sistema de riego.

Ventajas de este diseño:

- Facilidad de montaje de la zona de cultivos y reciclaje de materiales.
- Contenedor independiente para cada cultivo.
- Tubo de riego comercial estándar, facilidad de adquisición y menor coste.
- Tubo no rígido, facilidad de instalación.
- Mayor aprovechamiento del espacio debido a la forma de los contenedores.

Inconvenientes:

- Se requiere de una llave de paso cercana para alimentar el circuito de riego.
- Es necesaria una electroválvula con conexiones de entrada y salida concretas que se adapten al tubo de riego y a la llave de paso.
- Dos sensores por cada maceta, mayor uso de recursos y coste.
- Monitoreo de cada suelo de forma independiente pero un único sistema de riego.

Se mejoraron varios puntos desfavorables del primer diseño pero aún había ciertas carencias que alejaban el diseño de los objetivos planteados. Por ello, se optimizó el diseño creando así la versión final del prototipo:



Figura 4 – Diseño final del prototipo

Para el diseño final se sustituyó de nuevo el contenedor de cultivos, esta vez, por un banco de madera que recoge todos los cultivos en un único sustrato. Este método de cultivo fue ideado por Mel Bartholomew, un ingeniero estadounidense y jardinero aficionado que

revolucionó el mundo de los cultivos usando bancales de 1 m², dividiendo estos en celdas de aproximadamente 30x30 cm y cultivando una o varias plantas por celda según el tipo de cultivo. Esto permitió que muchas personas alrededor del mundo empezaran a cultivar su propia comida en su terraza o balcón.

Además de ser la manera más eficiente de cultivar según los avaladores del método, de esta manera se facilita aún más la instalación del sistema de riego. Por otro lado, se pasa a usar una única pareja de sensores de temperatura y humedad para el control del riego, ya que solo se precisa monitorear un único suelo.

Otro cambio significativo de este último diseño es el sistema de riego, este se realiza mediante un depósito y una pequeña bomba hidráulica, de esta forma se otorga al sistema independencia del suministro de agua y de la disponibilidad de una llave de paso cercana. El depósito incluye un sensor de nivel que enviará una alerta si el nivel de agua baja del umbral e inhabilitará la bomba para evitar que se dañe.

Ventajas de este diseño:

- Reciclaje de materiales, se puede realizar con madera de pallets.
- Mayor eficiencia del espacio de cultivo al tener un suelo único e intercambio de nutrientes entre plantas (asociación de cultivos).
- Solo dos sensores para el control de todo el suelo de cultivo.
- Tubo de riego comercial estándar, facilidad de adquisición y menor coste.
- Tubo no rígido, facilidad de instalación.
- Independencia de suministro de agua corriente.
- El depósito también permite utilizar agua de lluvia o reposada y de esta manera, reducimos la presencia de cloro y cal en el agua de riego.

Inconvenientes:

- Requiere la construcción o compra del bancal.
- Se requiere que la conexión de la bomba se ajuste al tubo de riego o adaptar este último a la salida de la bomba.

Como se observa, en esta versión final se obtienen más ventajas que inconvenientes y cabe decir, estos inconvenientes son mucho menores respecto de las versiones anteriores. Este diseño entonces sí permite alcanzar los objetivos y requisitos iniciales.

2.1.1. Estación meteorológica

Como se ha comentado, el propósito de la estación meteorológica, además de su propia función, es albergar todos los componentes y conexiones necesarias para el funcionamiento del sistema. En este punto se pretende desarrollar con más detenimiento su diseño y componentes.

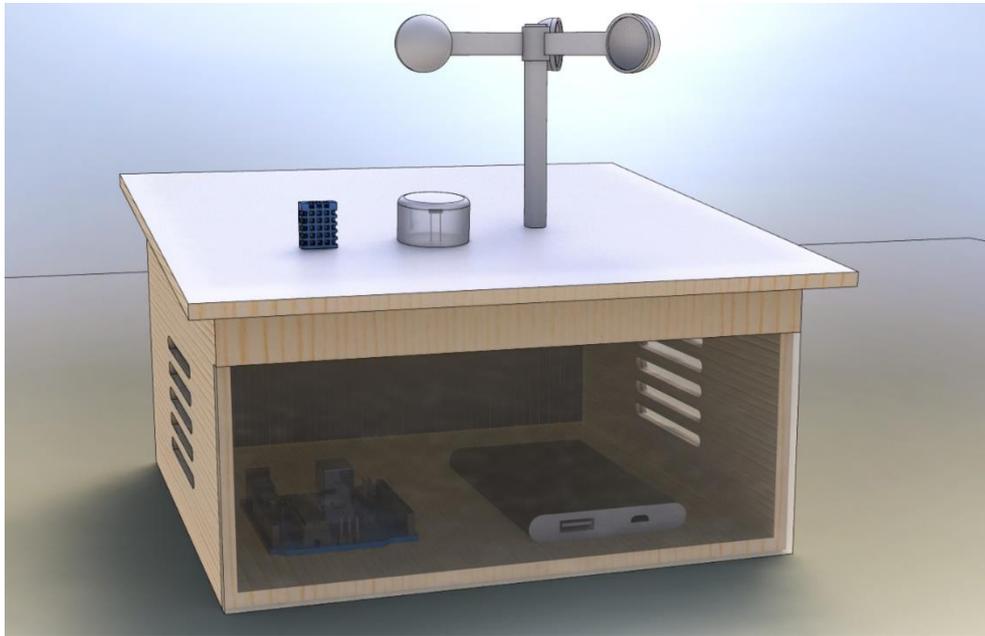


Figura 5 – Diseño de la estación meteorológica

La estación se diseñó en madera como material principal. Con una cubierta inclinada recubierta de algún componente hidrófobo, sea mediante pintura o adhesivo, que ayude a evacuar el agua más fácilmente. Dispone de unas aberturas laterales para ventilación y disipación del calor generado por los componentes electrónicos. También dispone de una puerta delantera que permite acceder al interior, que puede ser de la misma madera o de plástico, para que el interior quede visible.

En cuanto a los componentes que la forman, se encuentra la placa de desarrollo, la batería de alimentación de la placa (se ha representado un *powerbank* comercial, aunque puede ser un banco de pilas, pila de 9V, transformador AC/DC u otra alternativa), una placa de conexiones, un sensor de temperatura y humedad del aire, un anemómetro y una fotorresistencia, protegida del ambiente exterior, que ayuda a determinar el riego, puesto que en las horas de más sol no se debe regar ya que no sería eficiente.

2.2. Placa de desarrollo

Las placas de desarrollo se han popularizado en los últimos años mayormente gracias a *Arduino*, una plataforma de electrónica *open-source* que nació con la idea de facilitar y promover el uso de microcontroladores en proyectos técnicos tanto a nivel educativo como profesional.

El microcontrolador es el corazón de la placa y el encargado de ejecutar las instrucciones necesarias que controlan las entradas y salidas con un fin específico. El resto de la placa es, a modo general, extensiones de estas entradas y salidas preparadas para facilitar la conexión y comunicación con los sensores y actuadores.

Una de las características más importantes de esta plataforma es la programación de la placa. Esta se realiza mediante un software y un lenguaje de programación propio. Este IDE (*Integrated Development Environment*) tiene como objetivo facilitar la programación del microcontrolador mediante funciones y librerías que permiten controlar las diferentes entradas y salidas sin ser un experto programador.

Por todo ello, se ha escogido este componente como parte principal del prototipo para el control y comunicación de todos los sensores y actuadores necesarios.

2.2.1. Selección de la placa de desarrollo

Después del éxito de este tipo de plataformas no es extraño que hayan aparecido competidores que ofrezcan alternativas similares a los productos de *Arduino* y, para este proyecto, se ha escogido la placa *Elegoo Mega 2560 R3* de la casa china *Shenzhen Elegoo Technology*.



Figura 6 – Placa de desarrollo Elegoo Mega 2560 R3. [Imagen]. Recuperado de <https://www.elegoo.com>. (2020)

Esta placa ofrece unas características muy similares a las del *Arduino Mega* por un coste menor, ambas integran el mismo microcontrolador el *ATmega2560* de 8 bits y 256 KB de *memoria flash* (memoria destinada al programa) desarrollado por *Atmel* y distribuido actualmente por *Microchip*. Otro punto relevante de esta elección es que los productos ofrecidos por esta compañía son 100% compatibles con el IDE de Arduino.

Según datos del fabricante, la Elegoo Mega 2560 R3 ofrece las siguientes características:

- Voltaje operativo: 5 VDC.
- Voltaje de entrada: 7 – 12 VDC (límite mín. 6 VDC, límite máx. 20 VDC).
- Entradas y salidas digitales: 54 (15 de ellas pueden proporcionar señales PWM).
- Entradas analógicas: 16 (convertor ADC de 10 bits – valores 0 a 1023).
- Corriente máxima de las salidas: 40 mA.
- Corriente máxima del pin de 3,3 V: 50 mA.
- Corriente límite de la placa: 500 mA (dispone de fusible de seguridad).
- Memoria flash: 256 KB (de los que 8 KB se usan para el gestor de arranque).
- SRAM: 8 KB.
- EEPROM: 4 KB.
- Velocidad del Reloj: 16 MHz.

Como se aprecia en las características, disponemos de un gran número de entradas y salidas, tanto analógicas como digitales, esto, junto a una mayor memoria de programa, fue un punto determinante para no escoger el modelo más básico, la *Elegoo R3* o su equivalente *Arduino Uno*, los cuales solo disponen de 14 entradas y salidas digitales y 32 KB de memoria flash. La Elegoo Mega tiene un coste algo mayor, pero poco significativo si analizamos las prestaciones que ofrece frente a la Elegoo R3.

2.3. Selección de los sensores y actuadores

Una vez escogido el diseño óptimo y la placa de desarrollo, se realizó la selección de los sensores y actuadores más adecuados para llevar a cabo todas las funciones del prototipo descritas en el desarrollo del diseño.

Tal como se ha comentado al inicio de esta memoria, el objetivo era realizar un prototipo de bajo coste mediante una placa de desarrollo. Por tanto, el coste y la compatibilidad con la placa elegida fueron los parámetros más relevantes para la búsqueda y selección de todos los actuadores y sensores que a continuación se describen.

2.3.1. Bomba de riego

La bomba de riego seleccionada es una “minibomba” usada comúnmente para acuarios y estanques. La bomba se alimenta mediante una conexión USB a un voltaje de entre 3,5 y 9 VDC correspondientes a una potencia de entre 1 y 3 W respectivamente. Esto permite conectar la bomba fácilmente mediante un cargador de móvil o aprovechando la misma fuente de alimentación usada para la placa.



Figura 7 – Bomba escogida para el riego. [Imagen]. Recuperado de <https://www.amazon.es>. (2020)

Según los datos del fabricante, dicha bomba ofrece un caudal regulable de hasta 200 L/h y una altura máxima de bombeo de 1,5 metros. Estas características se ajustan a los parámetros requeridos un riego por goteo (lo que no supone un caudal elevado, y la altura del a vencer no supera los 0,5 metros.

2.3.2. Módulo relé KF-301

Este relé es el encargado de activar la alimentación de la bomba una vez se envíe la orden de activar el riego. Está integrado en un módulo preparado el uso en placas de desarrollo. Por un lado, dispone de tres bornes, uno para la entrada de voltaje y los dos restantes para determinar el funcionamiento del contactor, NA/NC (Normalmente Abierto/ Normalmente Cerrado). Por otro lado, se encuentra la parte de control y las conexiones por las cuales se comunica el relé con la placa y recibe la señal para abrir o cerrarse según la configuración escogida.



Figura 8 – Módulo Relé KF-301. [Imagen]. Recuperado de <https://www.az-delivery.de/es>. (2020)

Características principales según datos del fabricante:

- Señal de control: 3,3 a 5 VDC.
- Límites en Corriente Alterna: 10 A / 250 V.
- Límites en Corriente Continua: 10 A / 30 V.
- Tipo de contacto: Ambos - NA / NC.

2.3.3. Módulo sensor de ultrasonidos HC-SR04 y Sensor de nivel XKC-Y25-V

Tal como se ha indicado en apartados anteriores, el depósito de agua incluye un sensor de nivel para avisar al usuario de la falta de agua y, a su vez, proteger a la bomba de un mal funcionamiento.

Después de la búsqueda de información sobre los diferentes sensores que pudieran realizar esta función, se acotó a dos tipos de sensores, los sensores de ultrasonidos y los sensores de nivel capacitivos, concretamente el módulo *HC-SR04* y el sensor *XKC-Y25-V* respectivamente.



Figura 9.a y 9.b – Módulo *HC-SR04* (izquierda) y sensor de nivel *XKC-Y25-V* (derecha). [Imágenes]. Recuperado de <https://www.amazon.es>. (2020)

El módulo *HC-SR04* se basa en enviar una señal ultrasónica y calcular la distancia hasta el objeto de medición, midiendo el tiempo que demora la señal en regresar al sensor. Una vez determinado el tiempo y conociendo la velocidad a la que viaja el sonido en el aire, se obtiene la distancia fácilmente. Mediante este módulo calculamos la distancia desde el borde del depósito hasta el agua, sabiendo así cuándo el nivel del agua está por debajo del umbral.

Las características más relevantes del módulo *HC-SR04* son:

- Voltaje operativo: 5 VDC.
- Corriente de trabajo: 15 mA.
- Frecuencia de ultrasonido: 40 kHz.

- Distancia mínima medible: 2 cm.
- Distancia máxima medible: 400 cm.
- Precisión: ± 3 mm.
- Ángulo de medición: 15° .
- Señal del disparador: pulso TTL de 10 μ s.
- Señal de salida: señal TTL de amplitud proporcional al rango.

El sensor XKC-Y25-V, en cambio, es un sensor capacitivo, es decir, mide la variación de capacitancia provocada por un objeto que entre en el campo eléctrico generado. Materiales con una constante dieléctrica baja como el papel, el plástico o el vidrio no superaran el umbral mínimo y se obtendrá una señal de nivel bajo, en cambio los materiales con una constante dieléctrica mayor como el agua o metales generarán una variación de capacidad significativa y cambiarán la salida al nivel alto. Es por ello por lo que este sensor está especialmente indicado para medir la presencia de líquidos a través de tuberías y depósitos, siempre que estos sean de materiales aislantes y su espesor no supere el rango de medición.

Las características más relevantes del sensor XKC-Y25-V son:

- Voltaje operativo: 5 – 24 VDC.
- Señal de salida: 0 VDC (a nivel bajo) / Voltaje de entrada (a nivel alto).
- Corriente de trabajo: 5 mA.
- Tiempo de respuesta: 500 ms.
- Temperatura de trabajo: 0 – 105 °C.
- Humedad de trabajo: 5 – 100%.
- Rango de medición: 0 – 13 mm.
- Impermeabilidad: IP67.

2.3.4. Módulo higrómetro v1.0 y Módulo higrómetro v1.2

Uno de los puntos más importantes es el sensor de humedad del suelo ya que es el principal componente para determinar la activación del riego. Hay mucha discusión sobre qué tipo de higrómetros son mejores. Actualmente, entre los modelos más económicos, los higrómetros más usados son los resistivos y los capacitivos. En este caso se adquirieron dos sensores, uno de cada tipo, modelos propios del distribuidor alemán *AZ-Delivery*.

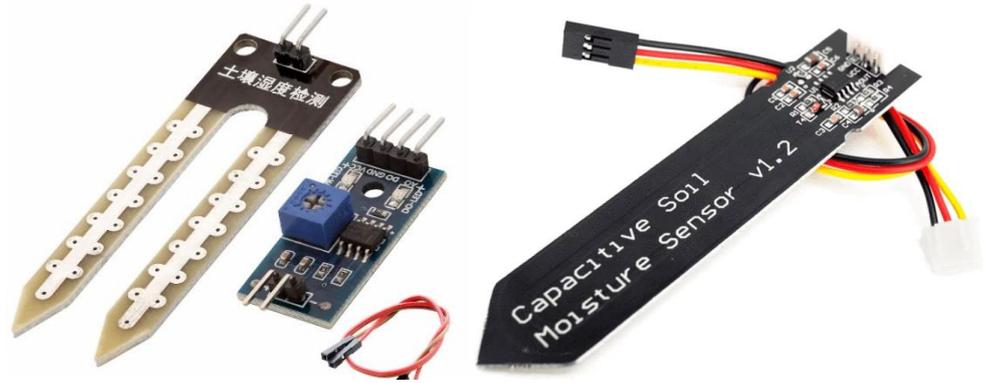


Figura 10.a y 10.b – Módulo higrómetro v1.0 (izquierda) y v1.2 (derecha). [Imágenes].
Recuperado de <https://www.az-delivery.de/es>. (2020)

El higrómetro v1.0 es un sensor resistivo, es el modelo más clásico y se basa en dos electrodos aislados que generan un flujo de corriente entre ambos a través del suelo, o el medio en el que se encuentre. A mayor humedad del suelo menor será la resistencia que ofrece entre los electrodos, esta variación de resistencia es medible y se usa para determinar la humedad relativa del suelo. Este modelo está compuesto por la parte pasiva, los electrodos, y una parte activa, la placa de control *LM393*, un chip comparador encargado de medir esta resistencia y convertirlo en una señal analógica.

Por otro lado, la versión 1.2 del higrómetro es un sensor capacitivo y su funcionamiento es el mismo explicado en el apartado anterior, dispone de dos placas que conforman un condensador. Al alimentar las placas se genera un campo eléctrico y cualquier material, en este caso el sustrato, que se acerque a este campo se transforma en el dieléctrico del condensador y varía su capacidad, este cambio se puede medir y activar una salida, como era el caso del sensor de nivel, o convertirlo en una salida analógica, como en este caso.

En la siguiente tabla se pueden observar las principales características y diferencias entre ambos sensores:

	Higrómetro v1.0	Higrómetro v1.2
Tipo de sensor	Resistivo	Capacitivo
Voltaje operativo	3,3 – 5 VDC	3,3 – 5 VDC
Consumo de corriente	8 mA	5 mA
Salida analógica	Sí	Sí
Salida digital	Sí	No
Convertidor	LM393	Integrado
Calibración inicial	No	Sí
Resistente a la corrosión	No	Sí

Tabla 1 – Tabla comparativa entre los higrómetros V1.0 y V1.2

Los higrómetros capacitivos han cogido fuerza frente a los resistivos ya que estos sufren problemas de corrosión debido al flujo de corriente entre los electrodos y la humedad del entorno de trabajo. Esto también provoca que sean más sensibles a los cambios de temperatura y a la condensación, además de un desgaste mayor y, por consiguiente, una vida útil mucho menor. Aunque no se especifica en los documentos ofrecidos por el proveedor, según la literatura encontrada en internet los higrómetros capacitivos ofrecen una sensibilidad, precisión, estabilidad y robustez mayores. Estos parámetros son estudiados en apartados posteriores de esta memoria.

2.3.5. Sonda de temperatura DS18B20

La temperatura del suelo no es el factor más determinante para el huerto, pero si puede aportar una valiosa información para varios temas, como precisar mejor la hora de riego, germinación de semillas, selección y trasplante de cultivos, etc. Es por ello por lo que se decidió incorporar una sonda de temperatura en el suelo del microhuerto.

El sensor de temperatura *DS18B20* es un sensor muy completo y económico que permite medir temperaturas de entre -55°C y $+125^{\circ}\text{C}$ con una resolución programable de 9 a 12 bits. El sensor se encuentra encapsulado herméticamente en una sonda de acero inoxidable que permite medir temperaturas de gases, líquidos o en ambientes especialmente húmedos, como es el caso que nos ocupa.



Figura 11 – Sonda de temperatura DS18B20. [Imagen]. Recuperado de <https://www.az-delivery.de/es>. (2020)

La sonda dispone de la tecnología de comunicación *1-Wire* que permite comunicarse tanto con el controlador como con otros sensores usando un único cable de datos o bus. Los sensores que incorporan esta tecnología disponen de una memoria ROM, programada de fábrica, donde se almacena el modelo del sensor, un código de comprobación de errores y un número de serie único para cada sensor, de esta manera se asegura la correcta comunicación e identificación de cada sensor conectado al bus. Otra ventaja de esta tecnología es el modo “parásito”, por el cual se prescinde de la alimentación individual de cada dispositivo y se alimentan mediante la propia línea de comunicación.

Estas son sus características más relevantes:

- Voltaje operativo: 3 – 5,5 VDC.
- Corriente de trabajo: 1,5 mA.
- Corriente en standby: 1 μ A.
- Rango de medición: -55°C – 125 °C.
- Precisión: $\pm 0,5^\circ\text{C}$ en la banda de -10°C a 85°C y $\pm 2^\circ\text{C}$ en todo el rango.
- Resolución: $\pm 0,5^\circ\text{C}$ / $\pm 0,25^\circ\text{C}$ / $\pm 0,125^\circ\text{C}$ / $\pm 0,0625^\circ\text{C}$ (9 a 12 bit respectivamente).
- Tiempo de adquisición: 93,75 ms / 187,5 ms / 375 ms / 750 ms (9 a 12 bits respectivamente).

2.3.6. Sensor de temperatura y humedad DHT11

En la parte de la estación meteorológica nos encontramos con el sensor de humedad y temperatura del aire *DHT11*. Es un sensor económico y muy usado en proyectos con placas de desarrollo por su fácil implementación. Aunque ofrece una sensibilidad y precisión limitadas, es apto para este proyecto debido a que estos datos son meramente informativos y ninguna función depende de ellos.

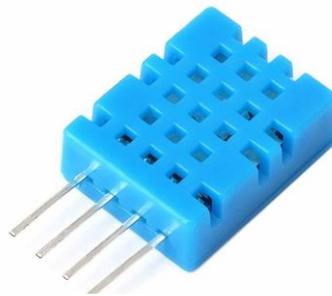


Figura 12 – Sensor de temperatura y humedad DHT11. [Imagen]. Recuperado de <https://www.az-delivery.de/es>. (2020)

Estas son sus características más relevantes:

- Voltaje operativo: 3 – 5,5 VDC.
- Corriente de trabajo: 2,5 mA.
- Rango de medición de temperatura: 0°C – 50°C ($\pm 2^\circ\text{C}$ de precisión).
- Rango de medición de humedad: 20% – 90% ($\pm 5\%$ de precisión).
- Resolución: 0,1°C para temperatura / 1% para humedad.
- Frecuencia máxima de muestreo: 1 Hz.

2.3.7. Anemómetro y Sensor de luminosidad

Uno de los componentes más característicos de las estaciones meteorológicas es sin duda el anemómetro. Debido a la dificultad para hallar un anemómetro de bajo coste y compatible con la placa de desarrollo, se tomó la decisión de construir uno propio. Para su construcción se utilizó un motor de 2,4 VDC recuperado de una afeitadora eléctrica, unas “hélices” creadas con un tapón de garrafa, un lápiz y una huevera de plástico, y una pieza de desagüe de PVC a modo de base.

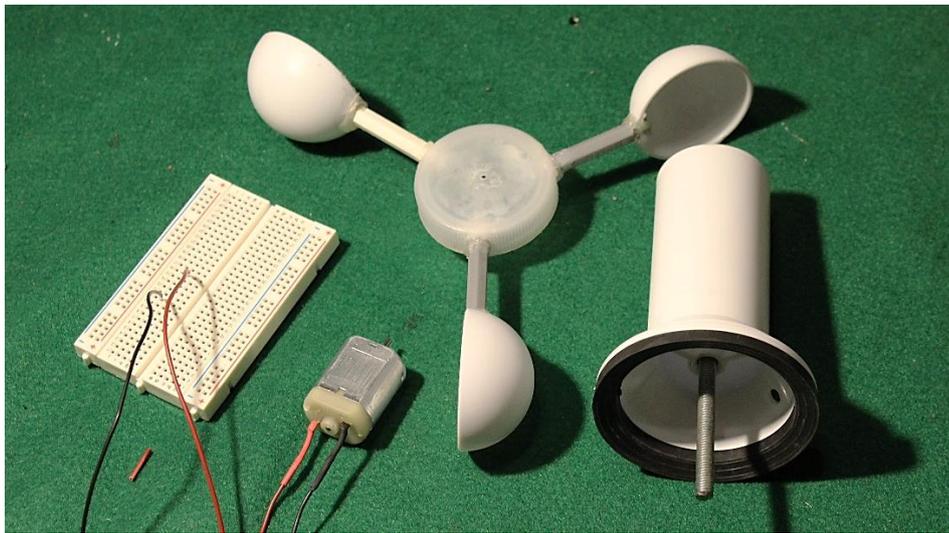


Figura 13 – Anemómetro de fabricación propia

Mediante dos pequeñas piezas de madera se fijaron las hélices al eje del motor, y este conjunto, a su vez, a la base. Aprovechando también la rosca hembra que incorpora la pieza de desagüe, se introdujo una varilla roscada que sirviera para fijar el dispositivo posteriormente a la cubierta de la estación meteorológica. Por último, se soldaron a la alimentación del motor dos cables rígidos que facilitan la conexión con la placa sin necesidad de componentes adicionales. Mediante estos cables, y a través de una de las entradas analógicas, se leen los valores de voltaje devueltos por el motor.

En último lugar, en lo que a sensores se refiere, se encuentra el sensor de luminosidad que indicará las horas de mayor sol. Tal como se señaló anteriormente, el riego no se puede realizar a pleno sol debido a que el agua se evapora a mayor velocidad y las gotas pueden causar quemaduras en los cultivos por efecto lupa, por tanto, además de ineficiente, puede ser perjudicial.

El sensor no es más que una fotorresistencia, o LDR por sus siglas en inglés *Light-Dependent Resistor*, dispuesta a modo de divisor de tensión entre la alimentación y la masa de la placa de desarrollo, como se observa en la figura 14. Del divisor de tensión se obtiene

una lectura analógica y se calcula la resistencia ofrecida por el fotorresistor en cada momento. Con el valor de la resistencia, mediante los valores de conversión que proporciona el fabricante, se obtiene la intensidad lumínica del entorno (lux).

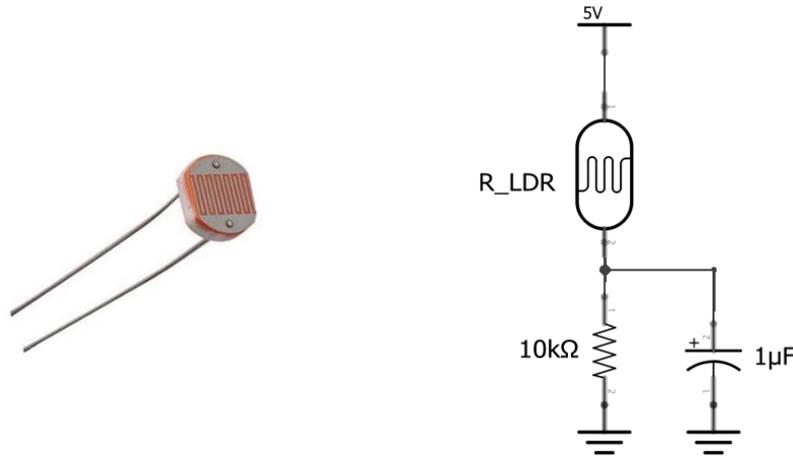


Figura 14 – LDR y Esquema de montaje del sensor

2.3.8. Módulo Wifi ESP8266

En 2014 apareció el microcontrolador *ESP8266* de la mano de la empresa china *Espressif*. Este chip Wifi de bajo coste incorpora una pila TCP/IP y suele venir encapsulado en diferentes formatos con todos los componentes necesarios además del chip. La primera versión, el *ESP-01*, es el encapsulado más sencillo y el modelo escogido para este proyecto, su objetivo principal es el de dotar de conectividad Wifi cualquier sistema compatible. De esta manera podremos comunicarnos mediante Wifi con otros dispositivos lo que permite crear la aplicación móvil para el control del microhuerto.

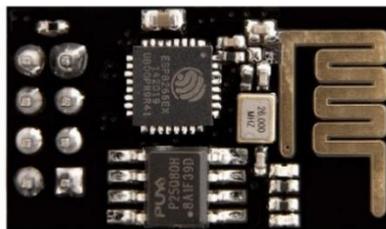


Figura 15 – Módulo Wifi ESP8266 ESP-01. [Imagen]. Recuperado de <https://www.electronilab.co>. (2020)

Aunque su origen fue como complemento para las placas de Arduino, a medida que fue aumentando el uso de las placas de desarrollo, este chip se popularizó hasta el punto de que hoy en día, se usan en módulos adaptados que son capaces de realizar las mismas tareas que un Arduino u otras placas de desarrollo, y hoy es su principal competidor. A continuación, se detallan sus características más relevantes:

- Voltaje operativo: 2,5 – 3,6 VDC.
- Corriente media de trabajo: 80 mA.
- Corriente máxima: Hasta 250 mA en arranques y transmisiones largas.
- Rango de frecuencia 2,4 – 2,5 GHz.
- Protocolos Wi-Fi: 802.11 b/g/n.
- Protocolo de comunicación serie: UART con 2 pines dedicados.
- Pines de propósito general (GPIO): 2.
- Procesador: *Tensilica L106* de 32 bits a 80 MHz.
- Memoria flash: 512 KB.

Como se observa en sus características, el módulo funciona a un voltaje de 3,3 VDC con un máximo de 3,6 V, por tanto, se podría alimentar mediante el pin de 3,3 V de la placa, el problema es la corriente de salida, limitada a 50 mA en este pin. Como se aprecia, el módulo tiene un consumo de corriente máximo de 250 mA, por lo que es menester una fuente de alimentación externa que nos proporcione el voltaje y amperaje requeridos.

Para solventar este inconveniente se adquirió una fuente de alimentación para *protoboard*, disponible en tiendas de electrónica. La ventaja de este dispositivo es que dispone de dos canales de salida individuales de voltaje seleccionable, mediante interruptor físico, a 3,3 o 5 VDC, lo que permite alimentar componentes que operen con cualquiera de ambos voltajes de forma simultánea.



Figura 16 – Fuente de alimentación para protoboard. [Imagen]. Recuperado de www.geekbotelectronics.com. (2020)

Seguidamente se describen sus características más relevantes:

- Voltaje de entrada por clavija tipo Jack: 7 – 12 VDC.
- Voltaje de entrada por clavija microUSB: 5 VDC.
- Voltaje de salida: Doble canal seleccionable a 3,3 o 5 VDC.
- Corriente máxima de salida: 700 mA.

3. FUNCIONAMIENTO DE LOS COMPONENTES

3.1. Módulo de riego

Por módulo de riego se entiende el conjunto de dispositivos encargados del sistema de riego y el depósito, estos son: el módulo de ultrasonidos, el sensor de nivel, el relé y la bomba. Esta prueba se realizó en conjunto ya que de los componentes mencionados el único que requiere una calibración previa es el sensor ultrasónico. De esta manera se agiliza el proceso y permite realizar una simulación más fiel a la realidad.

3.1.1. Calibración del módulo de ultrasonidos HC-SR04

Para su calibración se realizó una sencilla prueba de medición con una plancha de madera de superficie adecuada, según el fabricante, de al menos 0,5 m² para evitar errores.

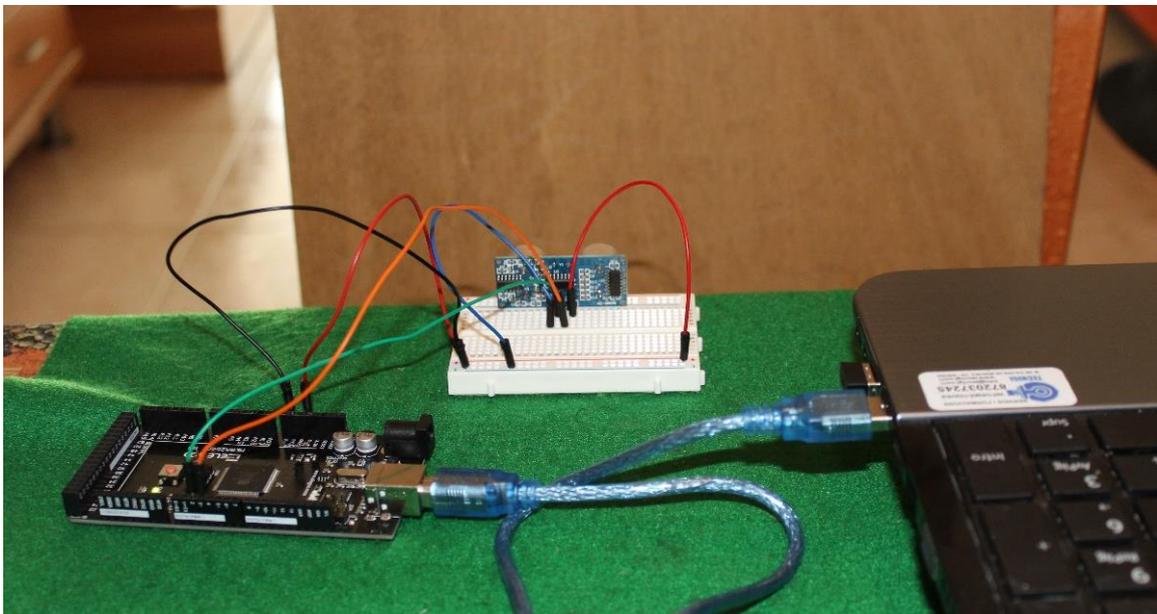


Figura 17 – Montaje para la prueba de calibración del módulo HC-SR04

Tal como se ha explicado anteriormente, la placa solo mide el tiempo que tarda la señal en rebotar en el objeto y volver, el cálculo de la distancia se realiza manualmente conociendo la velocidad del sonido. La velocidad del sonido depende principalmente del medio de transmisión y de la temperatura, en el aire se puede calcular según la siguiente expresión:

$$V_s = c + (0,6 \cdot T)$$

Donde c es la velocidad del sonido a 0°C (331,5 m/s) y T es la temperatura del aire en °C.

Aunque este cálculo se puede ver ligeramente afectado por otras condiciones como la humedad y la presión, en condiciones estables se obtienen buenos resultados. La prueba se realizó a una temperatura de 29,8°C, correspondiendo a una velocidad de 349 m/s aprox., con una humedad del 55% y a una altura de 80 metros sobre el nivel del mar. Se realizaron pruebas tanto alejando como acercando el objeto a medir y una prueba de repetibilidad. A continuación, se observan los resultados obtenidos:

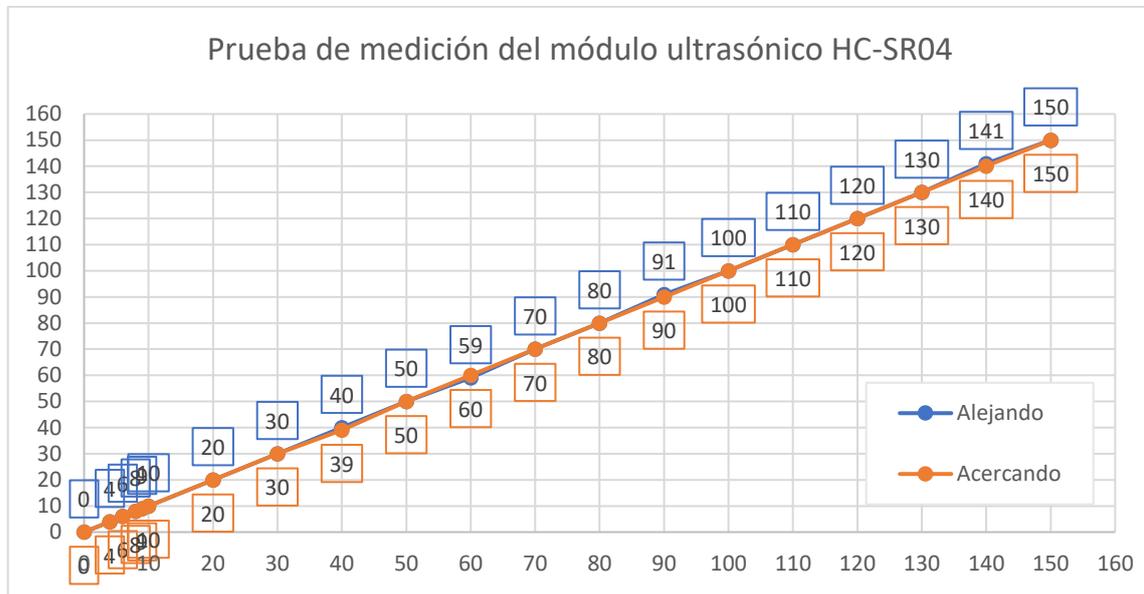


Figura 18 – Gráfica de los resultados obtenidos en la prueba del módulo HC-SR04

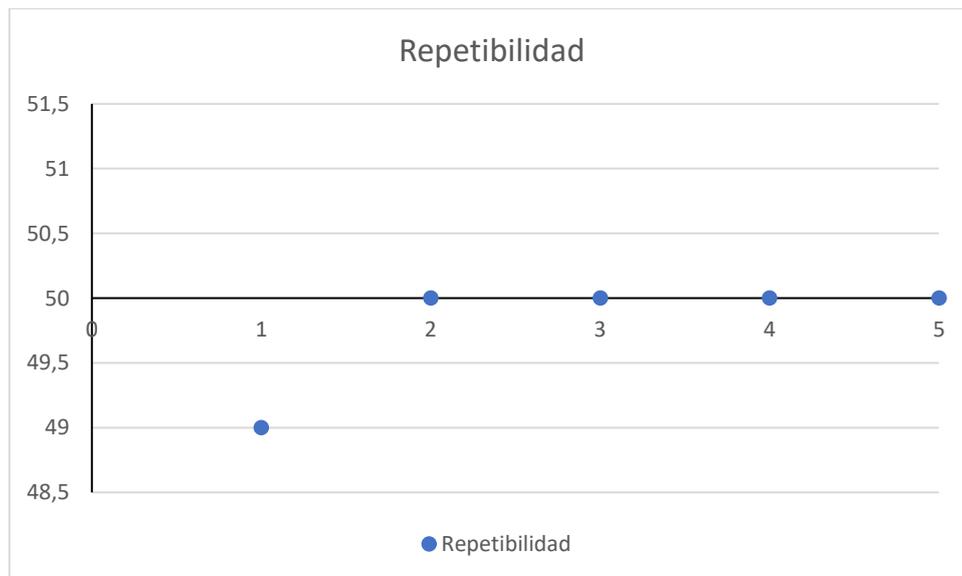
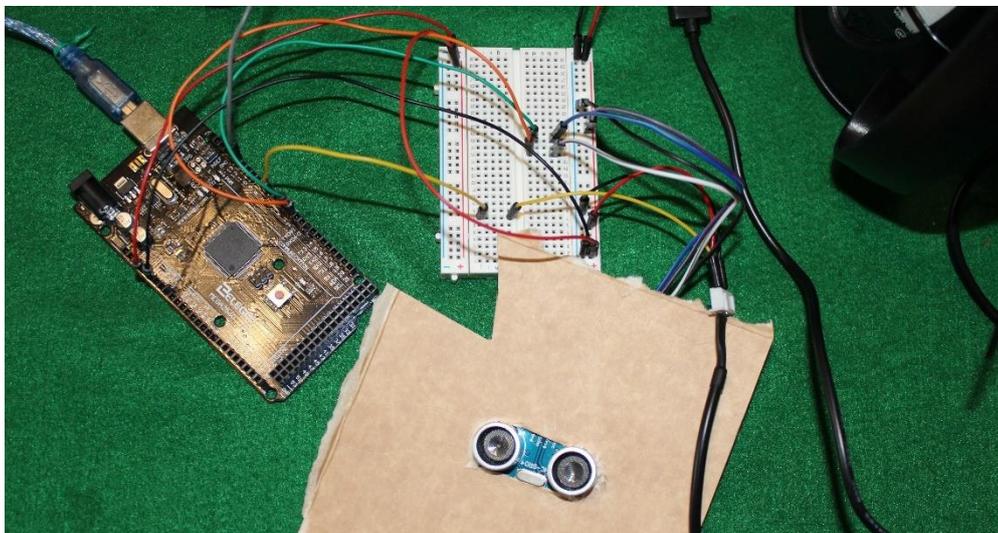


Figura 19 – Gráfica de repetibilidad en la medición con el módulo HC-SR04

Pese a que esta prueba no requirió la máxima precisión del sensor, 3 mm según el fabricante, los resultados fueron muy buenos con solo 4 medidas imprecisas y un error prácticamente nulo, 2,5% en el peor caso, además de una repetibilidad muy alta, como se puede observar.

3.1.2. Simulación del sistema de riego

Una vez calibrado el sensor de ultrasonidos, se realizó la prueba del conjunto. Como depósito se usaron recipientes de diferentes materiales, plástico, cristal y metal para comprobar si el material podía afectar al funcionamiento de los sensores. Por un lado, el sensor capacitivo se fijó al depósito mediante bridas a la medida deseada, por otro, el sensor de ultrasonidos se fijó, en una pieza de cartón, a modo de tapa en el borde superior. Por último, una señal enviada desde el ordenador activaba el relé que activaba la alimentación de la bomba, un cargador de móvil puentado con el relé y la bomba mediante un adaptador USB hembra.



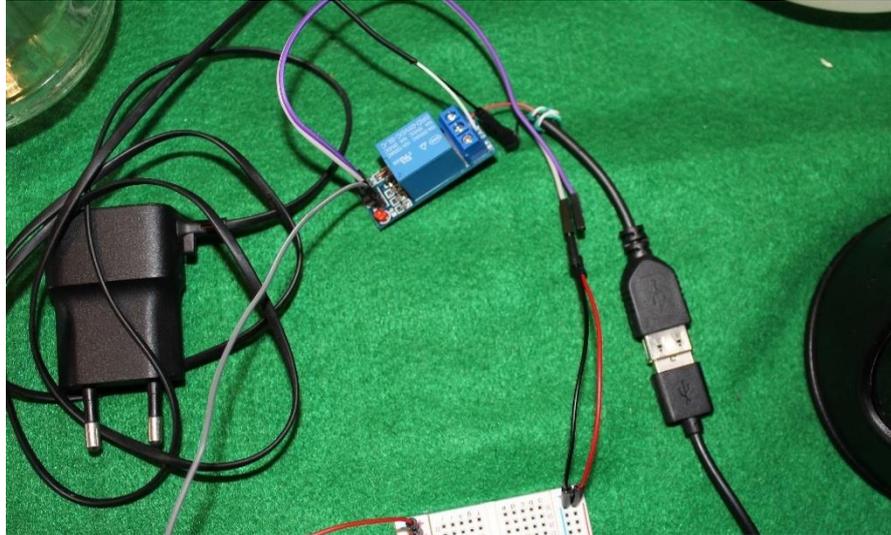


Figura 20.a, 20.b y 20.c – Imágenes del montaje para la prueba del sistema de riego y detalles de los componentes y sus conexiones

Para la prueba se fijaba el centro del sensor capacitivo a una distancia determinada, la misma distancia a la que se programaba la señal de paro en el módulo ultrasónico. Se realizaron varios vaciados, comprobando que en todos los intentos ambos sensores dieran la señal de paro. En la siguiente tabla se detalla en qué momento mandaban la señal y el tiempo que desfasaba un sensor respecto el primero que enviaba la señal de paro:

		Plástico transparente			Plástico opaco		
		Med. 1	Med. 2	Med. 3	Med. 1	Med. 2	Med. 3
HC-SR04	Medida final (cm)	11,2	11,2	11,1	12,2	12,3	12,3
	Desfase (seg)	2	3	1	1	2	2
XKC-Y25-V	Medida final (cm)	11	11	11	12	12	12
	Desfase (seg)	0	0	0	0	0	0

		Cristal			Metal		
		Med. 1	Med. 2	Med. 3	Med. 1	Med. 2	Med. 3
HC-SR04	Medida final (cm)	10,5	10	10,3	6,3	6,3	6,4
	Desfase (seg)	2	0	1	-	-	-
XKC-Y25-V	Medida final (cm)	10	10	10			
	Desfase (seg)	0	0	0			

Tabla 2.a y 2.b – Tabla de resultados de la prueba del sistema de riego

Como se puede apreciar, ambos sensores funcionaron correctamente enviando la señal de paro en todos los casos, aunque el sensor capacitivo XKC-Y25-V fue más preciso y con una mayor velocidad de respuesta. También se pudo comprobar cómo, en efecto, el sensor capacitivo no puede trabajar con un depósito metálico ya que siempre está activo, y cómo el sensor de ultrasonidos, aparentemente, no se ve afectado por el material del recipiente.

Más allá de los resultados, el sensor capacitivo también ofrece mayor facilidad de montaje y no requiere una calibración previa. Por otro lado, el módulo HC-SR04 requiere una recalibración periódica según la variación de temperatura y no está especialmente indicado para trabajos en entornos húmedos. Es por todo ello que, aunque ambos sensores arrojaron buenos resultados y se instalaron en el prototipo, en la parte de control se le dará prioridad al sensor XKC-Y25-V tal como se detalla en siguientes apartados de esta memoria.

3.2. Módulos higrómetro v1.0 y v1.2

Para la comprobación y calibración de estos sensores se realizaron varias pruebas en diferentes entornos para ver como respondían y observar sus diferencias. El higrómetro v1.0 (resistivo) no requiere calibración previa, una vez conectado a la alimentación arrojará diferentes valores a través de la entrada analógica según el entorno en el que se encuentra, valores cercanos a 1023 en aire y a 0 en agua. En cuanto al higrómetro v1.2 (capacitivo) sí requiere de calibración, por lo que mediante la prueba de funcionamiento se obtuvieron los umbrales inferior y superior del sensor.

La prueba consistió en introducir ambos sensores en diferentes entornos, partiendo siempre desde un valor estable en aire, y observar las medidas obtenidas por cada sensor. Estos entornos fueron, aire, agua, agua con sal, tierra seca y tierra recién humedecida. La prueba en agua con sal se realizó con la finalidad de observar cómo afectaba a las mediciones de los sensores ya que en el suelo de cultivo la presencia de sales es inevitable. Se realizaron tres medidas por cada sensor y entorno, siendo estos los resultados obtenidos:

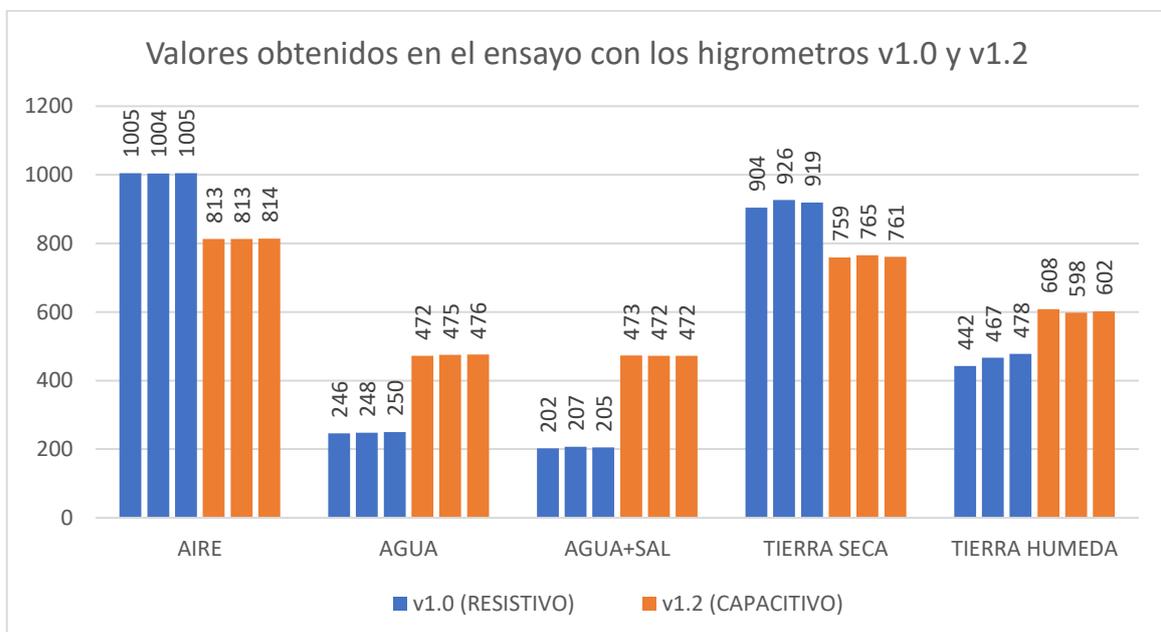
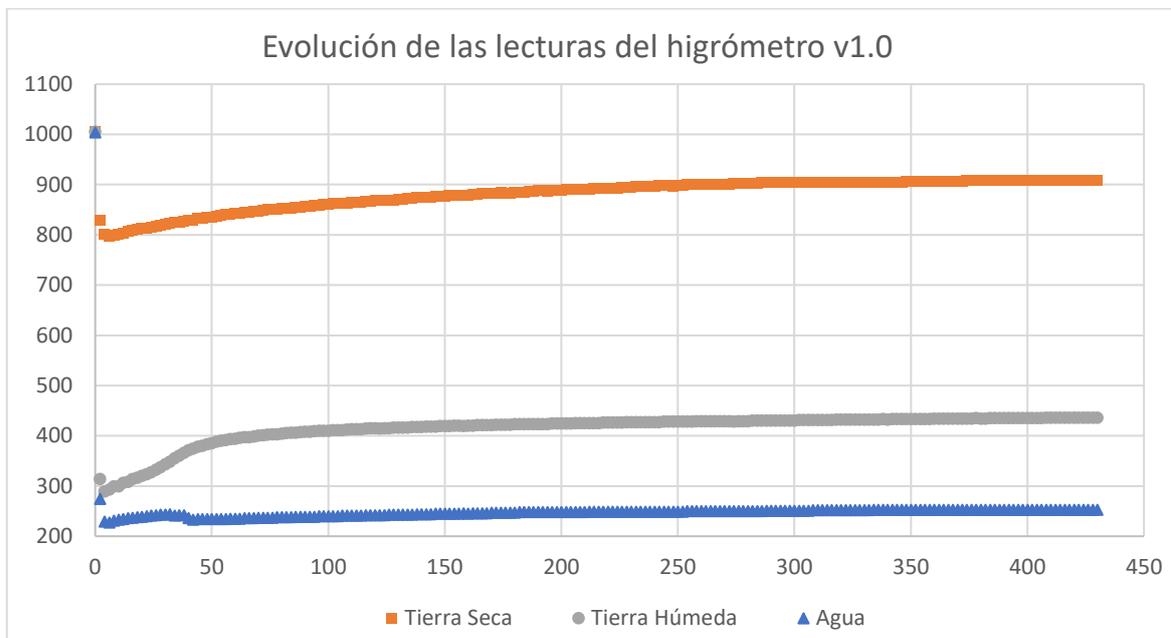


Figura 21 – Gráfica de los valores obtenidos en el ensayo de medición

Como se pudo comprobar, el higrómetro resistivo (v1.0) devuelve unos valores correctos en aire, cercanos al valor máximo de 1023, mientras que para el agua (100% de humedad) no se acercaba al valor de 0 como se esperaba pese a estar calibrado de fábrica, por lo que este nuevo valor se fijó como referencia para la conversión a % de humedad. También se observó como su lectura se ve altamente afectada por la sal, en cambio, el higrómetro capacitivo es más robusto y sus lecturas no se vieron afectadas por la presencia de esta, dando unos valores prácticamente idénticos al recipiente con agua normal. Esta prueba también permitió conocer los valores umbral del sensor capacitivo (v1.2) y realizar un *mapeado* del rango de medición, fijando así valores de: 813 para el aire (0%) y 473 para el agua (100%).

Otra característica observable en los datos es la estabilidad y repetibilidad de las lecturas, el sensor capacitivo devuelve valores más estables mientras que el sensor resistivo es menos preciso y arroja valores más dispares en general. Traducido en datos, es posible observar esta diferencia escogiendo el peor caso para cada dispositivo, es decir, para el v1.0 en tierra húmeda, se aprecia una diferencia entre valores de hasta 36 puntos lo que respecto a su fondo de escala aire-agua (1005-248) supone una diferencia del 4,8%. Mientras que para el v1.2, también en tierra húmeda, la máxima diferencia es de 10 puntos lo que en su fondo de escala (813-473) se traduce en una diferencia del 2,9%.

Durante el mismo ensayo se monitorearon las evoluciones de algunas lecturas con tal de observar el tiempo de estabilización de ambos sensores, estos fueron los resultados:



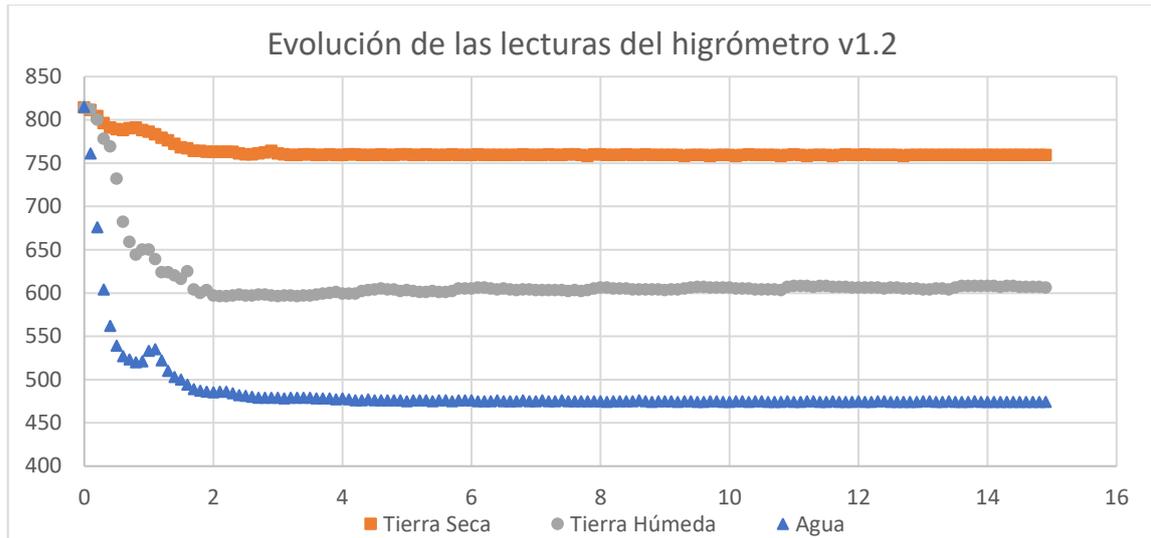


Figura 22.a y 22.b – Gráfica de la evolución de las lecturas de los higrómetros v1.0 y v1.2

Tal como se observa, el sensor resistivo dibuja una forma más subamortiguada con una tendencia al origen infinita, puesto que en las pruebas nunca llegaba a dar una lectura estable y, pasados unos minutos, se detenía la prueba para proteger al sensor de la corrosión. Aun así, usando el último valor como referencia, el tiempo de estabilización a 4τ (98%) es de 146 segundos para el agua, 228 para la tierra húmeda y 202 para la tierra seca.

En cambio, el sensor capacitivo dibuja una forma más sobreamortiguada ofreciendo un valor estable y de manera mucho más rápida en comparación al primero: 2 segundos para el agua, 1,4 para la tierra húmeda y 1,4 para la tierra seca.

Después de la lectura de los datos se puede concluir que la literatura hallada antes de la prueba era correcta, pudiendo afirmar así que los higrómetros capacitivos son ampliamente mejores que los resistivos en todos los aspectos, estabilidad, tiempo de respuesta, robustez y sobre todo la vida útil ya que la sensibilidad a la corrosión de este último lo hace poco eficiente para su aplicación. Así pues, ambos sensores se instalaron, aunque se le dió prioridad al sensor capacitivo delante del resistivo en la toma de decisiones para el correcto funcionamiento del sistema.

3.3. Sonda DS18B20 y Sensor DHT11

Para comprobar el funcionamiento de ambos sensores, se realizó una sencilla prueba que consistió en medir la temperatura a lo largo del día durante varias horas por dos días consecutivos, así como la humedad relativa del aire para verificar el correcto funcionamiento del DHT11. Durante las horas centrales del día no se tomaron datos ya que la variación de las condiciones de temperatura y humedad eran mínimas.

Para este ensayo las temperaturas fueron tomadas tanto con el sensor DHT11 como con la sonda DS18B20, en concreto dos sondas ya que es la unidad mínima que se puede adquirir de este dispositivo. Puesto que los datos de ambas sondas eran prácticamente idénticos se realizó la comparativa únicamente con la media de ambas, mostrando así de forma más clara los datos obtenidos. Para la comprobación de estas medidas se usó un termómetro digital comercial y uno de mercurio. Estos fueron los resultados obtenidos en el ensayo:

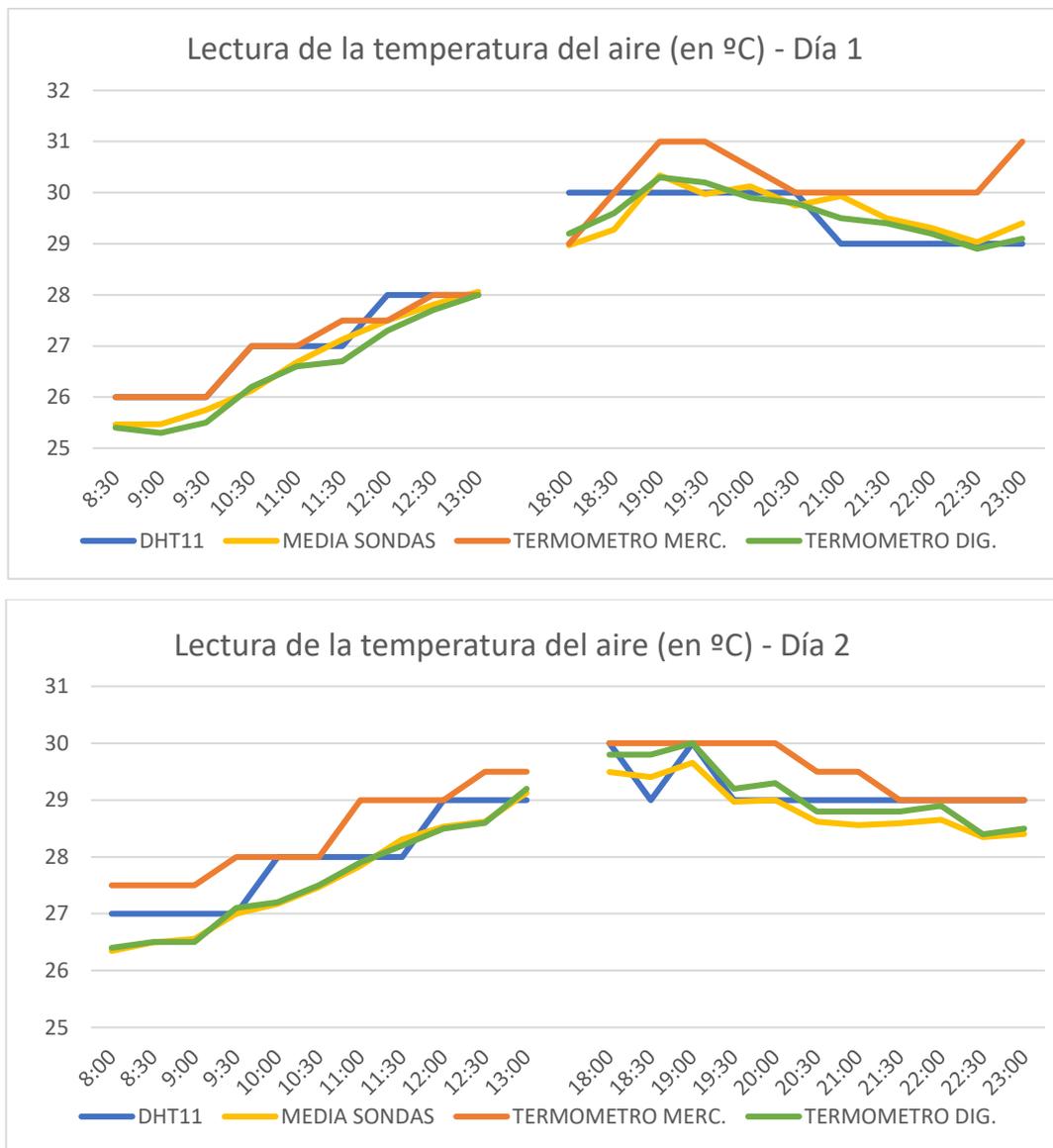


Figura 23.a y 23.b – Gráfica de las lecturas de temperatura (°C)

Como se puede observar, la sonda DS18B20 devolvió valores muy precisos y dibuja una forma muy similar a la del termómetro digital con variaciones parecidas. Se pudo constatar, tal como se aprecia en las características del dispositivo, que es un sensor excelente y muy completo, más teniendo en cuenta su coste. Por otro lado, el DHT11 sigue aparentemente la forma de los dos anteriores, pero con variaciones más espaciadas y bruscas a causa de su

baja precisión y resolución, una función más parecida a la obtenida con el termómetro de mercurio.

Observando las características de estos dispositivos era esperado un mejor funcionamiento de la sonda DS18B20 respecto del DHT11 pero, tal y como se ha comentado anteriormente, la información obtenida por el DHT11 es meramente informativa por lo cual, en esta etapa de prototipado, los resultados obtenidos fueron satisfactorios y aptos para su aplicación.

Continuando con este último sensor, el DHT11 también nos aporta información sobre la humedad relativa del aire. Este dato fue comprobado a la par con las lecturas de temperatura, siendo este verificado con el higrómetro incorporado en el sensor digital.

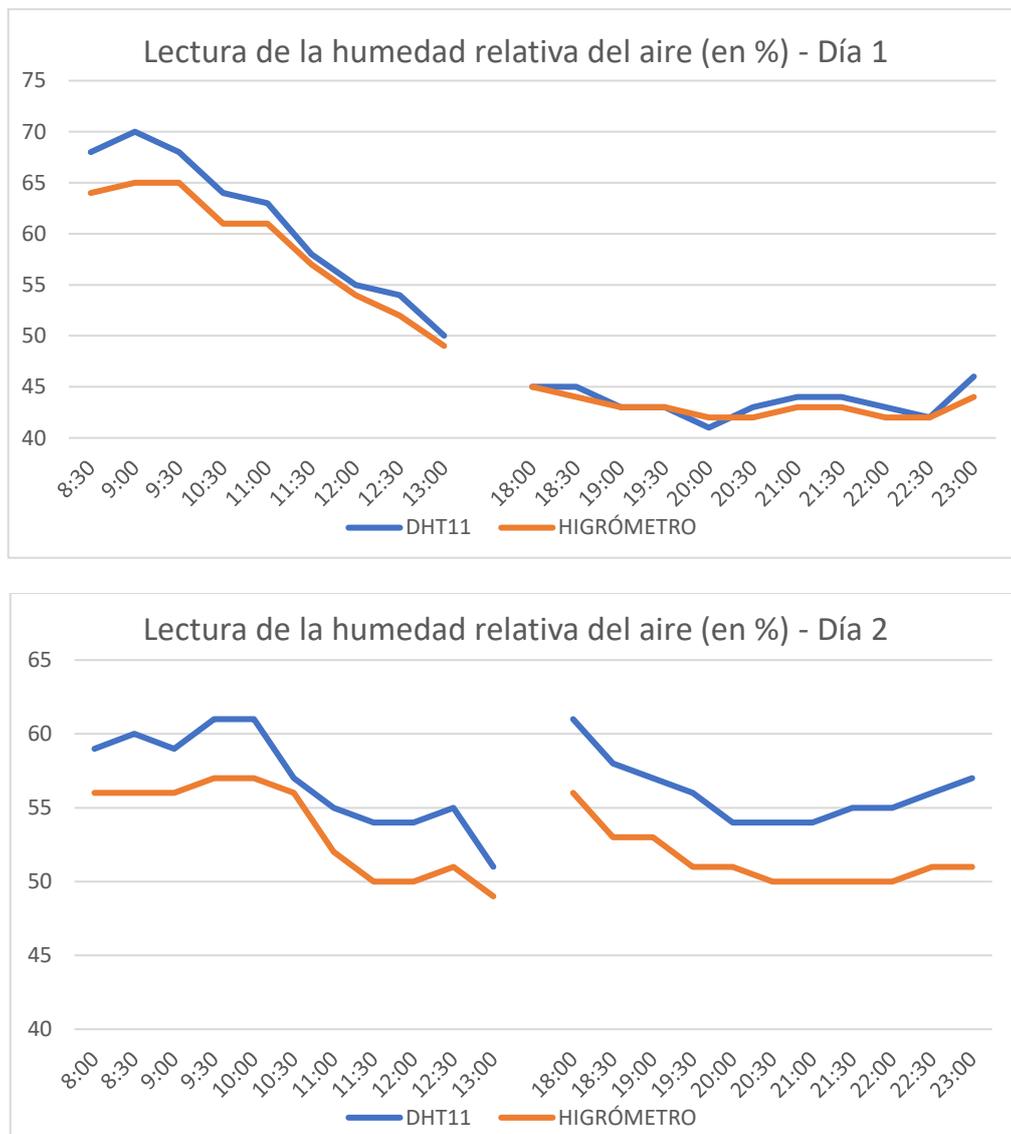


Figura 24.a y 24.b – Gráfica de las lecturas de humedad relativa del aire (%)

Como anticipaba la ficha técnica, la precisión en la medición de humedad es aún inferior que en la medición de temperatura lo que, en efecto, se traduce en un error considerable en las medidas, tal como se observa en las gráficas superiores. Se aprecia como el DHT11 detecta las variaciones de humedad de forma similar al higrómetro comercial, aunque de forma más abrupta y con un error de hasta el 5% en algunos casos. Pese a su imprecisión, los resultados son coherentes y están dentro del margen de error indicado por el fabricante por lo que se confirma así su correcto funcionamiento, aunque este se ve claramente limitado por sus características técnicas. Una vez más, esta baja precisión no es un inconveniente puesto que esta información también será informativa para el usuario.

3.4. Calibración del LDR

Cabe recordar que el sensor de luminosidad se basa en una fotorresistencia o LDR dispuesta a modo de divisor de tensión, tal como veíamos en la figura 14, lo que permite obtener el valor de su resistencia en cada momento, el cual varía en función de la luminosidad ambiente.

El valor de la resistencia del LDR se obtiene mediante la expresión característica de los divisores de tensión y conociendo la relación entre la entrada analógica y el voltaje real:

$$V_o = V_{in} \cdot \frac{R}{R + R_{LDR}} \quad \text{conociendo: } V_o = \frac{A_{val} \cdot V_{in}}{1023} \rightarrow R_{LDR} = \frac{1023}{A_{0val}} \cdot R - R$$

Donde:

- V_o es el voltaje de salida del divisor de tensión.
- V_{in} es el voltaje de entrada del divisor de tensión, 5 V provenientes de la placa.
- R es el valor de la resistencia fija, típicamente 10 K Ω .
- A_{val} es el valor leído en la entrada analógica (0 a 1023).
- R_{LDR} es el valor de la resistencia del LDR.

Una vez obtenido el valor de la resistencia, es posible traducir este valor a lux mediante la relación logarítmica que nos propone el fabricante en su ficha técnica.

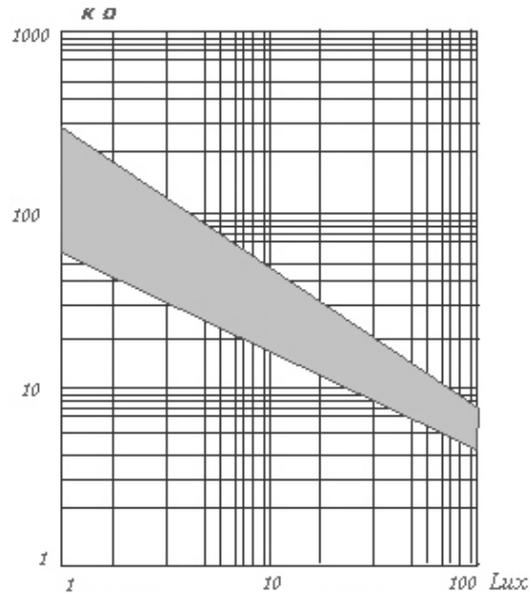


Figura 25 – Gráfica lux-resistencia del LDR. [Gráfica]. Recuperado de GL55 Series CdS Photoresistor Manual (Datashheet). (2016)

Como se aprecia en la gráfica, a escala logarítmica, la relación entre lux y resistencia es lineal, por lo que se puede expresar según la siguiente función:

$$\log(R_{LDR}) = m \cdot \log(L) + b$$

Dónde:

- m es la pendiente de la recta,
- b es el punto de corte con el eje vertical,
- L es el valor de la luminosidad en lux.

Escogiendo dos puntos medios dentro del espectro de la gráfica se obtiene que:

- Punto 1: 2 lux → 100 KΩ
- Punto 2: 70 lux → 6 KΩ
- m = -0,7913
- b = 2,2382

Gracias a esta relación se obtiene directamente la luminosidad ambiente en lux a partir de la lectura analógica del divisor de tensión. Para comprobar los valores que arrojaba el sensor construido se realizaron dos pruebas: En la primera, se introdujo el sensor en una caja opaca mientras se aumentaba la intensidad de la luz progresivamente mediante un pequeño LED conectado a una de las salidas PWM de la placa, estos fueron los resultados obtenidos:

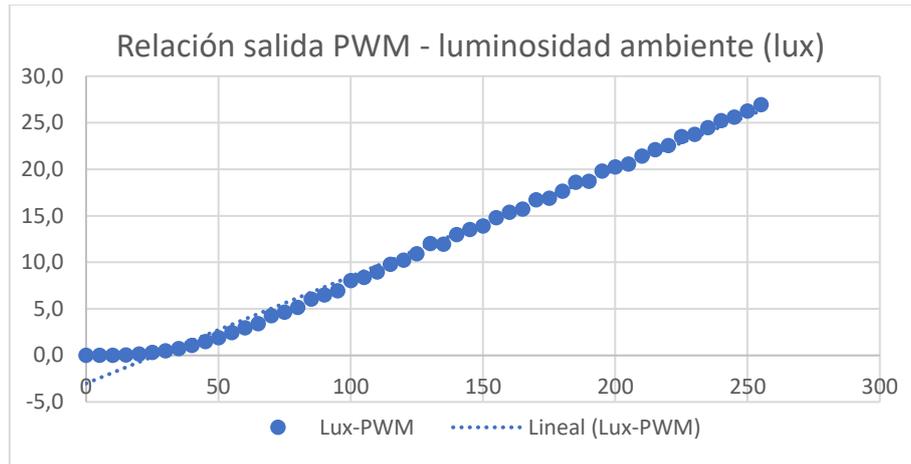


Figura 26 – Gráfica de la relación salida PWM - lux

Como se observa en la gráfica superior, se obtiene una relación lineal entre el valor de la salida PWM y la luminosidad ambiente (en este caso dentro de la caja), es decir, el aumento de la intensidad lumínica de la bombilla es directamente proporcional a la luminosidad ambiente que indica el sensor, lo que apunta a un correcto funcionamiento. También, los valores obtenidos fueron bastante coherentes ya que, tratándose de un pequeño LED, para el máximo valor de la salida PWM (5V) el sensor devolvió un valor de unos 27 lux.

Debido a la imposibilidad de obtener un luxómetro certificado que asegurara la verificación de las medidas, para la segunda prueba se utilizaron los sensores de luz que incorporan los smartphones actuales, cuya principal aplicación es regular el brillo de la pantalla según la luz ambiente. Se tomaron 20 medidas con el sensor y con dos smartphones diferentes, un *Samsung S9+* y un *Xiaomi Mi8*, y se reguló la entrada de luz natural a través del ventanal de la sala donde se recogieron las medidas. A continuación, se muestran las lecturas obtenidas por los diferentes dispositivos:

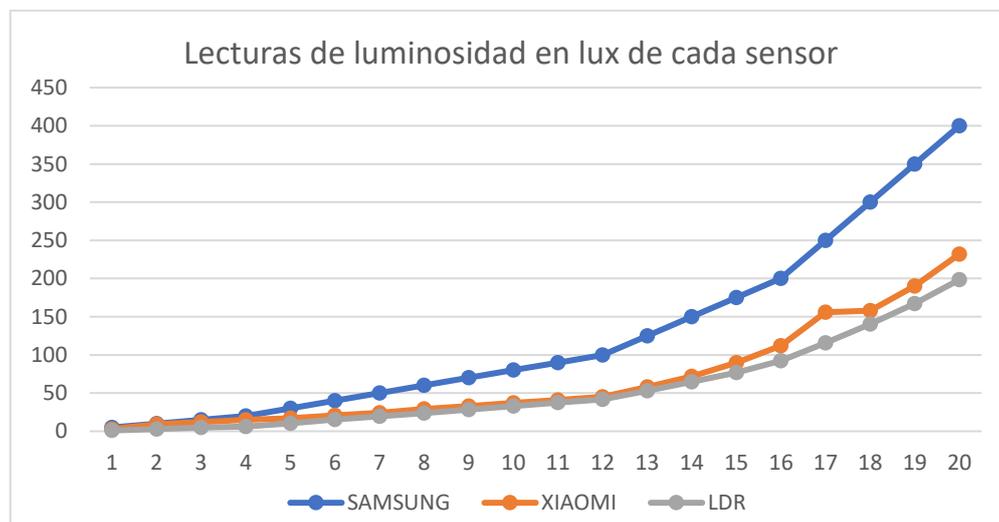


Figura 27 – Gráfica de las lecturas de luminosidad tomadas por cada sensor

Es posible apreciar que, pese a que no se pueden dar como exactas las medidas devueltas por los sensores de los smartphones, el sensor basado en el LDR arroja valores muy similares al sensor del smartphone Xiaomi y se alejan bastante del sensor del Samsung S9+. Como se ha mencionado, no se puede corroborar que ningún dispositivo sea más exacto que otro, pero sí que el LDR tiene un correcto funcionamiento dentro de sus limitaciones.

Debido a que no es necesario el valor exacto de lux ambiente, con esta comprobación es suficiente para conocer el funcionamiento de este sensor y poder discernir las horas de mayor sol para evitar la activación del riego durante ellas.

3.5. Calibración del anemómetro

Una vez construido el anemómetro, era preciso realizar un *mapeo* del motor de CC viendo que valores de voltaje devolvía según la velocidad del viento. Previamente a la toma de datos se modificó la referencia de voltaje para la conversión analógica – digital, puesto que la referencia por defecto es de 5 V y nuestro motor es de 2,4 V. Las referencias configurables para esta placa son de 1,1 o 2,56 V, se escogió la referencia de 1,1 debido que son motores de elevadas revoluciones y era muy improbable que el motor llegase a dar valores más allá del voltio. Así pues, con este ajuste y, sabiendo que el conversor ADC (Analog to Digital Converter) disponible es de 10 bits tal y como se ha comentado en la descripción de la placa, se pasó de una resolución de 4,89 mV a 1,08 mV.

Para poder variar a voluntad la velocidad del viento, se decidió fijar el dispositivo al techo de un automóvil aprovechando la varilla descrita anteriormente (ver apartado 2.3.7), y colocando un imán de neodimio en su extremo.



Figura 28.a y 28.b – Anemómetro fijado al techo (izquierda) y conexión con la placa de desarrollo y el ordenador (derecha)

Durante la prueba se tomaron cuantas medidas fueron posibles y, una vez concluida la prueba, se filtraron los resultados eliminando todas las mediciones incorrectas o valores claramente desviados. Después del filtraje de datos, los valores restantes eran algo dispares debido a la alta resolución del conversor, para encontrar un valor representativo de cada velocidad se realizó un promedio del valor de las lecturas tomadas para esa misma velocidad. Una vez encontrados estos valores, se tradujeron a valores de voltaje con el propósito de encontrar una función que relacionara directamente el voltaje del motor y la velocidad del viento, con independencia del conversor ADC que usamos. Seguidamente se muestran los resultados obtenidos:

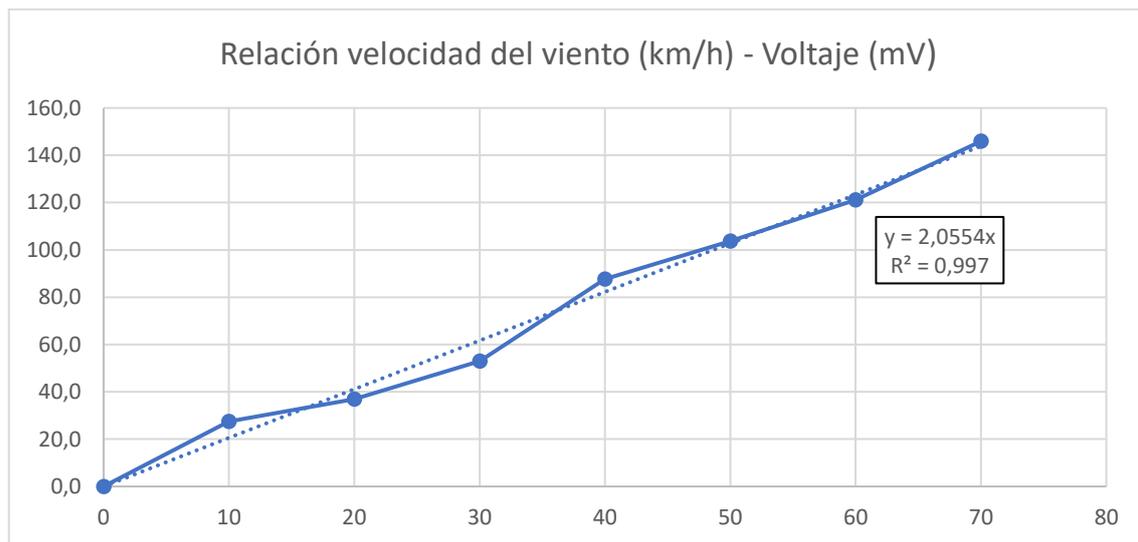


Figura 29 – Gráfica velocidad – voltaje del anemómetro

Observando la gráfica superior, pese a su dispersión, se aprecia una clara tendencia lineal. Así pues, mediante el método de mínimos cuadrados y forzando su paso por 0,0, puesto que a velocidad cero el motor no puede dar ningún valor de voltaje, se obtiene una función lineal cuya expresión es $V_m(v) = 2,055v$ (Dónde V_m es el voltaje generado por el motor y v es la velocidad del viento) con un coeficiente de correlación lineal muy alto, cercano a 1. De esta manera, haciendo la relación inversa de la función, se obtiene una ganancia que relaciona directamente el valor devuelto por el anemómetro con la velocidad real del viento.

4. SISTEMA DE CONTROL

4.1. Flow Chart

Se encuentra en el Anexo II de esta memoria un *flow chart*, o diagrama de flujo en español, donde se muestra de manera gráfica y simplificada el funcionamiento del sistema de control, el cual se traduce posteriormente en un *script*, o código de programa, donde se recogen todas las instrucciones que realizará el procesador o microcontrolador en este caso.

Como se observa, primero se declaran todas las variables de datos y se inician las comunicaciones y librerías necesarias para el uso de funciones específicas. Seguidamente se realiza la primera lectura de datos de todos los sensores y se inicia el *timer* 1 responsable de dar la alerta para las tomas de datos consiguientes.

A partir de este punto empieza el bucle de control. Primeramente, se comprueba si el timer 1 ha desbordado, es decir, si se ha cumplido el tiempo establecido, en caso afirmativo se realiza la toma de datos de todos los sensores, en caso contrario se salta al siguiente punto.

El siguiente punto es comprobar si se ha activado la *bandera* de riego automático (“Autoriego”), es decir, si según la última lectura de los sensores es necesario regar. Si es así, pasamos al proceso descrito en la parte derecha, se activa la bomba de riego y se inicia un segundo timer el cual controlará el tiempo de riego. Este tiempo de riego se estimó según el siguiente cálculo:

$$\frac{\text{volumen de agua necesario}}{\text{caudal gotero} \cdot \text{celdas de cultivo}} = \frac{3l}{4 \frac{l}{h} \cdot 9} = \frac{1}{12} h \cdot \frac{60 \text{ min}}{1 h} = 5 \text{ min}$$

El volumen de agua necesario depende de varios factores los cuales no son objetos de este proyecto, para el tamaño de nuestro suelo de cultivo, 3 litros es un volumen óptimo. El número de celdas de cultivo corresponde al número de celdas de 30x30 que constituyen el bancal, tal como se detalla en el diseño del prototipo en el punto 2.1 de esta memoria.

Mientras el timer 2 no desborde solo se comprobará el estado del timer y el nivel de agua del depósito. Una vez finalice el tiempo de riego, o el depósito se quede sin agua, se apagará la bomba, el timer 2 pasará a 0 de nuevo y se activará la bandera de riego finalizado (“riegoFlag”). Esta bandera evita que el programa vuelva a comprobar si es necesario activar el riego hasta que no haya una nueva lectura y se actualicen las variables, de esta manera se asegura no activar el riego 2 veces con la misma lectura.

Por otro lado, si no está activa la bandera del riego automático se comprueba si el usuario ha activado el modo manual. En caso afirmativo se comprueba si el usuario desea encender la bomba, también en caso en afirmativo se activa la bomba mediante el relé y quedará activa mientras el usuario no desee desactivarla o el nivel de agua baje del umbral mínimo. Para evitar activar o desactivar la salida del relé de forma repetitiva se añade una bandera de cambio de estado, de manera que solo cambia el estado de la salida cuando realmente es necesario. En el momento en que la bomba se desactive, bien por comando o bien por falta de agua en el depósito, se activa la misma bandera definida en el riego automático, así, en caso de que el usuario realice el cambio a modo automático, se asegura que la decisión de activar el riego automático se tome con las variables actualizadas.

El último caso que podemos observar es el modo automático (internamente que el modo manual no esté activado) donde se comprueba si se cumplen las condiciones para activar el riego automático. Estas son que el nivel de agua sea el correcto, que no se esté en horas máximas de sol y que la humedad del suelo de cultivo esté por debajo del 45%. Una vez más, el estudio del % de humedad óptimo depende de muchos factores que no son objeto de estudio en este proyecto. En términos generales se aconseja una humedad de entre el 50 y el 60%, por ello se estableció el límite mínimo en un 45% de humedad. Se añade un 5% de margen del cuál un 3% corresponde al error propio del higrómetro tal como se menciona en el apartado 3.2 de esta memoria.

Cuál sea de los 3 procesos descritos que esté activo en cada momento, el bucle siempre finaliza en el *Buffer Serial*. Normalmente este sería el medio de comunicación con un ordenador para enviar y recibir información de la placa, en este caso se sustituye por el módulo Wifi ya mencionado. En este punto se envía el valor actualizado de todas las variables y el estado del sistema y se reciben los comandos enviados por el usuario a través de la interfaz de usuario o App. Aquí se actualizan las banderas de funcionamiento escogido por el usuario (modo manual y activación de la bomba).

4.1.1. Sample Time de la lectura de datos

Puesto que los sensores del depósito aportan información de seguridad para el buen funcionamiento, y se activan independientemente de la toma de datos, el siguiente valor capital es el higrómetro, puesto que es el principal elemento que determina el riego además del luxómetro, el cual se considera un elemento de mejora de la eficacia del sistema.

Para determinar el tiempo de muestreo en la lectura de los sensores (*sample time*) se decidió monitorear la humedad del suelo de cultivo durante varias horas. Al ser un proceso lento se monitoreó, al mismo tiempo, la temperatura del suelo. Aunque no es un valor determinante se podrían perder cambios de temperatura por dilatar en exceso el *sample time*.

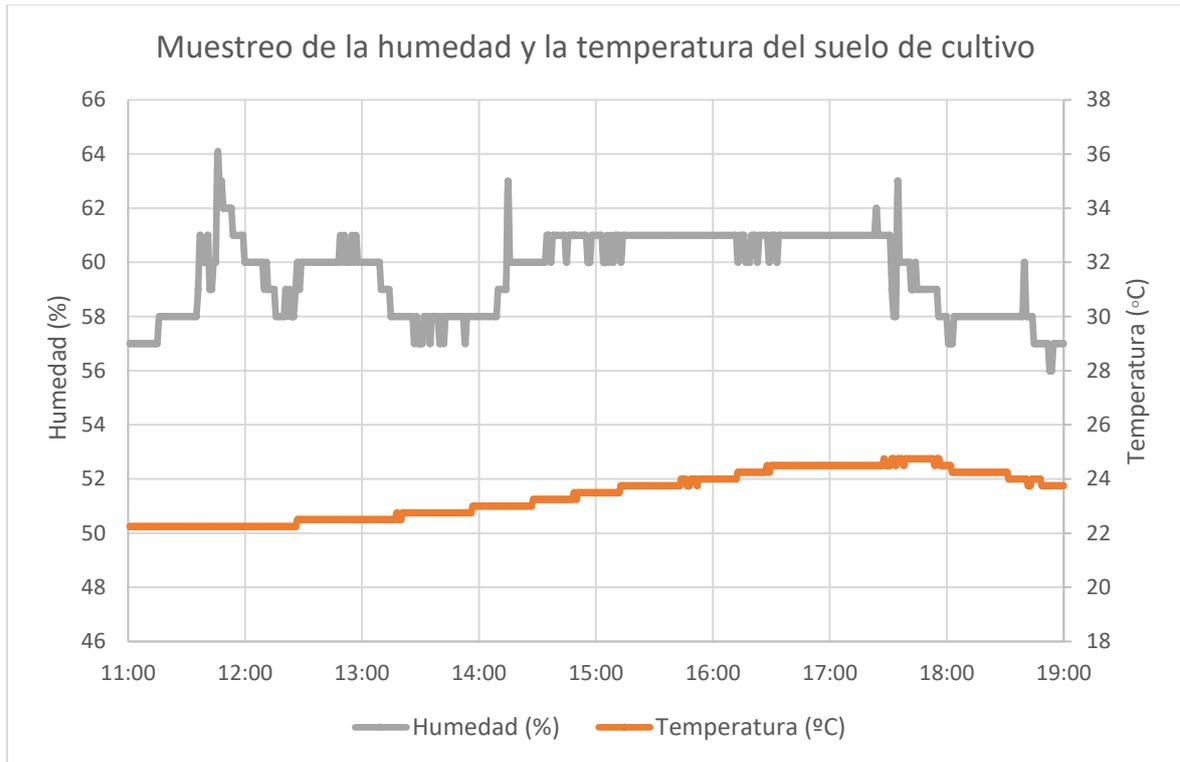


Figura 30 – Gráfica del muestreo de la humedad y la temperatura del terreno

Se tomaron muestras de ambas variables a cada minuto y se puede observar su evolución en la gráfica superior. Como se aprecia, en las mismas condiciones, la temperatura varía de forma mucho más lenta y suave, tardando en variar, de media, más de 1 hora por grado de temperatura. Aunque esta muestra no es representativa del comportamiento real, se observa que la temperatura, por lo general, es un proceso más lento, por lo que al determinar el tiempo de muestreo según la variación de humedad no sesgaría la adquisición de los cambios de temperatura.

Con ello, se recogen en la siguiente tabla los cambios más significativos en la evolución de la humedad:

Cambio (3%)	Hora Inicio	Hora Final	Lapso (min)	Nyquist (min)
57-60	11:15	11:38	23	11,5
58-61	14:09	14:35	26	13
61-58	17:31	17:56	25	12,5
Media				12,3

Tabla 3 – Tabla para el cálculo del sample time

Para encontrar la mínima variación medible, se buscaron puntos estables consecutivos que tuvieran una diferencia de al menos un 3%, puesto que es el error del higrómetro calculado anteriormente. Los cambios encontrados fueron los reflejados en la tabla 3.

Basándonos en el teorema de muestreo de Nyquist-Shanon, para poder muestrear correctamente estos cambios, el tiempo de muestreo o *sample time* se determina doblando su frecuencia máxima, es decir la mitad de los lapsos de tiempo calculados en la tabla anterior. Así entonces, el tiempo de muestreo para este caso se estableció en 12 minutos.

4.2. Código de Programa

4.2.1. Cabecera

Como se ha mencionado en el apartado anterior, el flow chart se traduce en un código de programa, también llamado *script*, adoptado del inglés. Este código se encuentra completo en el Anexo IV de esta memoria.

Como se aprecia, en la cabecera del programa se definen todos los pines de trabajo, las variables globales y se incluyen aquellas librerías necesarias para el uso de funciones específicas o relacionadas con algún sensor. Después se encuentra el *setup*, donde se establecen los pines como salidas o entradas, se inician las comunicaciones, se realiza la primera lectura de los sensores y se configura el timer 1, encargado de dar el aviso para la toma de datos.

4.2.2. Uso del timer 1

Dependiendo del microcontrolador encontramos diferentes timers, para la serie *ATmega* se diferencian 3 tipos, el timer 0 de 8 bits comúnmente usado en las placas de Arduino y similares para las funciones de tiempo (*delay()*, *millis()* y *micros()*), también asociadas a la generación de señales PWM y conversión ADC. El siguiente tipo son los timers 1, 3, 4 y 5, timers de 16-bits a disposición del programador, excepto del timer 1 usado para las funciones relacionadas con el uso de servos. Por último encontramos el timer 2, también de 8 bits y usado para la función *tone()*, la cual genera ondas cuadradas en el pin deseado.

El timer 0 no se suele usar puesto que se relaciona con la mayoría de las operaciones a realizar por la placa. El timer más usado es el 1 o sus análogos, debido que son de 16 bits y permiten registrar lapsos de tiempo mayores. En este caso se utiliza el timer 1, pues no es menester el uso de servos. En este fragmento del *setup* se realiza su configuración:

```

cli();
TCCR1A = 0;
TCCR1B = 0;
OCR1A = 62499;
TCCR1B |= (1 << WGM12);
TCCR1B |= (1 << CS10);
TCCR1B |= (1 << CS12);
TIMSK1 = (1 << OCIE1A);
sei();

```

Figura 31 – Fragmento de código de la configuración del timer 1

La función *cli()* desactiva cualquier interrupción interna que pudiera interferir en la configuración mientras que la función *sei()* las vuelve a activar. Después encontramos los registros TCCR1A y B (*Timer/Counter Register*) pertenecientes al timer 1. Cada timer tiene asociados estos registros de 8 bits, los cuales permiten configurar el funcionamiento del timer según los bits que activemos. El registro A no brinda ninguna configuración necesaria en este caso, por ello simplemente se queda a 0.

Bit	7	6	5	4	3	2	1	0
(0x81)	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Figura 32 – Esquema de bits del registro TCCR1B. [Datos]. Recuperado de Microchip ATmega640-2561/V (Datashheet). (2014)

De los bits que componen el registro, se activa el WGM12 que activa el CTC (*Clear Timer on Compare*), que a su vez activa el registro OCR1A. Al activar el CTC el valor del timer se compara constantemente con el valor grabado en el registro OCR1A. Finalmente, en el registro se activan los bits CS12 y CS10 lo que configura el timer con un *prescaler* de 1024 (el máximo) disminuyendo la resolución, es decir, aumenta el ciclo de reloj por el que el timer aumenta una unidad, ya que necesitamos medir un tiempo lo más largo posible. Teniendo en cuenta que la frecuencia de reloj del microcontrolador es de 16MHz, la resolución del timer pasa a ser:

$$resolución = \frac{1}{\frac{f_{clk}}{prescaler}} = \frac{1024}{16 \cdot 10^6 \text{ Hz}} = 64 \mu s$$

Esto significa, que al ser un timer de 16 bits, el tiempo máximo que podemos contar es de:

$$t_{max} = (2^{bits} - 1) \cdot resolución = (2^{16} - 1) \cdot 64 \cdot 10^{-6} s = 4,194 s$$

Para facilitar los cálculos se limita el tiempo de desborde a 4 segundos, de aquí la necesidad de activar el CTC. Por tanto, el valor del OCR1A es:

$$CTC = \frac{\text{tiempo deseado}}{\text{resolución timer}} - 1 = \frac{4 \text{ s}}{64 \cdot 10^{-6} \text{ s}} - 1 = 62499$$

Para registrar el desborde del timer 1 cuando vez llegue al valor del OCR1A, se usa el registro que vemos en la figura 30, el TIMSK1:

Bit	7	6	5	4	3	2	1	0
(0x6F)	–	–	ICIE1	–	OCIE1C	OCIE1B	OCIE1A	TOIE1
Read/Write	R	R	R/W	R	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Figura 33 – Esquema de bits del registro TIMSK. [Datos]. Recuperado de Microchip ATmega640-2561/V (Datashheet). (2014)

Para este registro se activa únicamente el bit OCIE1A. Con este bit activado, cuando el timer desborda se activa el vector de comparación interno A, lo que a su vez activa una interrupción. Esto quiere decir que el programa se detiene, independientemente de lo que este ejecutando, para realizar unas acciones determinadas, para luego seguir con su funcionamiento normal. Las acciones para ejecutar durante la interrupción deben ser lo más cortas posibles para no afectar al funcionamiento del programa, estas se definen en las siguientes líneas:

```
ISR(TIMER1_COMPA_vect) {
    segundos++;
    if (segundos == 180) {
        ciclo = true;
        segundos = 0;}}

```

Figura 34 – Fragmento de código de la rutina de interrupción

La función ISR (*Interrupt Service Routine*) permite definir las funciones a ejecutar cuando una interrupción es activada. Como se observa, esta función se activa en el momento que el vector de comparación A esté activado, es decir, cuando el timer 1 desborde. Para que esta rutina sea rápida, tan solo aumenta la variable segundos 1 unidad, que según la configuración anterior corresponde a 4 segundos. Cuando la variable llegue a 180 unidades (12 minutos) se activa la bandera “ciclo”, indicando el cumplimiento del sample time y por tanto que en el siguiente ciclo del bucle de control hay que tomar la lectura de los sensores.

4.2.3. Bucle de control

El bucle de control (*void loop*) empieza con la comprobación de la bandera “ciclo” y posteriormente entra en uno de los 3 procesos descritos en el apartado 4.1. El propio código incorpora varias anotaciones que aclaran la programación de estos procesos.

Uno de estos 3 procesos es el riego automático anteriormente detallado. Para este proceso era menester un segundo timer, debido que este timer solo se activará en el momento del riego, se aprovecha la función de tiempo asociada al timer 0 *millis()*. Esta función devuelve el tiempo desde que arrancó el programa y, mediante dos variables, podemos saber el tiempo que ha transcurrido entre dos puntos del programa. De esta forma se ahorra realizar toda la configuración anterior con un nuevo timer cada vez que se active el riego automático.

```
tiempoFin = millis();  
if(tiempoIni>0){tiempoTot += tiempoFin-tiempoIni;}  
tiempoIni = millis();
```

Figura 35 – Fragmento de código de la configuración del timer de riego automático

Una vez activada la bomba y tomada la lectura de los sensores del depósito, se ejecuta el fragmento que se observa en la imagen. De esta forma medimos el tiempo que transcurre desde este punto hasta completar el ciclo del bucle. Cabe recordar que, si la bandera de “Autoriego” está activada, este es el único proceso disponible, además de la toma de datos periódica y siempre pasaremos por este punto hasta que el riego automático finalice.

Debido a que las variables *tiempoIni* y *tiempoFin* por defecto están a 0 y no es posible saber el valor que devolverá la función *millis()*, se añade la condición *tiempoIni>0* para evitar sumar ningún valor al contador *tiempoTot* hasta que no se cumpla el primer ciclo.

Tal y como se ha mencionado en el apartado 4.1 al final del bucle de control se llama a la función *appData()*, encargada de realizar el intercambio de datos con el módulo wifi, actualizando las banderas del programa y la información visualizada en la interfaz de usuario.

4.3. Interfaz de Usuario

La interfaz de usuario queda limitada al procesamiento de datos que es capaz de ofrecer el módulo Wifi ESp8266 trabajando juntamente con la placa de desarrollo. Para poder ver la información de las lecturas y enviar comandos desde la interfaz, el módulo trabaja como servidor y cliente a la vez, usando un router o punto de acceso Wifi cercano. Dentro del código, antes del bucle de control, se configuran estos parámetros y se facilitan los datos de acceso a la red Wifi.

Una vez enviada la configuración al módulo, responde si el proceso se ha completado de forma satisfactoria o no, y devuelve la IP asignada al módulo dentro del router. Mediante la IP devuelta ya podemos conectarnos de forma local a la interfaz de usuario desde cualquier dispositivo conectado al mismo punto de acceso Wifi.

```
delay(500);
sendData("AT+CWMODE=3\r\n",1000,DEBUG); // Configuración punto de acceso (cliente o servidor)
  sendData("AT+CWJAP=\"SSID Red\", \"password\"\r\n",1000,DEBUG); //red wifi, usuario y clave
delay(10000);
sendData("AT+CWMUX=1\r\n",1000,DEBUG); // Múltiples conexiones activadas
sendData("AT+CIPSERVER=1,80\r\n",1000,DEBUG); // Servidor en el puerto 80
delay(2000);
sendData("AT+CIFSR\r\n",1000,DEBUG); // Se obtienen las direcciones IP
delay(1000);
webpage.reserve(2000);
```

Figura 36 – Fragmento de código de la configuración del módulo Wifi

Abriendo el puerto asignado al módulo, el acceso a la interfaz quedaría abierto a cualquier dispositivo en línea. Esta acción no se realizó puesto que el dispositivo no cuenta con protocolos de seguridad y dejaría vulnerable a ataques informáticos la red local y los dispositivos conectados a la misma red Wifi.

El portal web se construye mediante un *String* (cadena de caracteres) de gran tamaño en formato HTML básico. Como se aprecia, en la última línea de la configuración, se reservan 2000 bytes para la construcción del portal web. Esta construcción se realiza mediante la función antes indicada *appData()*.

A grandes rasgos, esta función comprueba si el módulo Wifi está enviando algún mensaje, vacía el módulo, se actualizan las banderas y se envía de nuevo el código HTML con la información actualizada. En el Anexo IV de esta memoria se encuentra el código completo de forma más detallada, con apuntes que facilitan la comprensión de su funcionamiento.

Primeramente, se encuentra la condición en la que el módulo esté enviando datos, seguidamente la comprobación del formato de entrada de datos para delimitar el inicio del mensaje correcto, luego se comprueba quien está enviando el mensaje (múltiples conexiones activadas) y, por último, se lee el resto del mensaje hasta vaciar el búfer para la entrada de nuevos datos.

Una vez leídos todos los datos, se actualizan las banderas correspondientes al modo de funcionamiento y al control de la bomba para el caso del modo manual. Una vez realizada esta actualización se empieza a construir la página HTML mediante la cadena de caracteres "webpage". El proceso es algo tedioso puesto que se trata de reunir toda la información de las variables de manera clara sin superar las limitaciones del módulo o el búfer de datos de la propia placa. El resultado se puede ver en la siguiente imagen.

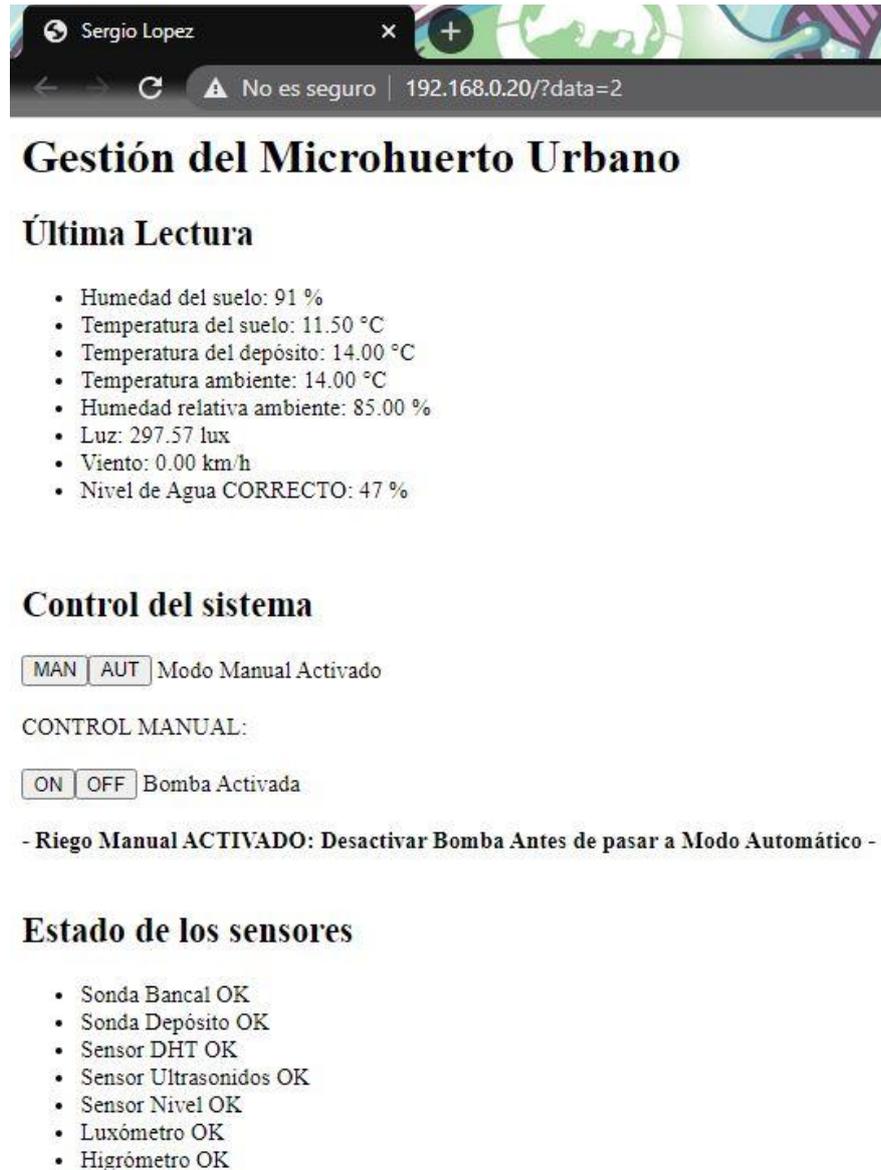


Figura 37 – Imagen de la interfaz de usuario

En la primera parte se encuentran, a modo de lista, los datos actualizados de la última lectura tomada. Seguidamente se observa el control del modo de funcionamiento y de la bomba, con dos pulsadores digitales para cada acción e información adicional para el usuario como el modo que se encuentra activo o avisos para el buen funcionamiento. Por último, se encuentra, también a modo de lista, el estado de los sensores donde se puede ver si algún sensor está fallando o, por el contrario, se encuentra en correcto estado. Esto es posible gracias a un sistema de banderas las cuales se activan si algún sensor devuelve valores extraños o, como en el caso de las sondas de temperatura y el DHT, se detecta el mal funcionamiento mediante funciones incorporadas en el propio sensor.

5. MONTAJE DEL PROTOTIPO

5.1. Construcción

5.1.1. Bancal

El bancal se realizó con maderas recicladas de pallets, se cortaron a la medida pertinente y se barnizaron para protegerlas de la humedad de la tierra y la lluvia. Posteriormente, se atornillaron las lamas entre sí, formando 3 pisos con una altura de unos 30 cm, y, finalmente, se fijó la estructura sobre un pallet entero para asegurar el correcto drenaje de la tierra de cultivo.



Figura 38 – Imagen de la estructura del bancal de cultivo

Una vez conseguida la estructura, se fijó la lona geotextil, se añadió el sustrato de cultivo y se plantaron los primeros cultivos, germinados y acrocentados previamente. También se añadieron cuerdas para delimitar las celdas de cultivo descritas en apartados anteriores.

5.1.2. Riego

Para el primer tramo del sistema de riego se usó un tubo flexible de espirocristal de PVC para uso alimentario de $\text{Ø}7 \times 10$ mm. Esto es debido a que la salida de la bomba dispone de una salida de 7 mm de diámetro y no era posible la conexión con ningún tubo de riego comercial convencional.

De este conducto principal se ramifican 3 ramas con un sistema de “Tes” para microtubo de riego convencional de 4 mm de diámetro. A cada rama se conectan 3 goteros de 4 l/h para cada celda de cultivo.



Figura 39 – Imagen del bancal con el sistema de riego instalado

Por último, el tramo principal se conecta con la bomba y esta se fija mediante sus ventosas al suelo del depósito, el cual se trata de un cubo convencional.



Figura 40 – Detalle del sistema de riego y el depósito

5.1.3. Estación meteorológica

Siguiendo el diseño propuesto, se construyó de forma sencilla una caja de madera a modo de estación meteorológica, donde se instalarían y conectarían todos los elementos del sistema.

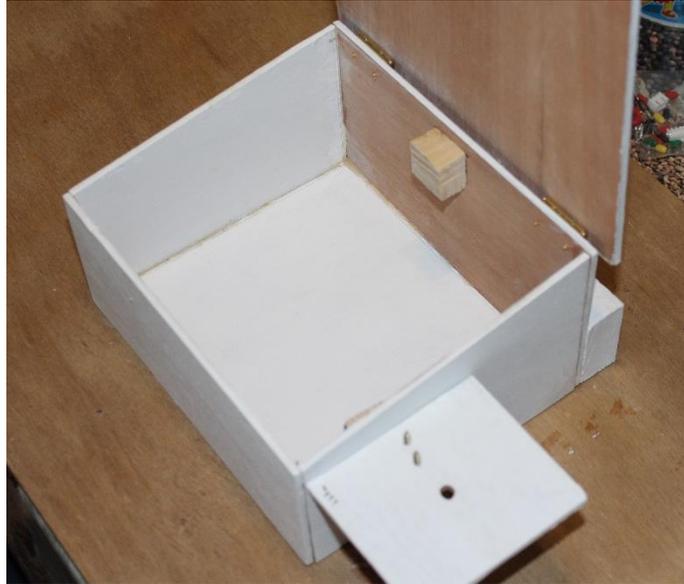


Figura 41 – Imagen de la estación meteorológica sin todos sus componentes

Sus medidas principales son 20x16x10 cm aproximadamente. Se añadieron dos pequeños refuerzos, uno para roscar en él un soporte de fijación en pared y el otro para fijar un soporte para el anemómetro mediante una pequeña “L” metálica. La cubierta es inclinada para la evacuación del agua y está fijada al cuerpo por dos bisagras que permiten el acceso al interior y realizar cómodamente las conexiones pertinentes. También se realizó una perforación en el suelo de la caja para el paso de todos los cables de los diferentes sensores.

Por último, se añadió un refuerzo en la parte posterior asegurando el paralelismo con la pared o superficie plana donde se fije y se pintó con pintura blanca plástica para exteriores, resultando la imagen de la figura 40.

5.2. Diagrama de conexiones

En el Anexo III de esta memoria se encuentra el diagrama de conexiones de todos los elementos del sistema en formato gráfico y esquemático. Con objeto de facilitar la visualización y buen entendimiento de estos, se han simplificado algunos elementos como las conexiones de potencia. Los pines de trabajo también varían respecto de la conexión final por conveniencia de la instalación, aunque se han respetado las diferencias entre pines digitales y analógicos, como es evidente.

Tal y como se describió en el análisis de los dos tipos de higrómetros, el sensor resistivo quedó descartado y no tiene uso en este sistema. Por el contrario, pese a que en el estudio de los sensores para el control del nivel del agua el sensor de ultrasonidos quedó descartado, se le otorgó otro uso distinto y es por ello por lo que sí aparece en el esquema de conexión.

A diferencia del sensor XKC, el sensor de ultrasonidos devuelve un valor de distancia, por lo cual esto permite calcular de forma aproximada el % de agua disponible en el depósito. Aunque estos datos serán informativos para el usuario y para una mejor prevención del llenado del depósito, también es útil para poder comprobar que el sensor XKC de nivel funciona correctamente, pese a que el control dependerá siempre de este último.

Cabe recordar que su medida se ve afectada por la temperatura ambiente puesto que el entorno de trabajo será en exterior, esta variable puede afectar a las medidas. Aprovechando que se disponía de dos sondas de temperatura DS18B20 (Apartado 2.3.5) preparadas para entornos de humedad, se decidió hacer uso de una de ellas para el control de la temperatura del depósito de agua. De esta forma se dispone de medidas más exactas para el sensor de ultrasonidos y también un valor de información útil para evitar problemas de congelación, proliferación de bacterias, hongos, etc.

5.2.1. Optoacopladores

Otro punto por destacar del diagrama es la aparición de tres optoacopladores 4N35. Uno de ellos es el encargado de aislar el pin digital usado para activar el relé de la bomba. Los relés pueden dar problemas de corrientes inversas, por lo que los módulos incorporan protecciones en la alimentación. Para añadir un grado más de protección se aisló también la señal digital proveniente de la placa mediante el esquema de la figura 41.a. Los otros dos optoacopladores actúan a modo de interruptor (esquema de la figura 41.b) en la alimentación del sensor de ultrasonidos y el higrómetro, puesto que han de trabajar en el exterior y en contacto con la humedad, no disponen de una correcta protección y poseen elementos electrónicos al descubierto. De esta manera, se añade una protección adicional para la placa y permite prolongar el tiempo de vida de estos sensores desconectando su alimentación mientras no están en uso.

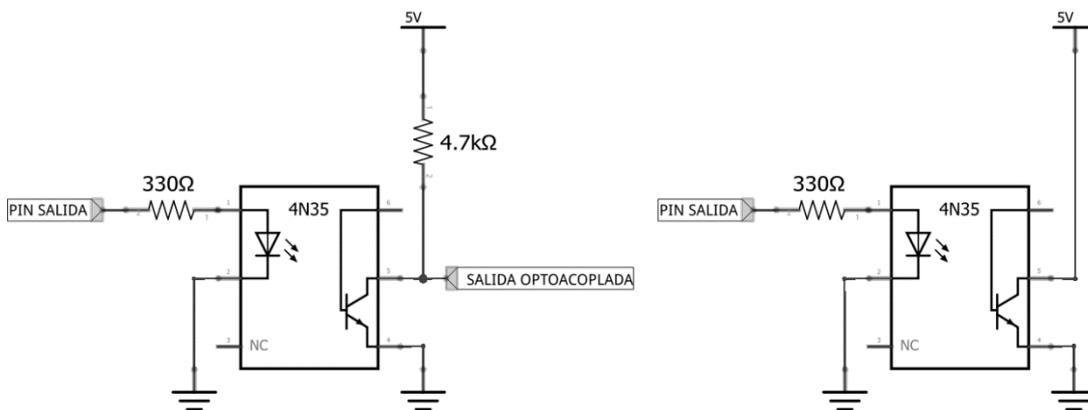


Figura 42.a y 42.b – Esquema para una salida optoacoplada y esquema de alimentación mediante optoacoplador

Las resistencias representadas en las figuras anteriores se usan para limitar la corriente: Unas en la entrada (para no dañar el fotodiodo emisor) y otra en la salida optoacoplada (para que la corriente sea absorbida por la resistencia y no por la salida).

5.2.2. Amplificador Operacional no inversor

Tal y como se relata en el apartado 3.5 sobre la calibración del anemómetro, los valores devueltos por este sensor son del orden de milivoltios. Para amplificar su señal y aprovechar el rango de conversión de 5V del ADC, se añadió un amplificador operacional LM358 previo a la entrada analógica para su conversión a velocidad. En la siguiente figura se observa el circuito correspondiente a un amplificador operacional no inversor.

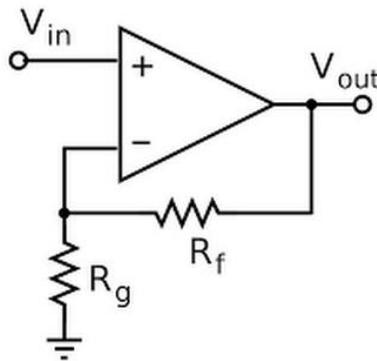


Figura 43 – Circuito de un amplificador operacional no inversor

Resolviendo el circuito, el valor del voltaje de salida se puede expresar como:

$$V_{out} = \left(\frac{R_F}{R_G} + 1 \right) V_{in} \rightarrow R_F = \left(\frac{V_{out}}{V_{in}} - 1 \right) R_G$$

Donde:

- V_{out} es el voltaje de salida del amplificador ($V_{out_m\acute{a}x} = 5V$ para el cálculo).
- V_{in} es el voltaje de entrada al amplificador ($V_{in_m\acute{a}x} = 0,45V$ para el cálculo).
- R_G es la resistencia a tierra y se fija en $1\text{ k}\Omega$.
- R_F es la resistencia de realimentación con un valor de $10\text{ k}\Omega$ según el cálculo.

5.2.3. Elementos resistivos

El resto de los componentes resistivos que figuran en los diagramas y esquemas, corresponden a las prescripciones dadas por los fabricantes de los diferentes dispositivos para su correcto funcionamiento.

5.2.4. Placa de conexiones

Se optó por agrupar todos estos elementos y demás componentes de conexión en una placa experimental para soldadura de estaño. A esta placa de conexiones se le añadieron patas de conexión para colocarla encima de la placa de desarrollo, de esta manera la conexión con los pines de trabajo y potencia es directa, además de optimizar el espacio.

También se construyeron cables para los sensores que bien no incorporasen cable, o su longitud fuera insuficiente, reciclando cables pares de cables de ethernet antiguos y conectores de 3 y 4 pines. En la placa se añadieron los conectores hembra correspondientes tanto para los conectores fabricados como para los cables ya existentes.

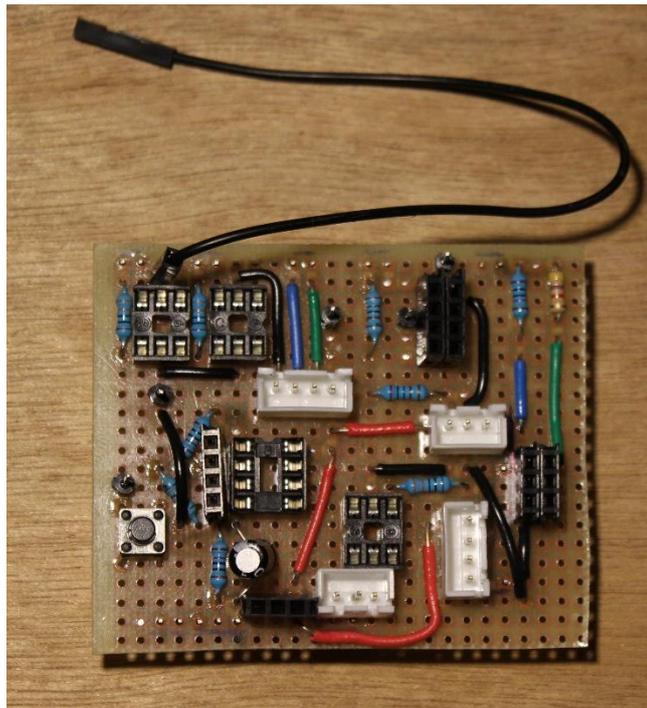


Figura 44 – Imagen de la placa de conexiones

Como se aprecia en la imagen superior, se añadieron cuatro zócalos para la conexión de chips, estos son para evitar soldar los optoacopladores y el amplificador operacional directamente a la placa y poderlos retirar a conveniencia. También se puede observar un pulsador, cuya función es replicar el botón de *reset* de la placa de desarrollo, esto se debe a que una vez colocada la placa de conexiones este queda cubierto e inaccesible.

5.3. Montaje final y puesta en marcha

Una vez construido todo, se pasó a realizar el montaje final de todos los componentes y a su primera puesta en marcha. En las siguientes figuras se muestra un despliegue de los todos los dispositivos y el resultado después de su instalación en la estación meteorológica.

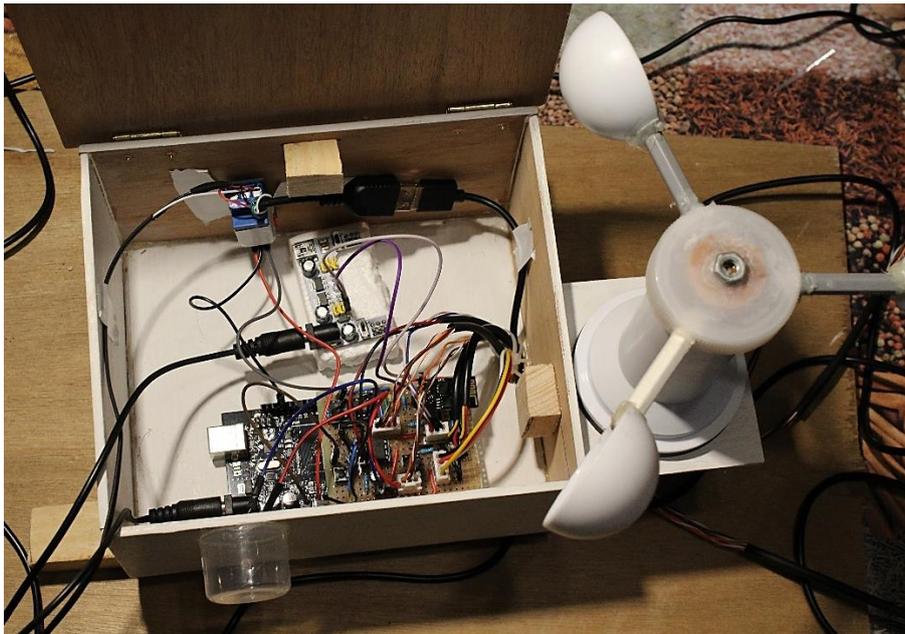
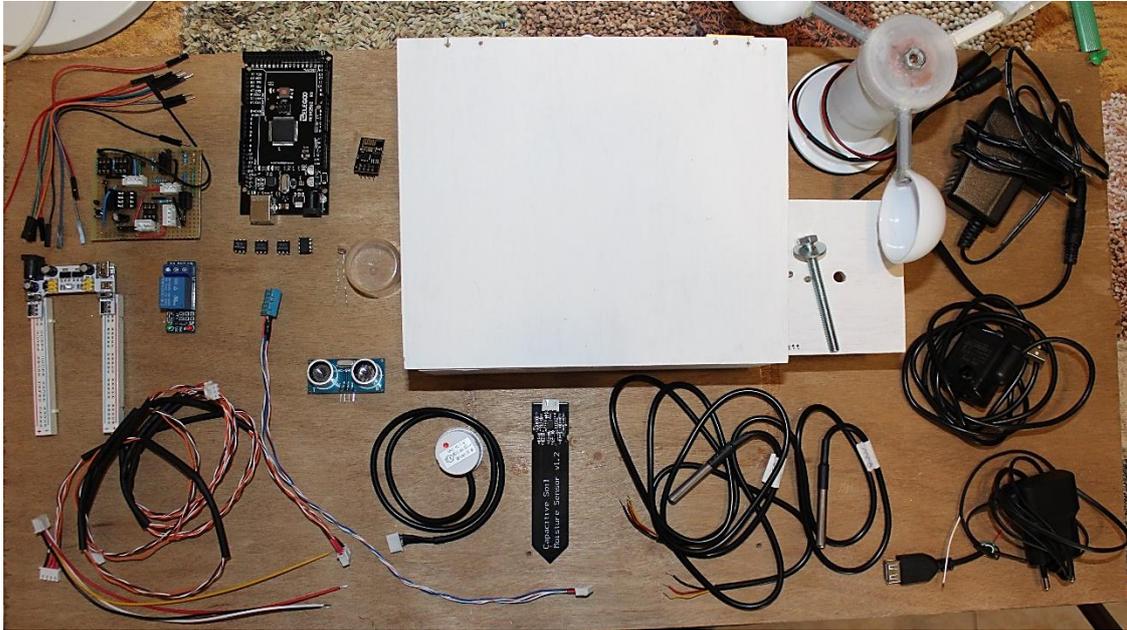


Figura 45.a y 45.b – Despliegue de todos los dispositivos que conforman el sistema e imagen del conjunto montado

Una vez montado el sistema completo, se realizaron las primeras pruebas y la consecuente depuración de errores, además de detallar la presentación del portal web.



Figura 46 – Primeras pruebas del sistema completo

Una vez conseguidos resultados satisfactorios se realizó el montaje final y su puesta en marcha, dando objeto a esta memoria y todo lo mostrado y desarrollado en ella.



Figura 47 – Imagen del montaje final del prototipo

6. MEJORAS Y PRÓXIMOS PASOS

Como se ha podido comprobar, la placa de desarrollo se basa en un microcontrolador, por lo que el procesamiento de datos queda limitado y las comunicaciones se antojan complicadas y lentas. Para mejorar el sistema cabría plantear la opción de basar el sistema en un microprocesador o bien reforzar el sistema actual con uno externo encargado de gestionar y procesar los datos enviados por el microcontrolador, mejorando la experiencia para el usuario final.

Una posible opción sería el sucesor del chip Wifi ESP8266 usado en este proyecto, el ESP32. Un chip de tipo SoC (*System on a Chip*), el cual cuenta con un procesador de 32 bits y doble núcleo operando a 160 o 240 MHz. Cuenta con tecnología Wifi y Bluetooth dual incorporada, además de otros protocolos de comunicación no inalámbrica y protocolos de seguridad en conexión a la red.

Además de las comunicaciones dispone de una gran variedad de periféricos, sensores de temperatura, reloj en tiempo real (RTC), conversor ADC de 18 canales, conversor DAC (Digital-Analógico), 16 salidas PWM, sensores capacitivos integrados y un largo etcétera.

Soporta diferentes lenguajes de programación, entre ellos el de Arduino, lo cual permite programarlo con el software open-source de Arduino. Existen diferentes módulos con este chip como núcleo ya preparados para desarrollar aplicaciones tal y como se ha realizado en este proyecto. También es posible desarrollar una placa de circuito impreso (PCB por sus siglas en inglés *Printed Circuit Board*) propia con este chip como núcleo, preparada con las salidas y entradas necesarias para este sistema.

En cualquier caso, este chip ampliaría el abanico de funciones de la aplicación, registro de datos, programaciones horarias, ahorro de consumo entre lecturas, interfaz de usuario más fluida con una mejor interfaz gráfica, etc.

Otra mejora a destacar sería el uso de una bomba hidráulica con una salida preparada para tubos de riego convencionales y alimentación de 12 VDC. De esta forma se podría alimentar a todo el sistema mediante un transformador de corriente convencional de 230VAC a 12VDC y 2 A de corriente.

En caso de querer desarrollar este prototipo a una versión comercial, sería menester un estudio de mercado previo y delimitar cual sería el producto final a distribuir, el software, una PCB propia, la estación meteorológica o todo el sistema completo con el bancal y el sistema de riego.

7. CONCLUSIÓN

La motivación de este trabajo académico era crear el prototipo de una unidad funcional y de bajo coste dentro del ámbito IoT (*Internet of Things*), que pusiera la tecnología en pro del ser humano y el avance hacia una sociedad más sostenible.

El diseño inicial planteaba un bancal de cultivos, el cual podría ser controlado mediante una aplicación móvil. El sistema debía ser de tamaño reducido y adaptable a pequeñas terrazas o balcones típicos de las urbes y el control del sistema se basaría en una placa de desarrollo compatible con el software open-source de *Arduino*. Tras la finalización de este proyecto académico se concluye:

- Aunque el prototipo no se ha desarrollado de forma profesional ni comercial, se han cumplido los objetivos iniciales, obteniendo una unidad funcional de bajo coste y tamaño reducido, adaptable a balcones y terrazas urbanas.
- El sistema ofrece una interfaz de usuario que facilita toda la información relativa a los cultivos y a su entorno, además del control del sistema de riego a través de cualquier dispositivo móvil conectado a la red. Por el momento, el acceso ha quedado limitado a redes locales por razones de seguridad informática.
- Pese a que el sistema planteaba inicialmente el uso de baterías, esto fue descartado debido a que la bomba necesita una alimentación difícilmente soportable por una batería durante largos periodos de tiempo.
- El sistema desarrollado podría tener una proyección comercial en caso de obtener financiación y contar con apoyo de profesionales cualificados, como se ha especificado, previo estudio de mercado.
- Este proyecto ha permitido trabajar en los diferentes ámbitos relacionados con las placas de desarrollo y las aplicaciones IoT, como son la electrónica y la microelectrónica, la calibración y el desarrollo de sensores, la programación, la comunicación inalámbrica y el desarrollo web. Lo que supone un crecimiento profesional y personal en ámbitos previamente desconocidos para el autor de la presente memoria.

8. BIBLIOGRAFÍA

1. ARDUINO. Arduino Reference. <https://www.arduino.cc/reference/en>.
2. ADAFRUIT. Adafruit Learn. <https://learn.adafruit.com>.
3. HOGARMANIA. Jardinatis. <https://www.hogarmania.com/jardinaria/fichas/>.
4. APLICACIONES DE LOS OPTOACOPLADORES. Luis Antonio de la Cruz Reyes, 2017. Recuperado de <https://ingenieriaelectronica.org/aplicaciones-de-los-optoacopladores/>.
5. EL AMPLIFICADOR OPERACIONAL. Vicente García, 2010. Recuperado de <https://www.diarioelectronicohoy.com/blog/el-amplificador-operacional>.
6. FUNCIONES DEFINIDAS POR EL USUARIO. Enrique Crespo, 2016. Recuperado de <https://aprendiendoarduino.wordpress.com/2016/11/16/funciones-definidas-por-usuario-2/>.
7. ESP8266, MÓDULO WIFI ESP-01. J. Morales, 2018. Recuperado de www.playbyte.es/electronica/arduino/esp-01s-modulo-wifi-basado-en-esp8266.
8. CAP 3. INTERRUPCIONES. David Álvarez, 2012. Recuperado de <https://tallerarduino.wordpress.com>.
9. HIGRÓMETRO. Wikipedia, 2018. Recuperado de <https://es.wikipedia.org/wiki/Higrómetro>.
10. CONCEPTOS CLAVE PARA EL RIEGO ÓPTIMO. SensaCultivo, 2018. Recuperado de <https://sensacutivo.es/2018/11/01/4-conceptos-clave-riego-optimo/>.
11. HUMEDAD EN SUELOS DE DIFERENTE TEXTURA. Traxco, 2009. Recuperado de <https://www.traxco.es/blog/tecnologia-del-riego/humedad-en-suelos-de-diferente-textura>.

9. ANEXOS

9.1. Anexo I: Manual de Usuario

9.1.1. Configuración del acceso Wifi

Dentro del código de programación, en el apartado de *setup*, será necesario introducir el nombre de nuestra red y la contraseña, tal como se aprecia en la siguiente imagen y se indica en el propio código. Una vez introducidas las credenciales se procede a cargar el programa en la placa de desarrollo mediante el soporte del software de *Arduino* y el cable USB. Para cualquier duda con el software de *Arduino* consultar la documentación de su página web: <https://www.arduino.cc/en/Guide>.

```
sendData("AT+CWMODE=3\r\n",1000,DEBUG); // Configuración punto de acceso (cliente o servidor)
sendData("AT+CWJAP=\"SSID Red\", \"password\"\r\n",1000,DEBUG); //red wifi, usuario y clave
```

Figura 48 – Fragmento del código donde se introducen las credenciales Wifi

La primera vez que iniciemos el sistema, es menester conectarnos al monitor serie del software a través del puerto USB del computador. Importante NO conectar la alimentación del transformador de 12 VDC! En el monitor serie empezarán a aparecer mensajes conforme la configuración se ha realizado correctamente. En caso de recibir caracteres extraños comprobar, en la parte inferior del monitor, que la velocidad de comunicación seleccionada es 19200 baud y el retorno de carro está activo (“Ambos NL & CR”).

Si la configuración se realiza satisfactoriamente, se mostrará un último mensaje con una lista de direcciones IP. El módulo Wifi se asigna automáticamente una dirección IP libre dentro de la red Wifi en la que esté. Esta dirección corresponde a la IP estática (STAIP en la figura 48) y será el acceso a la interfaz de usuario del sistema desde cualquier dispositivo conectado a la misma red. Una vez realizado este paso ya no será necesario estar conectado al ordenador mediante el puerto USB y se podrá conectar la placa mediante el *jack* de alimentación de 12 VDC.

```
OK
AT+CIFSR
+CIFSR:APIP,"192.168.4.1"
+CIFSR:APMAC,"4e:11:ae:03:2b:89"
+CIFSR:STAIP,"192.168.0.20"
+CIFSR:STAMAC,"4c:11:ae:03:2b:89"
```

Figura 49 – Ejemplo de la lista de direcciones IP recibida

9.1.2. Conexión Intranet – Internet

Como ya se ha mencionado, el sistema no dispone de protocolos de seguridad en red, por lo que no se recomienda abrir la interfaz a internet. En cualquier caso, si el usuario lo desea y bajo su propia responsabilidad, puede abrir el puerto 80 a la IP asignada dentro de la configuración del punto de acceso Wifi local. De esta manera, el acceso a la interfaz será posible desde cualquier dispositivo conectado a internet.

9.1.3. Interfaz de usuario

En la primera parte de la interfaz se observan los datos de la última lectura de los sensores. En este punto no hay interacción, son datos informativos.

Última Lectura

- Humedad del suelo: 91 %
- Temperatura del suelo: 11.50 °C
- Temperatura del depósito: 14.00 °C
- Temperatura ambiente: 14.00 °C
- Humedad relativa ambiente: 84.00 %
- Luz: 330.12 lux
- Viento: 0.00 km/h
- Nivel de Agua CORRECTO: 52 %

Figura 50 – Imagen del primer apartado de la interfaz de usuario

El siguiente apartado es para la selección del modo de funcionamiento del sistema y el control de la bomba. Al lado de cada botón podemos observar que modo de funcionamiento se encuentra activo y el estado de la bomba. En la parte inferior de este mismo apartado aparecerán avisos según el estado del sistema. Pueden ser uno o varios entre estos tres avisos.

Control del sistema

MAN AUT Modo Manual Activado

CONTROL MANUAL:

ON OFF Bomba Desactivada

- Riego Automático ACTIVADO: Control del Sistema Bloqueado -
- Riego Manual ACTIVADO: Desactivar Bomba Antes de pasar a Modo Automático -
- Nivel de Agua BAJO!: Imposible Activar Riego -

Figura 51 – Imagen del segundo apartado de la interfaz de usuario

Si el riego automático está activo, el control del sistema queda bloqueado en pro del buen funcionamiento del sistema. Por seguridad, en caso de activar el riego de forma manual no es posible el cambio a modo automático hasta que se desactive la bomba. Es muy importante mantener el correcto nivel de agua del depósito para el buen funcionamiento del sistema. En caso de que el nivel del agua rebaje el umbral mínimo, no será posible la activación ni del riego manual ni automático.

El tercer y último punto se refiere al estado de los sensores. El sistema es capaz de detectar valores anormales durante las lecturas de los sensores, en ese caso aparecerá en este apartado como que el sensor está fallando y será menester su pronta revisión o sustitución. En caso de fallo del higrómetro o del sensor de nivel, el sistema no activará el riego automático o manual.

Estado de los sensores

- Sonda Bancal OK
- Sonda Depósito OK
- Sensor DHT OK
- Sensor Ultrasonidos OK
- Sensor Nivel OK
- Luxómetro OK
- Higrómetro OK

Figura 52 – Imagen del tercer y último apartado de la interfaz de usuario

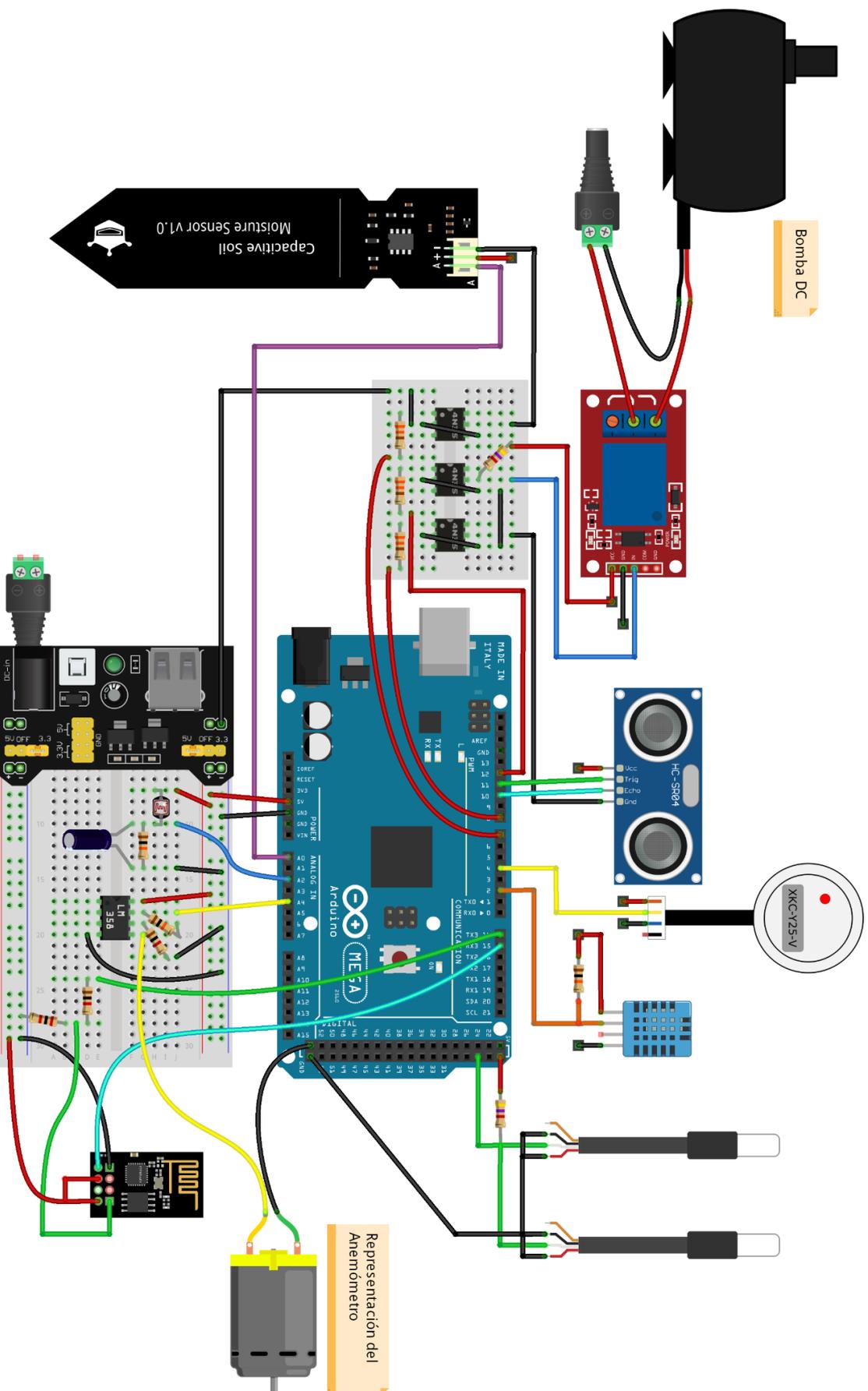
9.1.4. Mantenimiento

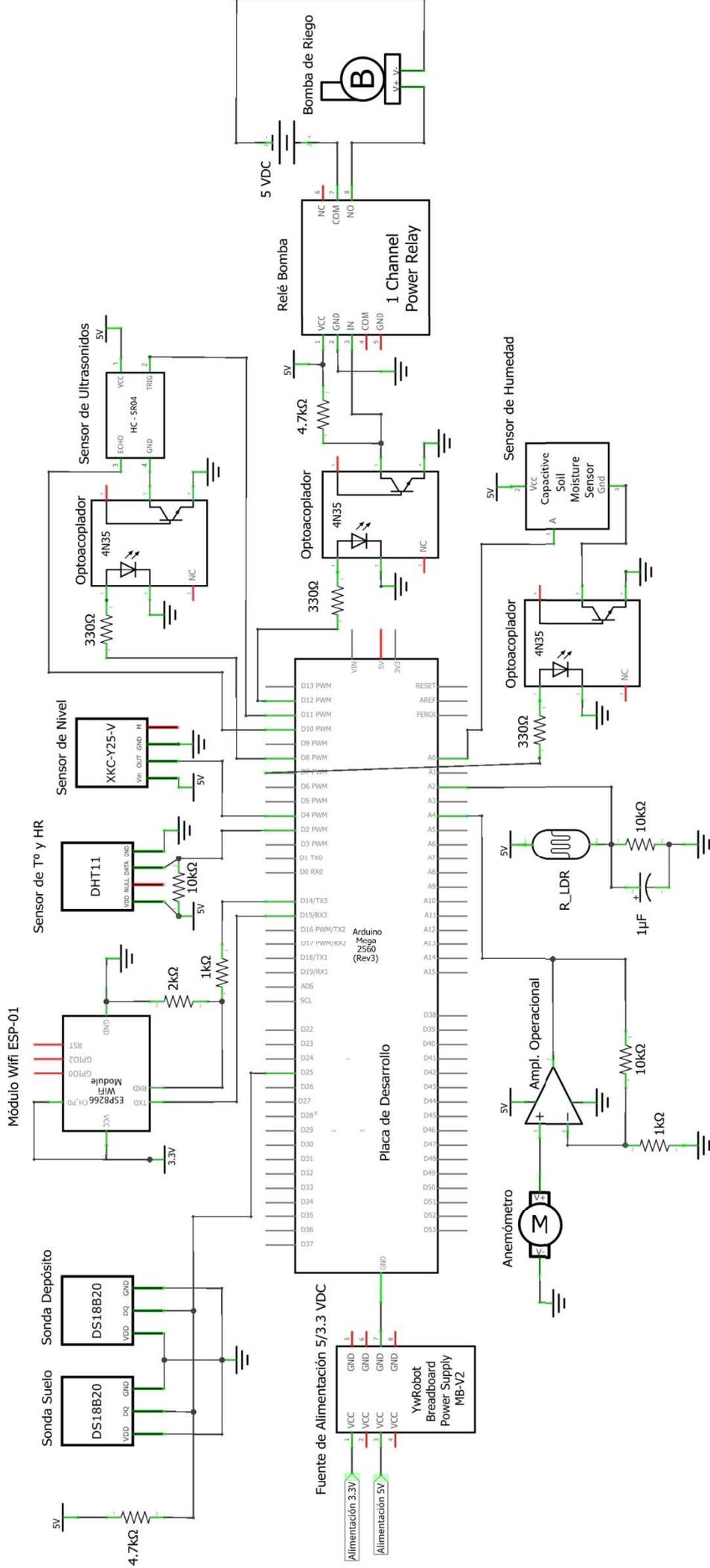
No se requiere ningún mantenimiento específico del sistema más que el control del agua del depósito. Hay que asegurar que siempre se encuentre en el nivel adecuado y comprobar que no haya elementos sólidos que puedan taponar la entrada de agua de la bomba. Por el mismo motivo, se recomienda la inspección y limpieza periódica del filtro que incorpora la bomba.

Se recomienda usar agua de lluvia o reposada para evitar la presencia de cloro y cal en la medida de lo posible. También se aconseja la aplicación de algún tipo de anti-algas no perjudicial para los cultivos.

Como recomendación, inspeccionar periódicamente el estado de los sensores especialmente en época de lluvias o calor intenso. En época de heladas, excepto para uso en invernadero, no se recomienda su uso en exterior, podría provocar mal funcionamiento de los componentes y congelación del depósito.

9.3. Anexo III: Esquemas eléctricos





9.4. Anexo IV: Código del Programa

```

#include <OneWire.h>           //Libreria para bus 1-Wire
#include <DallasTemperature.h> // Libreria para las funciones de las
sondas DS18B20
#include <SimpleDHT.h>        // Libreria del sensor de temperatura y
humedad
#include <math.h>             // Libreria para calculos complejos como
potencias y logaritmos
#include <avr/io.h>           //Libreria para la configuración del
timer
#include <avr/interrupt.h>    // Libreria para la configuración de la
interrupción

#define pinUS_POWER 3        // Pin 3 Alimentación HC_SR04
#define pinRELE 7           // Pin 7 Salida optoacoplada del relé
#define pinUS_ECHO 14       // Pin 14 HC-SR04 Echo
#define pinUS_TRIGGER 15    // Pin 15 HC-SR04 Trigger
#define ONE_WIRE_BUS 43     // Pin 43 Bus 1-Wire
#define pinXKC 50           // Pin 50 Sensor Nivel de Agua
#define pinHIGRO_POWER 44   // Pin 44 Alimentación Higrómetro
#define pinANEM 0           // Pin A0 Lectura Analogica del Anemómetro
#define pinLDR 4            // Pin A4 Lectura Analogica del LDR
#define pinHIGRO 9          // Pin A9 Lectura Analogica del Higrómetro
Capacitivo
#define DEBUG true          // Atajo para programación del modulo wifi

//Definimos las variables globales
volatile unsigned int segundos;
volatile unsigned ciclo = false;
bool stringComplete = false, flagAgua = false, flagManual = false,
flagBomba = false, stateFlag = false, riegoFlag = false, autoRiego =
false, checkAuto = false, releFlag = false, optoFlag = false;
bool errflagSondaB = false, errflagSondaD = false, errflagDHT = false,
errflagUS = false, errflagXKC = false, errflagLDR = false, errflagHigro =
false;
int err = SimpleDHTErrSuccess;
int pinDHT11 = 39;
unsigned int humedadSuelo = 0, nivelAgua = 0;
unsigned long tiempoIni = 0, tiempoFin = 0, tiempoTot = 0;
const long tiempoRiego = 300000;
float tempAire = 0, humedadAire = 0, tempBancal = 0, tempDeposito = 0;
double luz = 0, viento = 0;
String webpage = "", Deposito = "";

OneWire oneWire(ONE_WIRE_BUS); //Definimos el bus 1-Wire y el pin
de datos
DallasTemperature sensors(&oneWire); //Definimos el acceso a los sensores
dentro del bus
DeviceAddress bancal = {0x28, 0x8D, 0x8E, 0x8C, 0x3A, 0x19, 0x01,
0xA8}; //Definimos la dirección a la sonda del bancal
DeviceAddress deposito = {0x28, 0x12, 0x76, 0x38, 0x3A, 0x19, 0x01,
0x3A}; // Definimos dirección a la sonda del depósito
SimpleDHT11 dht11(pinDHT11); //Definimos el pin de trabajo para
el sensor DHT11
HardwareSerial &esp8266 = Serial1; //Modificamos el nombre del búfer
Serial para Módulo Wifi

```

```

void setup() {
  Serial.begin(19200); //Iniciamos comunicación con el
  búfer Serial para el monitor Serie
  esp8266.begin(19200); //Iniciamos la comunicación del
  módulo wifi
  sensors.begin(); // Inicializamos las sondas
  conectadas a 1-Wire
  sensors.setResolution(bancal, 10); //Definimos la resolución de los
  datos de la sonda del bancal
  sensors.setResolution(deposito, 9); //Definimos la resolución de los
  datos de la sonda del depósito

  //Definimos las salidas digitales y las ponemos a 0 (Por defecto los
  pines trabajan como entradas)
  pinMode(pinUS_TRIGGER, OUTPUT);
  pinMode(pinUS_POWER, OUTPUT);
  pinMode(pinRELE, OUTPUT);
  pinMode(pinHIGRO_POWER, OUTPUT);
  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(pinUS_TRIGGER, LOW);
  digitalWrite(pinUS_POWER, LOW);
  digitalWrite(pinRELE, LOW);
  digitalWrite(pinHIGRO_POWER, LOW);
  digitalWrite(LED_BUILTIN, LOW);

  //configuramos el modulo wifi
  delay(500);
  sendData("AT+CWMODE=3\r\n",1000,DEBUG); //
  Configuración punto de acceso (cliente o servidor)
  sendData("AT+CWJAP=\"SSID Red\", \"password\"\r\n",1000,DEBUG); // Red
  wifi: usuario y clave
  delay(10000);
  sendData("AT+CIPMUX=1\r\n",1000,DEBUG); //
  Múltiples conexiones activadas
  sendData("AT+CIPSERVER=1,80\r\n",1000,DEBUG); //
  Servidor en el puerto 80
  delay(2000);
  sendData("AT+CIFSR\r\n",1000,DEBUG); // Se
  obtienen las direcciones IP
  delay(1000);
  webpage.reserve(2000); //Se
  reservan 2KB para la construcción de la página web

  //Primera lectura de todos los sensores
  digitalWrite(pinUS_POWER, HIGH); //Damos la señal al optoacoplador
  para alimentar el sensor de ultrasonidos
  digitalWrite(pinHIGRO_POWER, HIGH); //Damos la señal al optoacoplador
  para alimentar el higrómetro
  delay(100);
  sondasTemp(tempBancal, tempDeposito); // - Se recojen los datos de cada
  sensor mediante funciones definidas por nosotros
  dhtRead(); // - Las variables se actualizan
  a través de punteros para ahorrar espacio
  anemometro(viento); // - El sensor DHT11 actualiza
  las variables a través de una función propia, no se le pasan valores
  luxometro(luz);
  higrometro(humedadSuelo);
  depositoUS(nivelAgua);
  depositoXKC(flagAgua);
  digitalWrite(pinUS_POWER, LOW); // Damos la señal al optoacoplador
  para desactivar el sensor de ultrasonidos

```

```

    digitalWrite(pinHIGRO_POWER, LOW); // Damos la señal al optoacoplador
    para desactivar el higrómetro

    //seteamos Timer1 (ver memoria)
    cli();
    TCCR1A = 0;
    TCCR1B = 0;
    OCR1A = 62499;           // Tiempo de comparación (4 seg)
    TCCR1B |= (1 << WGM12); // Activamos la comparación
    TCCR1B |= (1 << CS10);  // Prescaler a 1024
    TCCR1B |= (1 << CS12);
    TIMSK1 = (1 << OCIE1A); // Vector para la interrupción
    sei();
}

void loop() {
    // En cualquier caso, si desborda el timer lo primero es la toma de
    datos
    if (ciclo) {
        if(!optoFlag){
            digitalWrite(pinUS_POWER, HIGH); //Activamos la alimentación del
            ultrasonidos (si no estaba activado por un proceso previo)
            optoFlag = true;
        }
        digitalWrite(pinHIGRO_POWER, HIGH); //Activamos la alimentación del
        higrómetro
        delay(500);
        sondasTemp(tempBancal, tempDeposito);
        dhtRead();
        anemometro(viento);
        luxometro(luz);
        higrómetro(humedadSuelo);
        depositoUS(nivelAgua);
        depositoXKC(flagAgua);
        digitalWrite(pinUS_POWER, LOW);
        digitalWrite(pinHIGRO_POWER, LOW);
        ciclo = false; // Se desactivan las banderas propias del proceso
        checkAuto = false; // y también las banderas que evitan la
        repetición de procesos
        riegoFlag = false;
        optoFlag = false;
    }

    //Si está activado el autorriego no se puede activar ningún otro
    proceso más que la toma de datos.
    if(autoRiego){
        if(!optoFlag){
            digitalWrite(pinUS_POWER, HIGH); // Durante el tiempo de autoRiego
            se mantiene activo el sensor de ultrasonidos
            optoFlag = true;
        }
        delay(500); // Ya que el loop no para en ningún momento, se añade un
        delay de 500ms (sample time mínimo requerido por el sensor XKC de nivel)
        if(!errflagUS && !errflagSondaD){depositoUS(nivelAgua);} // Se
        recogen los datos de nivel de agua (si el sensor de ultrasonidos o la
        sonda del deposito no funciona esta medida no se toma)
        depositoXKC(flagAgua); // La medida
        capital es la del XKC, la medida del ultrasonidos informativa
    }
}

```

```

    if (flagAgua && !stateFlag) {
        digitalWrite(pinRELE, HIGH); // Si el nivel de agua es correcto y
        la bomba no está activa, mandamos señal al relé para encender el riego
        stateFlag = true;
    }

    //timer para el autoriego (ver memoria)
    tiempoFin = millis();
    if(tiempoIni>0){tiempoTot += tiempoFin-tiempoIni;} // cada vez que
    pasemos por este punto se añade el tiempo de ciclo
    tiempoIni = millis();
    if (!flagAgua || tiempoRiego <= tiempoTot){ // Si el nivel de agua
    baja del umbral minimo o el timer desborda acaba el proceso
        digitalWrite(pinRELE, LOW); // Cerramos la bomba
        digitalWrite(pinUS_POWER, LOW); //Desactivamos el ultrasonidos
        stateFlag = false; //Todas las banderas y variables de
    tiempo a 0
        riegoFlag = true; //excepto la de Riego, esto hará
    que el sistema no vuelva a activar el autoriego hasta que no se tome una
    nueva lectura
        autoRiego = false;
        optoFlag = false;
        tiempoFin = 0;
        tiempoTot = 0;
        tiempoIni = 0;
    }
}
//si no está activo el autoRiego y el usuario manda la señal de modo
manual el sistema entra en modo manual
if (flagManual && !autoRiego) {
    if(!optoFlag){
        digitalWrite(pinUS_POWER, HIGH); //mientras el modo manual esté
    activo el sensor de ultrasonidos también
        optoFlag = true;
    }
    delay(500); //Igual
    que para el autoriego
    if(!errflagUS && !errflagSondaD){depositoUS(nivelAgua);} // Igual
    que para el autoriego
    depositoXKC(flagAgua);

    // Si el nivel de agua no es suficiente y la bomba esta activa se para
    if (!flagAgua && flagBomba && stateFlag == flagBomba) {
        digitalWrite(pinRELE, LOW);
        riegoFlag = true;
        stateFlag = false;
    }
    // Si el nivel es suficiente y se manda la señal de activar riego (y
    la bomba NO estaba activa previamente) se activa la bomba
    if (flagBomba && flagAgua && flagBomba != stateFlag) {
        digitalWrite(pinRELE, HIGH);
        stateFlag = flagBomba;
    }
    // Si se manda la señal de desactivar el riego (y la bomba SÍ estaba
    activa previamente) se desactiva la bomba
    if (!flagBomba && flagBomba != stateFlag) {
        digitalWrite(pinRELE, LOW);
        stateFlag = flagBomba;
        riegoFlag = true;
    }
}
}

```

```

// Si no esta activo el modo manual entramos en modo automático
if (!flagManual && !autoRiego && !riegoFlag && !checkAuto) {
    if (humedadSuelo <= 45 && luz <= 50.00 && flagAgua && !errflagHigro)
    { // Si se cumplen las condiciones necesarias se activa el aviso para el
autoriego
        autoRiego = true;
    }
    else{checkAuto = true;} //Si no se cumplen las condiciones se activa
la bandera de comprobación hasta la siguiente toma de datos. No tiene
sentido comprobar los mismos valores.
    if(optoFlag){
        digitalWrite(pinUS_POWER, LOW); // Al pasar a modo automático
desactivamos el sensor de ultrasonidos.
        optoFlag = false;
    }
}

appData(); // Llamamos a la función appData para el intercambio de
datos con el módulo WiFi
}

//Función para la interrupción del timer
ISR(TIMER1_COMPA_vect) {
    segundos++; //Cada cuatro segundos la cuenta aumenta una
unidad
    if (segundos == 180) { //180x4=720 720/60=12 min sample time (ver
memoria)
        ciclo = true;
        segundos = 0;
    }
}

void sondasTemp(float &tempSondaB, float &tempSondaD) {
    sensors.requestTemperatures();
    tempSondaB = sensors.getTempC(bancal);
    tempSondaD = sensors.getTempC(deposito);
    if (tempSondaB == -127.00) {
        if (!sensors.isConnected(bancal)) { // Si presentan mal
funcionamiento devuelven -127
            errflagSondaB = true; // adicionalmente nos cercioramos
que esten conectadas mediante la función isConnected (1 Sí 0 No)
        }
    }
    if (tempSondaD == -127.00) {
        if (!sensors.isConnected(deposito)) {
            errflagSondaD = true;
        }
    }
    return;
}

void dhtRead (void) {
    if ((err = dht11.read2(&tempAire, &humedadAire, NULL)) !=
SimpleDHTErrSuccess) { // Si no hay error se actualizan las variables de
temperatura y humedad
        errflagDHT = true;
    }
    return;
}

```

```

void depositoUS (unsigned int &level) {
    digitalWrite(pinUS_TRIGGER, LOW);
    delayMicroseconds(3);
    digitalWrite(pinUS_TRIGGER, HIGH); // Trigger Impuls 10 us
    delayMicroseconds(10);
    digitalWrite(pinUS_TRIGGER, LOW);
    unsigned long tiempo = pulseIn(pinUS_ECHO, HIGH); // Medición del
tiempo por la entrada Echo
    float velocidad_sonido = 331.5 + tempDeposito * 0.6; // Cálculo de la
veocidad del sonido conociendo la temperatura dentro del depósito
    unsigned int distancia = round((float(tiempo) * velocidad_sonido) /
20000.0); // Cálculo de la distancia  $v = x \cdot (t/2)$  y conversión de unidades
    byte d_sensor = 4; // Distancia del cabezal del sensor al nivel
máximo de agua
    byte l_deposito = 21; // Nivel máximo de agua
    level = (l_deposito - distancia + d_sensor) * 100 /
(l_deposito); // Cálculo del % de agua restante en el
depósito
    if (tiempo == 0) { // Si no detecta lapso de tiempo el sensor
presenta mal funcionamiento
        errflagUS = true;
    }
    return;
}

void depositoXKC (bool &agua) {
    if (digitalRead(pinXKC) == LOW) { // No detecta, el nivel de agua
reabaja el umbral
        agua = false;
        Deposito = "<li>Nivel de Agua INSUFICIENTE!: "; // Nos facilita la
creación de la interfaz de usuario
        if (nivelAgua > 20 && !errflagUS && !errflagSondaD) {errflagXKC =
true;}
    }
    if (digitalRead(pinXKC) == HIGH) { // Sí detecta, el nivel de agua esta
correcto
        agua = true;
        Deposito = "<li>Nivel de Agua CORRECTO: "; // Nos facilita la
creación de la interfaz de usuario
    }
    return;
}

void higrometro (unsigned int &valor) {
    valor = analogRead(pinHIGRO); // Lectura de 0 a 1023
    if (valor == 0) {errflagHigro = true;} // Valor minimo 473, 0 indica
mal funcionamiento
    valor = constrain(valor, 473, 813); // Comprimimos el valor dentro
de los limites para evitar malos resultados
    valor = map(valor, 473, 813, 100, 0); // Transformamos el valor a un
valor entre 0 y 100%
    return;
}

void luxometro (double &lux) {
    double input = double(analogRead(pinLDR)); // Leemos el valor del LDR y
lo pasamos a double (formato requerido para los calculos)
    if (input == 0) {errflagLDR = true;} // Es posible tener un valor
de 0 pero es muy complicado, se añade este error a modo de precaución
porque no bloquea ninguna función
    double a = 1023.00;
}

```

```

    double R = 10.00;
    double m = -0.7913;
    double b = 2.2382;
    double R_ldr = ((a * R) / input) - R; // Cálculo de la función para
obtener el equivalente en Ohms (ver memoria)
    double exp = (log10(R_ldr) - b) / m; // Cálculo del exponente para
la transformación a lux
    lux = pow(R, exp); // Paso final para la
transformación a lux
    return;
}

void anemometro (double &v) {
    double conversion = 0.2162; // Lectura analogica X conversion a Volts
(5/1023) X conversion del AO [Vin=Vout/((Rf/Rg)+1)] -> (1/11) X
conversion mV-km/h (1000/2,0554)
    v = double(analogRead(pinANEM)) * conversion; // Adquisición y
transformación
    return; // En este caso no hay
posibilidad de detección de errores
}

void appData(void) {
    if(esp8266.available()) { // Comprueba si el ESP8266
esta enviando mensaje
        if(esp8266.find((char*)" +IPD, ")){ // La información enviada por
el módulo empieza siempre con +IPD
            String respuesta="";
            delay(1000);
            int connectionId = esp8266.read()-48; // Restar 48 para conversión
ASCII-DECIMAL

            while(esp8266.available()){ // Leemos caracter a caracter
hasta vaciar el buffer
                char c = esp8266.read();
                respuesta+=c;
                if(respuesta.indexOf("?data=0") >0){ // Leemos si el usuario ha
pulsado algún botón y cuál
                    if(!autoRiego){flagManual = true;}
                }
                if(respuesta.indexOf("?data=1") >0){
                    if(!stateFlag){flagManual = false;}
                }
                if(respuesta.indexOf("?data=2") >0){
                    if(flagManual){flagBomba = true;}
                }
                if(respuesta.indexOf("?data=3") >0){
                    flagBomba = false;
                }
            }

            //Empezamos a rellenar el contenido de la web en formato HTML
            webpage = "<html><head><meta http-equiv=""refresh""
content=""5""><title>Sergio Lopez</title></head><body>";
            webpage+="<h1>Gesti&ocute;n del Microhuerto
Urbano</h1>"; // Cabecera
            webpage+="<h2>&Uacute;ltima
Lectura</h2><p>"; // Primer apartad
            webpage+=lecturas(); // ver función
            webpage+="<br><h2>Control del
sistema</h2>";

```

```

// Botones para el modo de funcionamiento y el control de la bomba.
    webpage+="<button
onClick=location.href='./?data=0'>MAN</button>"; // En función de qué
boton presionemos en la interfaz, se añadirá a la información enviada por
el módulo
    webpage+="<button
onClick=location.href='./?data=1'>AUT</button>"; // ?data=x, según el
valor de x se realizan las diferentes acciones necesarias tal como vemos
arriba.
    if(flagManual){
        webpage+="    Modo Manual
Activado"; // Añadimos junto al botón el
modo activo
    }
    else{
        webpage+="    Modo Autom&acute;tico Activado";
    }
    webpage+="<br><br> CONTROL MANUAL:";
    webpage+="<br><br><button
onClick=location.href='./?data=2'>ON</button>";
    webpage+="<button onClick=location.href='./?data=3'>OFF</button>";
    if(stateFlag){
        webpage+="    Bomba
Activada"; // Añadimos junto al botón
el estado de la bomba
    }
    else{
        webpage+="    Bomba Desactivada";
    }
    //Información adicional para el usuario (ver memoria)
    if(autoRiego){
        webpage+="<br><br><strong>- Riego Autom&acute;tico ACTIVADO:
Control del Sistema Bloqueado -</strong>";}
    if(stateFlag){
        webpage+="<br><br><strong>- Riego Manual ACTIVADO: Desactivar Bomba
Antes de pasar a Modo Autom&acute;tico -</strong>";}
    if(!flagAgua){
        webpage+="<br><br><strong>- Nivel de Agua BAJO!: Imposible Activar
Riego -</strong>";}
    webpage+="<br><br><h2>Estado de los sensores</h2>";
    webpage+=estadoSistema(); // Ver función
    webpage+="</body></html>"; // Cerramos HTML

    String cipSend = "AT+CIPSEND="; // Notificación del próximo envío
de datos
    cipSend += connectionId;
    cipSend += ", ";
    cipSend +=webpage.length();
    cipSend += "\r\n";

    sendData(cipSend,500,DEBUG); // Enviamos la notificación
    sendData(webpage,2000,DEBUG); // Enviamos el código HTML de la web

    // Cerramos la conexión
    String closeCommand = "AT+CIPCLOSE=";
    closeCommand+=connectionId;
    closeCommand+="\r\n";
    sendData(closeCommand,1000,DEBUG);
}
}
}

```

```

String sendData(String command, const int timeout, boolean debug){
    String response = "";
    esp8266.print(command);          // Se envían los datos deseados al
    módulo esp8266
    long int time = millis();
    while( (time+timeout) > millis()){ // Se añade un timeout para evitar
    colapsos
        while(esp8266.available()){
            char c = esp8266.read();    // Leemos carácter a carácter la
            respuesta del módulo a nuestro envío
            response+=c;
        }
    }
    Serial.print(response); // Imprimimos la respuesta para ver el correcto
    funcionamiento y la configuración (necesitamos conocer la IP asignada al
    módulo para conectarnos a la interfaz)
    return response;
}

String lecturas(void){ // Se añade poco a poco toda la información
    respectiva a los sensores en un String que posteriormente se añade al
    String de la web en la función appData()
    String datos="";
    datos="<ul><li>Humedad del suelo: "; datos+=humedadSuelo; datos+="
    %</li>";
    datos+="<li>Temperatura del suelo: "; datos+=tempBancal; datos+="
    &deg;C</li>";
    datos+="<li>Temperatura del dep&ocute;sito: "; datos+=tempDeposito;
    datos+=" &deg;C</li>";
    datos+="<li>Temperatura ambiente: "; datos+=tempAire; datos+="
    &deg;C</li>";
    datos+="<li>Humedad relativa ambiente: "; datos+=humedadAire; datos+="
    %</li>";
    datos+="<li>Luz: "; datos+=luz; datos+=" lux</li>";
    datos+="<li>Viento: "; datos+=viento; datos+=" km/h</li>";
    datos+=Deposito; datos+=nivelAgua; datos+=" %</li></ul>";
    return datos;
}

String estadoSistema(void){ // Se añade poco a poco toda la información
    respectiva al estado de los sensores en un String que posteriormente se
    añade al String de la web en la función appData()
    String systemState="";
    if(errflagSondaB){systemState="<ul><li>Sonda Bancal ERROR!</li>";}
    else{systemState="<ul><li>Sonda Bancal OK</li>";}
    if(errflagSondaD){systemState+="<li>Sonda Dep&ocute;sito
    ERROR!</li>";}
    else{systemState+="<li>Sonda Dep&ocute;sito OK</li>";}
    if(errflagDHT){systemState+="<li>Sensor DHT ERROR!</li>";}
    else{systemState+="<li>Sensor DHT OK</li>";}
    if(errflagUS){systemState+="<li>Sensor Ultrasonidos ERROR!</li>";}
    else{systemState+="<li>Sensor Ultrasonidos OK</li>";}
    if(errflagXKC){systemState+="<li>Sensor Nivel ERROR!</li>";}
    else{systemState+="<li>Sensor Nivel OK</li>";}
    if(errflagLDR){systemState+="<li>Lux&ocute;metro ERROR!</li>";}
    else{systemState+="<li>Lux&ocute;metro OK</li>";}
    if(errflagHigro){systemState+="<li>Higr&ocute;metro
    ERROR!</li></ul>";}
    else{systemState+="<li>Higr&ocute;metro OK</li></ul>";}
    return systemState;}

```

**Documento II. PLIEGO DE
CONDICIONES**

PLIEGO DE CONDICIONES

Objeto del pliego

El presente pliego de condiciones presenta los requisitos generales que se exige al contratista para que lleve a cabo el proyecto “Diseño e implementación de un sistema automático para la gestión de un microhuerto urbano”.

El objeto del documento es el de especificar las condiciones técnicas para el correcto desarrollo del proyecto. No se pretende definir todos los detalles constructivos del mismo, pues será responsabilidad del contratista que la selección de componentes y detalles constructivos se correspondan con las últimas normas de diseño y técnicas aplicables.

Documentación del proyecto y compatibilidad de documentos

El presente proyecto se compone de los siguientes documentos que son compatibles entre sí y se complementan unos a otros:

- Documento 1. MEMORIA y ANEXOS
- Documento 2. PLIEGO DE CONDICIONES
- Documento 3. PRESUPUESTO

Condiciones técnicas

Para la realización del proyecto se pondrá a disposición del contratista toda la documentación antes detallada. Cualquier cambio será consultado con la dirección facultativa. Cualquier cambio podrá ser valorado siempre y cuando no modifique la funcionalidad del sistema y sea acorde a los objetivos marcados para el presente proyecto.

El documento prioritario para seguir durante el desarrollo del proyecto será siempre la memoria y sus anexos. En este documento se detallan los objetivos, funcionalidades y metodología para el desarrollo del sistema.

Desde el punto de vista legal y facultativo el documento capital será el presente pliego de condiciones.

En cuanto a la jerarquía de documentos, el presupuesto quedaría en última posición pues no presenta obligado cumplimiento y solo pretende ser una cuantificación económica orientativa del proyecto, además de una medición de los materiales necesarios.

Condiciones facultativas

La dirección facultativa será responsabilidad de un ingeniero del ámbito industrial, electrónico o automático. Este será responsable del correcto desarrollo del proyecto y de alcanzar los objetivos propuestos en la memoria del proyecto.

En cuanto a la fabricación se trata, el contratista será el máximo responsable de los errores de fabricación y deberá responder ante la dirección facultativa. También el contratista deberá facilitar cualquier certificado necesario en materia de calidad de los materiales, maquinaria y equipo empleado para la fabricación.

Entregables

Al finalizar el proyecto la dirección facultativa deberá entregar una memoria técnica detallando todos los procesos y metodologías seguidas para su desarrollo. Anexado a esta memoria también se entregará el código programa desarrollado y todos los esquemas eléctricos pertinentes.

Como documento aparte se entregará un presupuesto y estado de mediciones donde se detallen todos los materiales utilizados y sus costes asociados.

Por último, se dispondrá de un pliego de condiciones donde se describan todas las disposiciones técnicas, legales y económicas derivadas del desarrollo del proyecto, así como todos los certificados y garantías relativas al producto final.

Documento III. PRESUPUESTO

PRESUPUESTO

El presente documento se introduce a modo informativo del estado de mediciones y costes asociados al proyecto “Diseño e implementación de un sistema automático para la gestión de un microhuerto urbano”. El bancal de cultivo no se adhiere al presente presupuesto ya que no se asocia al diseño o implementación del sistema de automatización. Sí se incorpora el sistema de riego debido a que se asocia a la elección de la bomba hidráulica.

Presupuesto de ejecución material (PEM)

A continuación, se listan todos los costes asociados a los materiales del proyecto:

DESCRIPCIÓN	UNIDADES	PRECIO (€)	TOTAL (€)
SENSORES			30,15
Módulo higrómetro capacitivo V1.2	1,00	4,99	4,99
Sonda DS18B20 1m cable	2,00	3,26	6,52
Módulo ultrasonidos HC-SR04	1,00	3,99	3,99
Sensor capacitivo de nivel XKC-Y25-V	1,00	7,95	7,95
Sensor de temperatura y humedad DHT 11	1,00	4,29	4,29
Motor de 2,4 VDC para Anemómetro	1,00	2,04	2,04
Fotorresistor LDR	1,00	0,37	0,37
ACTUADORES			12,98
Mini Bomba Sumergible <i>Flintronic</i> USB 5VDC	1,00	8,99	8,99
Relé 5V KF-301 canal único	1,00	3,99	3,99
SISTEMA DE RIEGO			10,83
Tubo de espirocristal de PVC Ø7x10 mm 2m	1,00	1,78	1,78
microtubo de riego Ø4mm 5m	1,00	3,89	3,89
Gotero 4 L/h para microtubo	9,00	0,23	2,07
Inserto para ramificación en microtubo	3,00	0,16	0,48
"Te" para empalme microtubo	9,00	0,29	2,61
CONTROL			19,48
Placa de desarrollo <i>Elegoo</i> Mega 2560 R3	1,00	13,99	13,99
Módulo Wifi ESP8266 ESP-01	1,00	5,49	5,49
ALIMENTACIÓN			11,14
Fuente de alimentación protoboard 5V y 3,3V	1,00	2,94	2,94
Transformador 240VAC a 12VDC y 2A	1,00	8,20	8,20
COMPONENTES ELECTRÓNICOS			5,23
Optoacoplador 4N35	3,00	0,64	1,92
Amplificador Operacional LM358	1,00	0,85	0,85
Placa para soldadura de componentes	1,00	1,90	1,90
Condensador electrolítico	1,00	0,04	0,04
Resistencias 1/4 W diferentes valores	13,00	0,04	0,52
TOTAL COSTES MATERIALES			89,81

En el siguiente cuadro se listan los recursos de software y humanos requeridos para el proyecto:

DESCRIPCIÓN	UNIDADES	PRECIO (€)	TOTAL (€)
SOFTWARE			77,00
Arduino IDE	-	-	-
Fritzing	1,00	8,00	8,00
Microsoft Office	1,00	69,00	69,00
COSTES POR RECURSOS HUMANOS			11.250,00
Ingeniero Junior	450,00	25,00	11.250,00
TOTAL SOFTWARE Y COSTES POR RECURSOS HUMANOS			11.327,00

Presupuesto de ejecución por contrato (PEC) y total

DESCRIPCIÓN	COSTE (€)
A. Costes Materiales	89,81
B. Software y Costes por Recursos Humanos	11.327,00
C. Presupuesto de Ejecución Material (PEM) (A+B)	11.416,81
D. Gastos Generales (13% del PEM)	1.484,19
E. Beneficio Industrial (6% del PEM)	685,01
F. Presupuesto de Ejecución por Contrato (PEC) (C+D+E)	13.586,00
G. Impuesto sobre el Valor Añadido (IVA) (21% del PEC)	2.853,06
COSTE TOTAL DEL PROYECTO (F+G)	
	16.439,06

Por lo tanto, el coste del proyecto es de DIECISÉIS MIL CUATROCIENTOS TREINTA Y NUEVE CON SEIS CÉNTIMOS. Este presupuesto tiene efecto a noviembre de 2020, la fluctuación de precios en el tiempo puede hacer variar la presente partida.