



Gradu amaierako lana / Trabajo fin de grado
Ingeniaritza Elektronikoko gradua / Grado en Ingeniería Electrónica

Sistema para la gestión del desarrollo agrícola

Egilea/ Autor/a:
Kevin Martínez Quintana
Zuzendaria/Director:
Iker Caballero Ortiz de Zarate

Leioa, 2023ko ekainaren 20a / Leioa, 20 de junio de 2023

Índice

1	Introducción	1
2	Objetivos	2
3	Motivaciones	4
4	Diseño	5
4.1	Arduino UNO WiFi Rev.2	6
4.2	Sensor analógico de humedad de suelo	8
4.3	Sensor analógico de temperatura LM35	9
4.4	Sensor analógico de luz ambiental	9
4.5	Módulo relé digital 5A	10
4.6	Motor paso a paso 28BYJ-48	11
4.7	Micro bomba	12
4.8	Gestor de energía solar 1.0	13
4.9	Panel solar de epoxi	15
4.10	Batería LiPo LP-523450-1S-3	16
4.11	Sensor de medición de color RGB TCS34725	17
5	Desarrollo	18
5.1	Control del sistema de riego	20
5.2	Control de iluminación	21
5.3	Comienzo de implementación de sensores	21
5.4	Implementación sensor de humedad	24
5.5	Implementación sensor temperatura	25
5.6	Implementación sensor de luz ambiental	26
5.7	Implementación del relé	27
5.8	Implementación de la bomba	28
5.9	Implementación de motor paso a paso 28BYJ-48	29
5.10	Implementación gestor de energía solar	30
5.11	Implementación de panel solar	31
5.12	Implementación de la batería LP-523450-1S-3	31
5.13	Implementación de envío de datos	33
5.14	Implementación sistema de riego y toldo manual	36
5.15	Implementación sensor de color TCS34725	38
5.16	Consumo	39
6	Prototipo	41
7	Costes de proyecto	44
8	Conclusiones	45
9	Líneas futuras	46

10 Anexo **48**

- 10.1 Instrucciones de montaje 48
- 10.2 Instrucciones código 48
- 10.3 Sensor de humedad 49

1 Introducción

La agricultura es un sector primordial para el ser humano desde hace más de 10000 años, permitiendo a los nómadas asentarse en una localización y persistir sin necesidad de abandonarla. Dejar atrás su antigua vida les dio la oportunidad de desarrollar otras ciencias como pueden ser la matemática, filosofía u otras disciplinas científicas. La agricultura ha sido desarrollada durante siglos para poder optimizar las cosechas y mejorar las calidades de los productos obtenidos mediante diferentes métodos como los químicos o la rotación de cultivos.

La agricultura es uno de los sectores más expandidos e investigados, por su carácter mundial y las variaciones que surgen debido a los diferentes climas y ambientes que se encuentran en todo el planeta. La necesidad de alimentarse ha llevado al ser humano a intentar cultivar diferentes alimentos en zonas no predispuestas a dicho desarrollo mediante modificaciones genéticas mejorando sus características y aumentando sus resistencias para poder subsistir en situaciones de climas extremos, dando la posibilidad a aumentar la producción y suministrar alimentos a poblaciones que sufren de escasez de alimentos por causa de estos climas.

El desarrollo de la tecnología ha proporcionado muchas posibilidades de ayuda para el cultivo, estas pueden ser de diferentes escalas como para plantas individuales, jardines medianos o latifundios¹. Para la última dimensión, el desarrollo es de gran interés debido a lo mencionado anteriormente y la financiación invertida en ella. En los demás tamaños la tecnología no es muy avanzada y la financiación no suele ser generosa.

Teniendo conocimiento de esta división de intereses y financiación, el proyecto se enfoca en el diseño de un sistema de gestión del desarrollo agrícola a pequeña y media escala, para ello se hace uso de un sistema de Arduino UNO WiFi Rev.2, mediante el cual se monitorizan diferentes parámetros y se actúa de manera diferente dependiendo de ellos. Los parámetros a tener en cuenta son **humedad, temperatura e incidencia solar**, indispensable en el crecimiento de los cultivos e integridad y salud de las plantas; estos son medidos mediante sensores específicos. Además de las diferentes actuaciones debido a los parámetros se añade diferentes sensores para saber el color del fruto a cultivar y un sistema de envío de datos mediante WiFi, para poder transmitir los datos obtenidos. El suministro de energía se hace mediante una batería recargable y una placa solar para conseguir una independencia energética y reducir costes de infraestructura. El sistema se hace mediante Arduino debido a su precio asequible y fácil manejo para poder transferir el sistema a diferentes plantaciones y además, cuenta con un gran surtido de sensores para conectar al sistema de fácil uso e implementación, generando así un sistema fácilmente replicable para el usuario. A lo largo de este proyecto se diseña un prototipo y se hace uso del mismo para monitorizar una planta y observar su desempeño en su desarrollo. Se busca la maniobrabilidad y un diseño compacto para ocupar el menor espacio posible.

¹Explotación agraria de grandes dimensiones

2 Objetivos

La agricultura siempre ha sido un tema de mi agrado, debido a los años que he realizado este tipo de actividades. Es por esto que el objetivo de este TFG es aunar todo el conocimiento teórico adquirido en la carrera y generar un sistema de control de desarrollo agrícola para monitorizar los cultivos. El inicio de la idea de este proyecto se remonta años atrás con el grupo IE Makers en el segundo año de carrera. En este grupo se creó un sistema de riego simple mediante Arduino que regaba cuando la humedad de la tierra se bajaba de un límite impuesto. Con el conocimiento adquirido en los siguientes años y con el entusiasmo de desarrollar más este proyecto se ha puesto como objetivo crear un sistema más avanzado de Arduino, profundizar en el sistema Arduino y obtener conocimiento de programación en C. Teniendo en cuenta estos objetivos los puntos a cubrir son los siguientes:

- Aprender sobre las técnicas de desarrollo de sistemas de agricultura: Para tener un conocimiento previo al desarrollo del sistema se procede a hacer una investigación de las diferentes técnicas y sistemas que se usan hoy en día. Para ello se buscan diferentes artículos con estos contenidos para poder contrastar los diferentes métodos y elegir cual es el más adecuado de todos ellos para el sistema que se quiere desarrollar.
- Desarrollar el sistema completo con sensores para control de desarrollo agrícola: El sistema esta compuesto por Arduino para la gestión del desarrollo agrícola de plantaciones de pequeña y media escala. Es sistema debe ser capaz de monitorizar una plantación y suministrar datos al usuario para el conocimiento del desarrollo de estas. Para eso se hace uso de diferentes sensores que proporcionan los parámetros con los que se actúa en la plantación para su correcto desarrollo y posterior envío de datos al usuario vía WiFi. La creación se realiza mediante sensores especializados en cada una de sus tareas, para comenzar se usa un sensor de humedad para calcular la necesidad de agua, un sensor de temperatura para actuar de forma diferente según el grado de calor y un sensor de incidencia solar con el cual se acciona un toldo motorizado para reducir la cantidad de luz directa que la plantación recibe. Se desea un sistema independiente de cualquier estructura de energía para así poder subsistir sin la necesidad de infraestructura eléctrica. Además de su independencia energética se reduce al máximo el consumo obteniendo así un producto mas eficiente.
- Sistema compacto, económico y fácil de usar: Otra faceta que se quiere cumplir es el de un sistema compacto, económico y de fácil implementación, para facilitar el uso del sistema en diferentes plantaciones, facilitar su transporte y para que sea un sistema asequible para la gran mayoría de agricultores; siendo capaz de reducir costes de transporte del usuario, generando así beneficios debido al reciente incremento en precio de los combustibles. Además la edad de los agricultores cada vez es mayor debido al desinterés de este campo por los jóvenes, así que se desea facilitar el trabajo de los mayores a bajo coste con este sistema. El último requerimiento está directamente unido a lo anteriormente dicho de los mayores, al facilitar el uso al punto de tan solo tener que añadir el código, instalarlo y no necesitar mucho mantenimiento, ahorrando muchos problemas que pueden surgir.

- Crear prototipo para experimentación: Obtenido el sistema se crea un entorno de experimentación que consiste en un prototipo de maceta con la infraestructura necesaria para realizar todas las acciones deseadas por el sistema en los diferentes casos y así poder obtener unos resultados fidedignos de su funcionamiento. El prototipo es construido a partir de materiales disponibles en cualquier tienda de bricolaje para la facilitar así la réplica del proyecto para uso personal. Se considera que un prototipo de tamaño adecuado debe ser transportable para poder observar diferentes comportamientos en diferentes ambientes y situaciones.

Una vez cumplidos todos estos puntos el objetivo principal del TFG se habrá cumplido. Otro objetivo pero menos prioritario es acercar la experiencia de realizar el TFG al que sería un trabajo realizado en una empresa, debido a su carácter de investigación que abarca desde métodos de desarrollo, comparación y selección de los elementos del sistema, reducción de costes y viabilidad del desarrollo del proyecto como producto y/o herramienta de estudio.

3 Motivaciones

Las principales motivaciones de este TFG son: optimizar el uso del agua, facilitar el mantenimiento de huertos y la posibilidad de usarse como herramienta de aprendizaje.

- La emergente emergencia global de la escasez de agua está siendo un gran problema en la producción de alimentos para la población mundial. Este problema está escalando a grandes velocidades y el uso del agua está siendo cada vez más descuidado bajo la suposición de muchos de que el agua es ilimitada. La escasez de agua está generando restricciones de uso en muchas localidades de nuestro país, impidiendo regar las plantaciones y dejando a muchos pueblos, de pocos residentes, sin un flujo constante de agua por priorizar dicho flujo a las ciudades principales. Prohibir el regadío de los latifundios está generando muchas pérdidas monetarias y la pérdida de muchos trabajos. Añadiendo el efecto del calentamiento global y los cambios tan bruscos de temperaturas, las producciones agrarias están siendo destruidas en pleno momento de flor o incluso cuando ya ha fructificado. Aunque el sistema diseñado en este trabajo no es adecuado para tales dimensiones, se enfoca en tamaños medios y pequeños. En los cuales se pretende reducir el volumen de agua utilizado, reducir las probabilidades de contraer enfermedades e intentar prevenir la pérdida de producto por los cambios de temperaturas.
- El mantenimiento de un huerto de tamaño medio conlleva un gran tiempo, además de un trabajo físico duro. Regar los campos plantados es una tarea indispensable para que crezcan. Esta tarea es realizada por todo tipo de personas y por todo tipo de edades, además por experiencia propia se conoce el cansancio que sufren los mayores al realizarla. Por ese motivo se desea crear un producto que facilite el proceso de regado y mantenimiento de los huertos.
- Adicionalmente, este TFG podría utilizarse como herramienta pedagógica para incentivar a los jóvenes a programar y aprender sobre el desarrollo hardware. La necesidad de ingenieros con conocimientos de sistemas hardware está en auge y la escasez de estos genera un vacío difícil de rellenar. Por eso, motivando a los jóvenes a enfocar sus conocimientos en el desarrollo software y hardware de sistemas ayudaría a rellenar dicho vacío y fomentar el estudio de ingenierías enfocadas en hardware. Para ayudar en el progreso de replicar el proyecto se dan todas las pautas necesarias para conseguir el producto desarrollado.

Como motivación personal el haber estado años llevando una huerta con mi familia ha generado en mi un gusto por la agricultura que ha quedado arraigado. Esta actividad, parte de la preparación del terreno, pasando por el mantenimiento de las plantaciones, hasta la recolección del producto final. Aún siendo todas las tareas arduas, la gratificación de obtener un producto después de realizar tanto trabajo y la calma y tranquilidad que se obtiene durante el proceso es algo que me gustaría compartir con el mayor número posible de personas, tanto jóvenes como adultos, puesto que siempre hay tiempo para una nueva afición.

4 Diseño

En esta sección se lleva a cabo la explicación de los elementos que componen el sistema y su funcionamiento. Dichos elementos son los siguientes:

- **Arduino UNO WiFi Rev.2:** Placa de Arduino encargada de gestionar y controlar las mediciones y acciones de los diferentes sensores [3].
- **Sensor de humedad:** Sensor encargado de medir la humedad del suelo en la que el cultivo se encuentra, necesario para el control de riego [6].
- **Sensor de temperatura:** Sensor encargado de medir la temperatura ambiente, necesario para el control de riego y control del toldo [8].
- **Sensor de luz ambiental:** Sensor encargado de medir la luminiscencia del ambiente, necesario para el control del riego y control del toldo [11][10].
- **Módulo relé:** Relé encargado de activar o desactivar la bomba de riego o humidificación [17] [15].
- **Motor paso a paso:** Motor encargado de abrir o cerrar el toldo o sistema de control solar, necesario para el control del toldo [2].
- **Bomba:** Bomba encargada de transferir el agua a la planta, necesaria para el control de riego.
- **Gestor energía solar:** Encargado de controlar la energía suministrada al sistema completo. Administra la energía y se hace cargo de recargar la batería mediante paneles solares [14].
- **Panel solar:** Encargado de alimentar corriente al gestor para recargar la batería y alimentar el sistema .
- **Batería:** Suministra corriente cuando las placas solares no tienen potencia suficiente para alimentar el sistema completo [7].
- **Sensor de color RGB:** Sensor encargado de devolver valores RGB para calcular el color del producto a monitorizar [5].
- **Módulo WiFi del Arduino:** Módulo Nina W102 uBlox encargado de conectar la placa Arduino al WiFi para mandar datos del sistema al receptor de datos [16].

Para una representación gráfica de la interconexión de los elementos y el funcionamiento del sistema dependiendo de los sensores ya dispuestos, se presenta el siguiente diagrama de bloques:

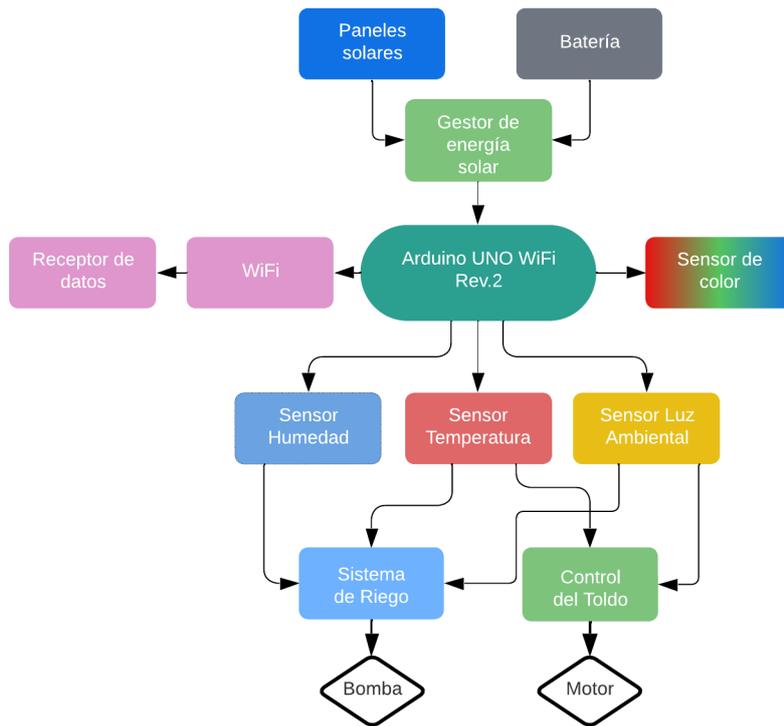


Figura 1: Diagrama de bloques del sistema completo

Todos los sensores y actuadores están directamente conectados al Arduino, sin embargo, los paneles solares y la batería van unidos al gestor de energía solar. Como actuadores solo se cuenta con las bombas de riego y el motor. Se ha representado un bloque de WiFi para representar el funcionamiento del envío de datos, sin embargo tan solo es un módulo de Arduino, no se trata de ningún sensor o actuador. Conociendo los sensores y actuadores que se utilizan no está de más tener cierto conocimiento sobre ellos, por ello se hace un resumen de funcionamiento y su uso en el proyecto de cada uno de ellos para luego exponer un esquemático del sistema completo.

4.1 Arduino UNO WiFi Rev.2

El origen de Arduino se encuentra en Ivrea Interaction Design Institute y fue creado como una herramienta fácil de prototipado. La placa Arduino está compuesta por un microcontrolador y está diseñada para el fácil uso del usuario principiante y también con gran versatilidad para usuarios avanzados. La facilidad de su uso se basa en su fácil manejo debido a sus dimensiones, lenguaje de programación simple y la disponibilidad de un amplio abanico de librerías en constante desarrollo por la comunidad y por los diseñadores del mismo. Su facilidad de uso se puede ver por el número tan amplio de proyectos que se llevan al cabo del año por principiantes. La programación se realiza a través de comandos sencillos que son ejecutados por el microcontrolador mediante el lenguaje C, programado mediante el software IDE (Integrated Development Environment). El uso de este sistema es usado en un amplio rango de proyectos desde herramientas para experimentos científicos hasta gente principiante con entusiasmo de aprender sobre robótica y programación.

La placa utilizada en el proyecto es la placa Arduino UNO WiFi Rev.2. Está compuesta por un microcontrolador ATmega4809 y tiene 14 pines input/output digitales y 6 entradas analógicas, alimentado a través de USB o cable de alimentación CC, un botón de reset, entre otras características. Contiene todo lo necesario para sustentar el microcontrolador y conservar su integridad. A continuación, se muestra una tabla resumen con las características principales:

Tabla 1: Especificaciones de la placa Arduino UNO WiFi Rev.2

Especificaciones	
Microcontrolador	ATmega4809
Tensión de operación	5V
Tensión de alimentación	6 - 20V
Pines I/O digitales	5
Pines entrada analógicos	6
Corriente DC pines I/O	20 mA
Corriente DC pin 3.3V	50 mA
SRAM	6,144 Bytes
EEPROM	256 Bytes
Velocidad de reloj	16 MHz
Módulo de radio	u-blox NINA-W102
Elemento de seguridad	ATECC608A
Unidad de medición inercial	LSM6DS3TR
Longitud	68.6 mm
Anchura	53.4 mm
Peso	25 g

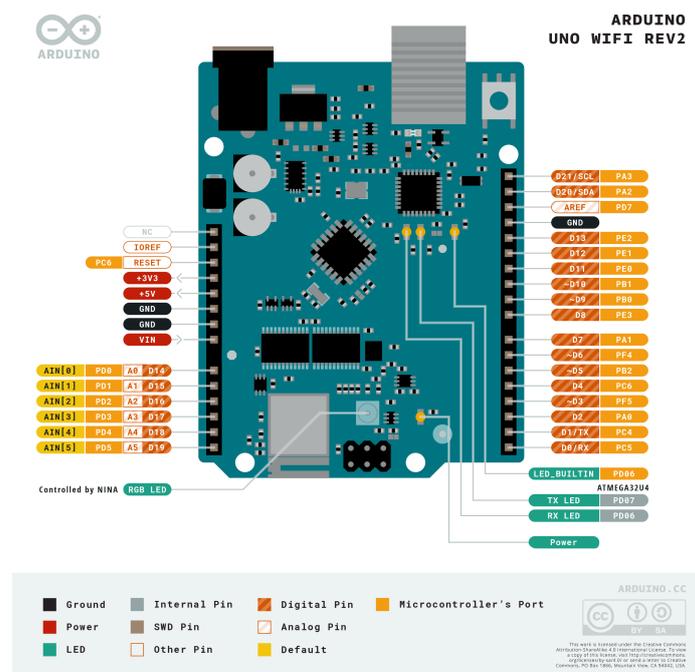


Figura 2: Esquemático Arduino UNO WiFi Rev.2

4.2 Sensor analógico de humedad de suelo

Más conocido como "Soil Moisture Sensor" es un sensor centrado en el cálculo de la humedad mediante la resistencia del material al que se adhiere, tierra en este caso. La resistencia es variable dependiendo de la humedad de la tierra: cuanto mayor sea esta, menor será la resistencia. Debido al constante contacto con el agua y tierra, de diferentes pHs y acidez, esto puede favorecer a la oxidación y al deterioro por el constante rozamiento del sensor con la tierra. Es por estos problemas que para evitar y ralentizar el efecto anteriormente explicado, se ha recubierto por una capa de niquelado químico capaz de reducir la oxidación de las conexiones y aumentar el grosor del sensor, retrasando el fallo catastrófico por destrucción de los filamentos. El funcionamiento es el siguiente a través de las sondas pasa una corriente que pasa a su vez por la tierra, sabiendo el valor de la corriente en el suelo se obtiene la resistencia y eso proporciona una aproximación de la humedad del suelo para poder el usuario crear límites con los que el sistema pueda accionar los diferentes sistemas de riego de los que disponga. La cantidad de agua hace variar la resistencia, es decir cuanto más agua se encuentre en la tierra menor resistencia y viceversa. A continuación, se muestra una tabla resumen con las características principales:

Tabla 2: Especificaciones del sensor analógico de humedad

Especificaciones		
Tensión alimentación	3.3 V o 5 V	
Tensión de señal de salida	0 - 4.2 V	
Corriente	35 mA	
Definición de pines:	Salida analógica	Azul
	Tierra	Negro
	Alimentación	Rojo
Longitud	60 mm	
Anchura	20 mm	



Figura 3: Sensor de humedad del suelo

Existen otros tipos de sensores de humedad de suelo que hacen uso de la capacidad en vez de la resistencia. Este tipo de sensores tienen mayor resistencia a la corrosión electroquímica. El sensor convierte su capacitancia en un valor analógico, esta capacidad varía con la cantidad de agua en la tierra (cuanta más agua haya menor capacidad tiene), para luego leerlo utilizando las entradas analógicas de Arduino.

4.3 Sensor analógico de temperatura LM35

Para la medición de la temperatura se usa un sensor de temperatura compuesto por el chip LM35, chip de temperatura lineal de tres terminales. El chip LM35 tiene una relación lineal tensión de salida/grados celsius ($10\text{mV}/^{\circ}\text{C}$), una característica no compartida en otros sensores, estos calculan la temperatura en Kelvin convirtiéndolo a celsius mediante la resta de una constante. Tiene una precisión de alrededor de $\pm 1/4^{\circ}\text{C}$, pudiendo variar a mayores temperaturas, permitiendo así no realizar calibraciones previas a su uso. Su bajo consumo de $60\ \mu\text{A}$ no genera casi auto calentamiento (alrededor de 0.1°C al aire libre). A continuación, se muestra una tabla resumen con las características principales:

Tabla 3: Especificaciones del sensor analógico de temperatura LM35

Especificaciones	
Tensión alimentación	3.3 V o 5 V
Desviación de la temperatura medida	$\pm 0.5^{\circ}\text{C}$
Rango de temperatura medida	-40 150 $^{\circ}\text{C}$
Temperatura operativa	-40 150 $^{\circ}\text{C}$
Corriente de operación	2.7 μA
Definición de pines:	Salida analógica Azul
	Tierra Negro
	Alimentación Rojo
Longitud	22 mm
Anchura	30 mm



Figura 4: Vista completa del sensor de temperatura

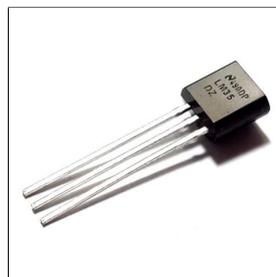


Figura 5: Circuito integrado LM35

4.4 Sensor analógico de luz ambiental

Para la medición de la luz ambiental se utiliza un sensor constituido por un sensor de tipo LDR (Light Dependent Resistor), que es una resistencia eléctrica que varía su valor según la cantidad de luz que incide sobre el sensor. Cuanta mayor es la luz que incide en el sensor menor es la resistencia y viceversa. Con esta resistencia se permite crear un sistema de rangos por los que se sabe cuánta luz incide y permite actuar acorde a dicha cantidad en nuestros proyectos. Al tratarse de un sensor analógico, la señal de salida es una tensión que depende de la densidad lumínica que recibe. A continuación, se muestra una tabla resumen con las características principales:

Tabla 4: Especificaciones del sensor analógico de humedad

Especificaciones		
Tensión alimentación	3.3 V o 5 V	
Tensión de señal de salida	0 - 5 V	
Rango de iluminación	1 a 6000 Luxes	
Tiempo de respuesta	15 μ s	
Definición de pines:	Salida analógica	Azul
	Tierra	Negro
	Alimentación	Rojo
Longitud	22 mm	
Anchura	30 mm	



Figura 6: Sensor de luz ambiental

4.5 Módulo relé digital 5A

El relé no es más que un interruptor controlado mediante un circuito eléctrico compuesto por una bobina y un electroimán. En este TFG se utiliza para controlar diferentes actuadores mediante comandos más en concreto el control de riego que se ha diseñado para el sistema. Los relés están conectados a las diferentes bombas de agua que se han dispuesto en el sistema para los dos modos de riego: riego continuo hasta alcanzar el límite o riego temporizado para humidificar. Cuando sea necesario activar un sistema de riego se activa el relé correspondiente. A continuación, se muestra una tabla resumen con las características principales:

Tabla 5: Especificaciones del módulo relé digital 5A

Especificaciones		
Tensión alimentación	5 V	
Máxima tensión controlable	150 VAC 24VDC	
Máxima corriente controlable	15 A	
Definición de pines:	Entrada digital	Verde
	Tierra	Negro
	Alimentación	Rojo
Longitud	38 mm	
Anchura	33 mm	

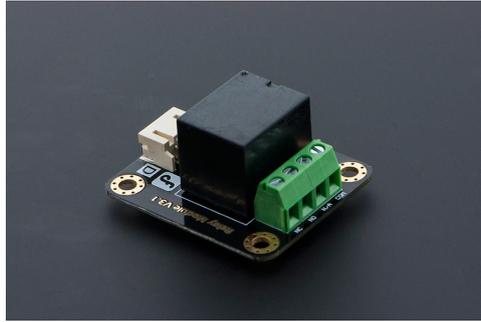


Figura 7: Relé digital 5A

4.6 Motor paso a paso 28BYJ-48

Para el desarrollo de los cultivos, el control de la luz incidente es prioritario debido a la necesidad de este pero en ciertas condiciones puede llegar a ser perjudicial. La incidencia de la luz por si misma no suele ser un problema, sin embargo, al añadir la variable de la temperatura y la humedad los hongos y enfermedades que pueden derivar de estas tienen una probabilidad alta de desarrollarse. Por ello, se considera la adición de un toldo retráctil u otro tipo de sistema para poder disminuir la incidencia de luz en los momentos adecuados, necesitando para ello un motor eléctrico. En pos de saber el motor que mejor se acomoda al sistema, se hizo una pequeña investigación [9] de los tipos de motores más utilizados que existen. Los tres tipos de motores son los siguientes: motor DC, servomotor y motor paso a paso; se hicieron pruebas para cada uno de estos, obteniendo así información de cuál se amoldaría mejor al proyecto.

1. Motor DC: Motor de corriente continua que convierte dicha corriente en un movimiento mecánico mediante la acción de campos magnéticos. Dicho motor necesita una alimentación externa para realizar el movimiento por su alto consumo. Hacer uso de la fuente de alimentación del Arduino genera tanto ruido que se generan interferencias con los demás sensores. Además el desplazamiento de estos motores no es exacto debido a su control mediante variable de tiempo.
2. Servomotor: Motor con movimiento de 0° a 180° . No necesita de una alimentación externa debido a su consumo reducido. Sin embargo su susceptibilidad a las interferencias y ruido provoca un movimiento indeseado; este movimiento puede ser reducido de dos formas: software o hardware. La forma hardware consiste en una resistencia de $1k\Omega$ de forma que se forme una pull-down para así inhibir las interferencias; la forma de software es desactivar el pin que controla el motor durante el uso de otros sensores. La fuerza de torque no es la suficiente para el caso práctico que se está realizando, además el rango de movimiento óptimo del servomotor no cumple los requisitos del sistema a utilizar.
3. Motor paso a paso: Motor que transforman los impulsos eléctricos en giros angulares discretos, siendo así capaz de girar un cierto número de grados dependiendo de la señal que se le proporcione [4]. Necesita de una fuente independiente del Arduino para no generar ruido en el sistema y que el funcionamiento sea óptimo. Las interferencias que otros sensores pueden generar en él, no alteran su comportamiento como puede ocurrir en el servomotor.

A raíz de las pruebas realizadas, se escogió el motor paso a paso 28BYJ-48 unipolar, motor de 4 bobinas controlado por ULN2003AN. Dicho controlador está compuesto por transistores Darlington y permite controlar cada bobina por separado mediante los pines digitales del Arduino. A continuación, se muestra una tabla resumen con las características principales:

Tabla 6: Especificaciones del motor paso a paso 28BYJ-48

Especificaciones		
Fases	4 por las 4 bobinas	
Par motor	34 N/m	
Consumo	55 mA	
Definición de pines:	Tierra	Negro
	Alimentación	Rojo
	Pin digital bobina 1	Blanco
	Pin digital bobina 2	Gris
	Pin digital bobina 3	Morado
	Pin digital bobina 4	Azul
Longitud	38 mm	
Anchura	33 mm	



Figura 8: Motor 28BYJ-48 con controlador ULN2003A

4.7 Micro bomba

Para dotar al sistema de regadío automatizado, se añade una pequeña bomba eléctrica de 5V ampliamente usada en proyectos DIY y acuarios. Se ha escogido esta bomba por su fácil uso y su conocimiento previo a la preparación del TFG, teniendo además un precio asequible y amplia disponibilidad. Esta bomba permite bombear alrededor de 100 litros/hora, siendo este valor más de lo necesario y pudiendo reducir la tensión necesaria para activarla. Dependiendo del tamaño de la instalación se requiere alterar dicho elemento para poder sustentar a las plantaciones con el agua requerida. Bomba diseñada para proyectos pequeños y precio asequible, utilizada frecuentemente en acuarios, riegos de plantas y otros proyectos a pequeña escala. Esta bomba suministra el agua necesaria para el proyecto y se amolda al tamaño del proyecto, no requiriendo nada más complejo. A continuación, se muestra una tabla resumen con las características principales:

Tabla 7: Especificaciones de la bomba

Especificaciones		
Tensión alimentación	5 V	
Máxima caudal	100 litros/hora	
Consumo máximo	100-200 mA	
Definición de pines:	Tierra	Negro
	Alimentación	Rojo
Diámetro interior de la salida de agua	4.5 mm	
Diámetro de entrada de agua	5 mm	
Longitud	42 mm	
Anchura	34 mm	



Figura 9: Bomba del sistema de riego

4.8 Gestor de energía solar 1.0

La gestión eléctrica del sistema debe ser balanceada para que el sistema esté constantemente encendido pero también debe tener un consumo bajo para que el sistema sea energéticamente independiente. Lograr esto puede ser un verdadero quebradero de cabeza, sin embargo el sistema Arduino tiene un gestor de energía con posibilidad de adición de batería y paneles solares. Además tiene variantes para poder elegir el más adecuado. Este elemento es llamado gestor de energía solar pero es mayormente conocido como "Solar power manager".

Es un gestor de energía capaz de alimentar tanto la placa Arduino como los demás sensores, mediante un panel solar desde 7V hasta los 30V máximo. Este actúa como fuente de alimentación suministrando hasta 2A de corriente de carga a una pila de 3.7V de LiPo que lo complementa. La pila se puede cargar de diferentes maneras a parte de la placa solar, por ejemplo mediante un adaptador AC o un cable USB. Además la placa tiene implementada la característica MPPT (Maximum Power Point Tracking) que permite mantener una tensión constante de salida bajo diferentes estados meteorológicos, además de maximizar la eficiencia del panel solar. Para ello, cuenta con 4 interruptores para especificar la tensión que se desea mantener, se puede elegir entre 9V, 12V y 18V. Para suministrar tensión tiene 3 convertidores DC-DC diferentes: 5V 1.5A, 3.3V 1A y 9/12V 0.5A, pudiéndose apagar mediante jumpers para que no haya corriente por esas salidas además de un switch para alternar entre los 9V y 12V. Estas son las tensiones que permiten alimentar los sensores que se utilizan en el proyecto de forma adecuada. Para alimentar la placa de Arduino esta equipado con un puerto USB para conectar un USB tipo B.

Como protección existen varios mecanismos como pueden ser: la protección de batería de Li; contra conexión inversa de batería/panel solar; contra sobrecalentamiento (además de una lámina de enfriamiento en la parte trasera) y contra cortocircuitos o sobreinten-

sidad, lo que convierte al dispositivo en un producto bastante seguro. La mayoría de las acciones y/o problemas que surgen, están indicadas con diferentes LEDs que se encienden en estos casos. A continuación, se muestra una tabla resumen con las características principales:

Tabla 8: Especificaciones del gestor de energía solar 1.0V

Especificaciones	
CI de gestión de energía solar	LTC3652
Máxima tensión (SOLAR IN)	7~30V
Corriente de carga (SOLAR IN)	2A
Tensión de corte	4.2V±1%
Tensión de MPPT	OFF/9V/12V/18V
Tensión USB IN	5V
Salidas de tensión	OUT1 5V 1.5A
	OUT2 3.3V 1A
	OUT3 9V/12V 0.5A
Protecciones	BAT IN
	OUT1-3/USB OUT
	SOLAR IN
	Sobrecarga
	Sobreintensidad
	Protección conexión invertida
Sobretensión de descarga	
Cortocircuito	
Sobreintensidad	
Sobrecalentamiento	
Reflujo	
Protección conexión invertida	
Longitud	68 mm
Anchura	78 mm

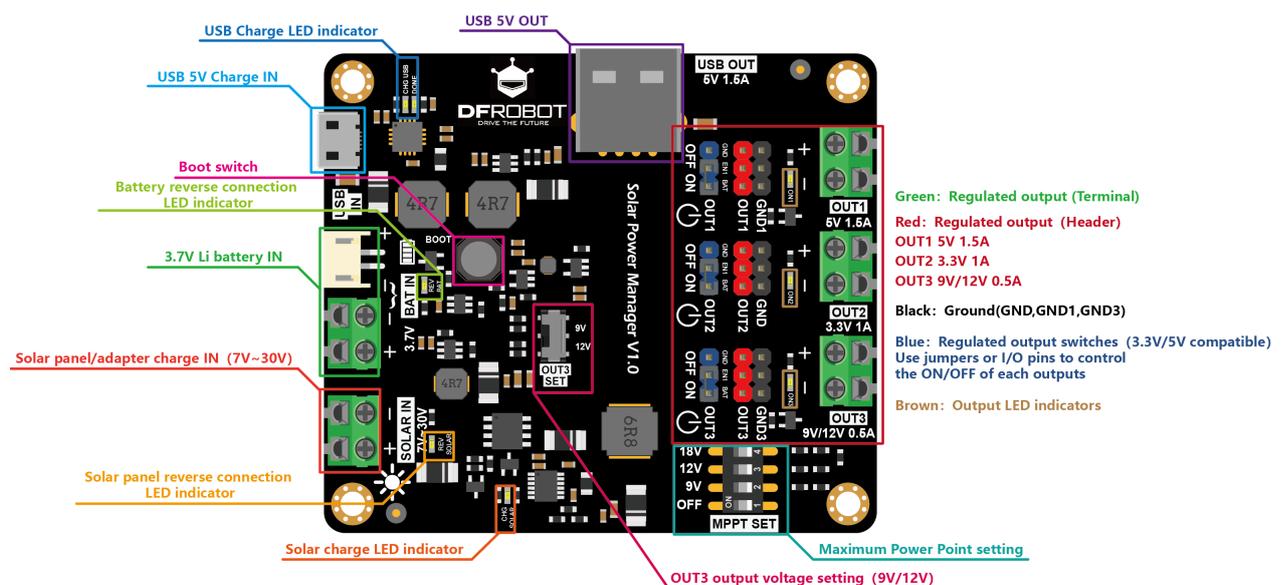


Figura 10: Gestor de energía solar

Lo más notable de este dispositivo es la manera de cargar la batería, ya que es lo más utilizado en este proyecto. La carga pasa por tres fases divididas de la siguiente manera:

- **Carga de goteo** : Cuando la tensión de la batería (al tener baja carga) es alta, cargarlo con una gran corriente al comienzo no es deseable por su alto calentamiento y por ende, pierde vida efectiva. Cuando su tensión es menor a 2.94V, entra en la siguiente fase de carga hasta llegar a dicha tensión.
- **Carga constante de corriente**: Una vez que llega a 2.94V se comienza a carga con una corriente constante de 2A.
- **Carga constante de tensión**: Cuando comienza a acercarse a los 4.2V (tensión de corte) el dispositivo comienza la fase de carga de tensión constante. Comienza a cargar con un tensión constante pero reduciendo poco a poco la corriente, llegando hasta los 200mA donde se para la carga.
- **Recarga automática**: Si la batería está completamente cargada, hasta que la tensión no se reduce hasta los 4.1V no se recarga, pero una vez llegado se comienza de nuevo el ciclo.

4.9 Panel solar de epoxi

La alimentación del sistema es un tema complejo de manejar debido a su variabilidad es por eso, que además de tener una batería para el sistema, se añaden cierto número de paneles solares para poder alimentar el sistema y recargar la batería de forma constante durante el día. Los paneles elegidos están compuesto de epoxi, material especializado en aislar eléctricamente los dispositivos y prevenir cortocircuitos, y sus conexiones son vía cable. Panel solar de DC 9V 2W con características para un buen efecto con luz reducida y alta eficiencia de conversión. Estos paneles de reducido tamaño, comparados con los de mayor tamaño, no tienen tanta potencia, sin embargo estos paneles son utilizados mayormente en proyectos de pequeña escala. Resistencia contra fuertes vientos y a prueba de nieve, además de fácil instalación. A continuación, se muestra una tabla resumen con las características principales:

Tabla 9: Especificaciones del panel solar de epoxi

Especificaciones		
Tensión funcionamiento	9 V	
Potencia máxima	2W	
Corriente de trabajo	220 mA	
Definición de pines:	Tierra	Negro
	Alimentación	Rojo
Espesor	2 mm	
Longitud	115 mm	
Anchura	115 mm	



Figura 11: Panel solar elegido

4.10 Batería LiPo LP-523450-1S-3

Para la alimentación del sistema los paneles solares no son suficientes al no suministrar energía suficiente en días nublados o de noche. Es por eso que una batería es necesaria, para ello después de una investigación se ha decidido usar una batería LiPo recargable. Batería recargable de polímero de litio de 3.7V y 950mAh, creada por Shenzhen BAK Technology Co.. La carga máxima recomendada es de 4.2V en caso de cargar más se podría dar el caso de errores en protecciones y su posterior destrucción debido a fallo catastrófico. Para evitar dichos problemas cuenta con varias protecciones antisobrecarga, antisobredescarga, antisobrecorriente de descarga y anticortocircuitos. Los tres pines permite la conexión al sistema y obtener la tensión tanto en negativo como en positivo. A continuación, se muestra una tabla resumen con las características principales:

Tabla 10: Especificaciones del panel solar de epoxi

Especificaciones		
Tensión nominal	3.7 V	
Tensión máxima	4.2 ± 0.05 V	
Tecnología batería	Polímero de Litio	
Máx. corriente de carga	930 mA	
Capacidad batería	950 mAh	
Definición de pines:	Negro	NTC
	Amarillo	Cable -
	Rojo	Cable +
Espesor	5 mm	
Longitud	33.5 mm	
Anchura	51 mm	



Figura 12: Paneles solares elegidos

4.11 Sensor de medición de color RGB TCS34725

Una aplicación que se desea añadir es la de conocer el color de las diferentes partes de las plantas. Permitiendo al usuario conocer el color de las diferentes partes, desde las hojas hasta el fruto, da la posibilidad de detectar posibles enfermedades en las hojas o conocer el grado de maduración del fruto. El sensor TCS34725 de CQRobot permite la medición del color mediante una matriz de 3x4 fotodiodos para rojo, verde, azul y sin filtro. Además cuenta con 4 conversores ADC para realizar la medición de los diodos. Las mediciones obtenidas permiten obtener el color en estilo RGB o Hexadecimal, sin embargo se usa RGB por su facilidad de visualización y adivinación por parte del usuario. El envío de datos se realiza mediante I2C bus de datos basado en el método de maestro-esclavo para la comunicación entre el dispositivo y periféricos. Las conexiones entre el Arduino y el sensor TCS34725 se realiza mediante SDA y SCL y se usa el pin D10 unido a la conexión LED del sensor para controlar el LED de iluminación para que no se encuentre constantemente encendido. A continuación, se muestra una tabla resumen con las características principales:

Tabla 11: Especificaciones del sensor de color TCS34725

Especificaciones		
Matriz fotodiodos	3x4	
Tensión funcionamiento	3.3 - 5V	
Corriente operativa	235 μ A	
Tipo de interfaz	I2C	
Capacidad batería	950 mAh	
Definición de pines:	GND	Negro
	Alimentación	Rojo
	SDA	Verde
	SDL	Azul
	INT	Amarillo
LED	Naranja	
Distancia de medición	20 mm	
Longitud	30.5 mm	
Anchura	23 mm	



Figura 13: Sensor de color TCS34725 de CQRobot

5 Desarrollo

Antes de comenzar con el desarrollo del sistema por elementos se deben conocer ciertos datos, para ser más exactos los datos de las plantaciones a controlar por el sistema de riego y tardo. Por ello se realiza una investigación sobre las diferentes plantas.

Cada planta es un mundo y sus cuidados son diferentes, es debido a esto que se ha hecho una pequeña base de datos en la que se dan diferentes consejos y especificaciones de riego para cada planta. Aun así no es un objetivo realizar un programa diferente para cada planta, por ello se hacen dos grupos diferentes que están divididos de la siguiente manera:

1. **Grupo A:** Grupo que contiene en su mayoría plantas que necesitan de un riego diario para obtener las mejores propiedades y crecimiento posibles. Estas plantas son las siguientes: lechuga/espínacas/hojas verdes, pepino, calabacines, pimientos, apio, coliflor, guisantes, brócoli, coles de Bruselas, vainas y tomate. La gran mayoría de estas plantas requieren de la luz en mayor medida a las del grupo B.
2. **Grupo B:** Grupo que se caracteriza por su alta cantidad de tubérculos y a diferencia del grupo A no necesita de riego diario para obtener mejores resultados. El grupo esta compuesto por rábano, chirimía, remolacha, zanahoria, coles, patatas y maíz. Este grupo de tubérculos soporta mejor las sequías y no necesitan tanta luz para desarrollarse,

Tabla 12: Grupos y respectivas plantas

Grupo A	Grupo B
Lechuga	Rábanos
Pepino	Chirimía
Calabacines	Remolacha
Pimientos	Zanahoria
Apio	Coles
Coliflor	Patatas
Guisantes	Maíz
Brócoli	
Coles de Bruselas	
Vainas	
Tomates	
Espinacas	

Para hacer más visibles y accesibles los datos se dispone de una tabla con diferentes apartados como es puntos críticos en los que sería mejor regar y notas de diferentes agricultores basadas en sus experiencias:

Tabla 13: Datos de plantaciones

Planta	Punto crítico	Agua por cada metro y medio	Notas
Lechuga Hojas verdes	Frecuentemente	9 litros a la semana	Mejor cosecha si no hay escasez de agua
Chirimía	Antes de estar muy seco	Máximo 14 litros a la semana	Regar cuando esta seco para mejor crecimiento
Pepino	Frecuentemente	14 litros a la semana	Regar bastante durante todo el proceso
Calabacines	Frecuentemente	14 litros a la semana	Regar durante el crecimiento y al formar la verdura
Cebolla	Al principio	14 litros a la semana	Restringir al madurar para mejor calidad
Remolacha	Antes de suelo muy seco	14 litros a la semana	Regar las hojas, las disminuye y mejora las raíces
Guisantes	Formación de flores, vaina y al recoger	14 litros a la semana	No regar mucho al principio para reducir follaje
Brócoli	Continuamente 4 semanas tras trasplantar	14-23 litros a la semana	Mejor cosecha sin escasez de agua
Coles de Bruselas	Continuamente 4 semanas tras trasplantar	14-23 litros a la semana	Aguantan bien si están estables
Pimientos	Regular	14-27 litros a la semana	Humedad equilibrada lo mejor
Patatas	Cuando tienen tamaño canica	14-27 litros a la semana	27 litros cada 10 días durante crecimiento
Rábanos	Necesita bastante, humedad consistente	14-27 litros a la semana	Mantener el suelo húmedo
Zanahoria	Antes de suelo muy seco	14 principio 27 litros maduras a la semana	Las raíces se pueden separar si se riega después de tener suelo seco
Apio	Frecuentemente	Mínimo 27 litros a la semana	Si las condiciones son secas regar a diario
Vainas	Formación de flores, vaina y al recoger	27 litros a la semana	Suelo seco al crearse la vaina suele dar mala cosecha
Coles	Frecuentemente en ambiente seco	27 litros a la semana	Si sufren de sequedad regar bastante 2 semanas antes de recoger
Coliflor	Frecuentemente	27 litros a la semana	Regar 27 litros antes de recolectar
Maíz	Cuando salen los pelos y al salir la mazorca	27 litros a la semana	La mazorca puede ser pequeñas si no se riega al salir la espiga
Tomates	3 o 4 semanas después del trasplante y cuando la flor o la fruta se hace	27 litros a la semana	Mucho agua mas tomates pero menos sabor

Estos dos grupos se usan para generar dos métodos de riego e iluminación diferentes.

Teniendo en cuenta los datos anteriormente expuestos se crea el sistema de riego. El sistema de riego varía dependiendo de las diferentes variables que se tienen en el sistema que son la humedad, temperatura y luz incidente. Para tener mejor control de estas se han dividido en diferentes rangos:

- H \uparrow : Se encuentra húmedo
- T \uparrow : Temperatura alta ($\geq 40^{\circ}\text{C}$)
- L \uparrow : Luz incidente alta
- H \sim : Ha llovido ayer
- T \sim : Temperatura media ($30-40^{\circ}\text{C}$)
- L \sim : Luz incidente media
- H \downarrow : No ha llovido ayer
- T \downarrow : Temperatura adecuada ($\leq 30^{\circ}\text{C}$)
- L \downarrow : Luz incidente baja

5.1 Control del sistema de riego

Es sistema de riego está controlado mediante el sensor de humedad de tierra descrito anteriormente y se tienen en cuenta las variables de temperatura y luz incidente para así evitar posibles enfermedades y hongos causados por las diferentes combinaciones posibles de estas. Con tal de evitarlo se hace una tabla de la verdad (Tabla 14), en esta se plasman las diferentes combinaciones que se pueden dar para cada grupo de hortalizas y se actúa según estas características. La siguiente tabla se ha realizado mediante información obtenida mediante internet y experiencia del autor:

Tabla 14: Tabla verdad de riegos

	H \uparrow	H \sim \wedge L \downarrow	H \downarrow \wedge L \downarrow	H \sim \downarrow \wedge L \uparrow \sim \wedge T \uparrow	H \sim \wedge L \uparrow \sim \wedge T \sim \downarrow	H \downarrow \wedge L \uparrow \sim \wedge T \sim \downarrow
A	×	✓	✓	×	✓	✓
B	×	×	✓	×	×	✓

Sin embargo para ciertos casos se procede al riego de maneras diferentes, es decir para ciertos casos se riega cierta cantidad de agua por sus características ideales para ello, sin embargo para otras se riega en menor medida para disminuir las posibilidades de enfermedades y además no poner en peligro la integridad de la planta. Es por eso que los siguientes casos tienen diferentes modos de riego:

Tabla 15: Caso especial de riego (\checkmark_- = riego reducido \checkmark_+ = riego abundante)

	H \sim \wedge L \uparrow \sim \wedge T \sim	H \sim \wedge L \uparrow \sim \wedge T \downarrow
A	\checkmark_-	\checkmark_+
B	×	×

Tabla 16: Más casos especiales de riego

	H \downarrow \wedge L \uparrow \sim \wedge T \sim	H \downarrow \wedge L \uparrow \sim \wedge T \downarrow
A	\checkmark_-	\checkmark_+
B	\checkmark_-	\checkmark_+

Por un lado en los casos que \checkmark_- este presente el riego es en menor medida para nutrir a la planta con la suficiente agua para no sufrir ningún tipo de daño, como podría ser por enfermedad o por el clima, para ellos se riega un tiempo determinado no superior al que sería el riego completo de la planta. Así se consigue paliar los efectos de la temperatura y luz solar en la planta, dando así la oportunidad de regar en mayor medida en otro momento del día. Por otro lado en los casos que esté \checkmark_+ se riega lo suficiente como para que el detector de humedad enterrado (10 cm aprox.) detecte que se ha regado.

5.2 Control de iluminación

La iluminación a la hora de cultivar una plantación es esencial, debido a la necesidad de la plantas de la luz solar para producir los nutrientes necesarios para crecer. Sin embargo, esta puede generar efectos adversos en las plantas si se dan ciertos parámetros. Cuando la luz incidente es alta y la temperatura es alta se suele dar un efecto de quemado en las plantas que perjudica su salud y producción. El control de la luz es un parámetro importante a tener en cuenta por ello se añade un toldo al prototipo, este se activa según los parámetros obtenidos a partir de los sensores de luz incidente y temperatura, la siguiente tabla de verdad resume los casos en los que se debe quitar o poner el toldo.

Tabla 17: Tabla verdad toldo

	$L \downarrow$	$L \uparrow \wedge T \uparrow \sim$	$L \sim \wedge T \uparrow$	$L \uparrow \wedge T \downarrow$	$L \sim \wedge T \sim \downarrow$
A B	Quitar	Poner	Poner	Quitar	Quitar

El toldo está compuesto de una tela con material opaco para reducir la luz solar incidente. Este material puede ser alterado para reducir en mayor o menor medida la incidencia o para aumentar su durabilidad frente a acontecimiento climatológicos.

5.3 Comienzo de implementación de sensores

Una vez explicado tanto los sensores como los sistemas de control de riego y luz y los datos de las diferentes plantaciones, se procede a describir el proceso de implementación de todos los sensores.

El funcionamiento del sistema sigue un orden. Dicho orden se ve reflejado en el siguiente diagrama de bloques que explica el modo de acción del sistema en cuanto a los sensores.

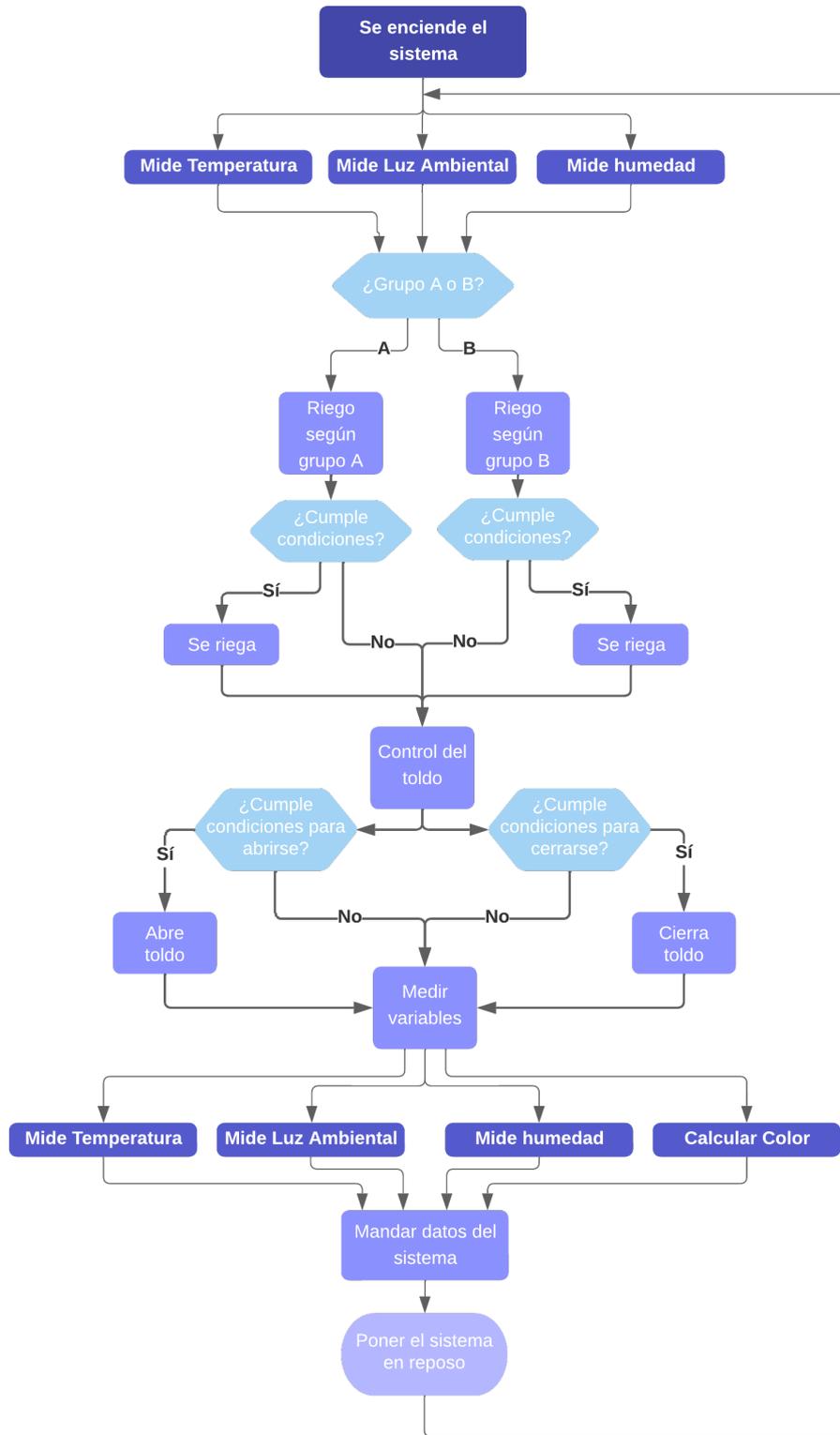


Figura 14: Acciones del sistema en orden

Antes de entrar más profundamente en la implementación de cada elemento se muestra el esquema completo del sistema como todos sus elementos.

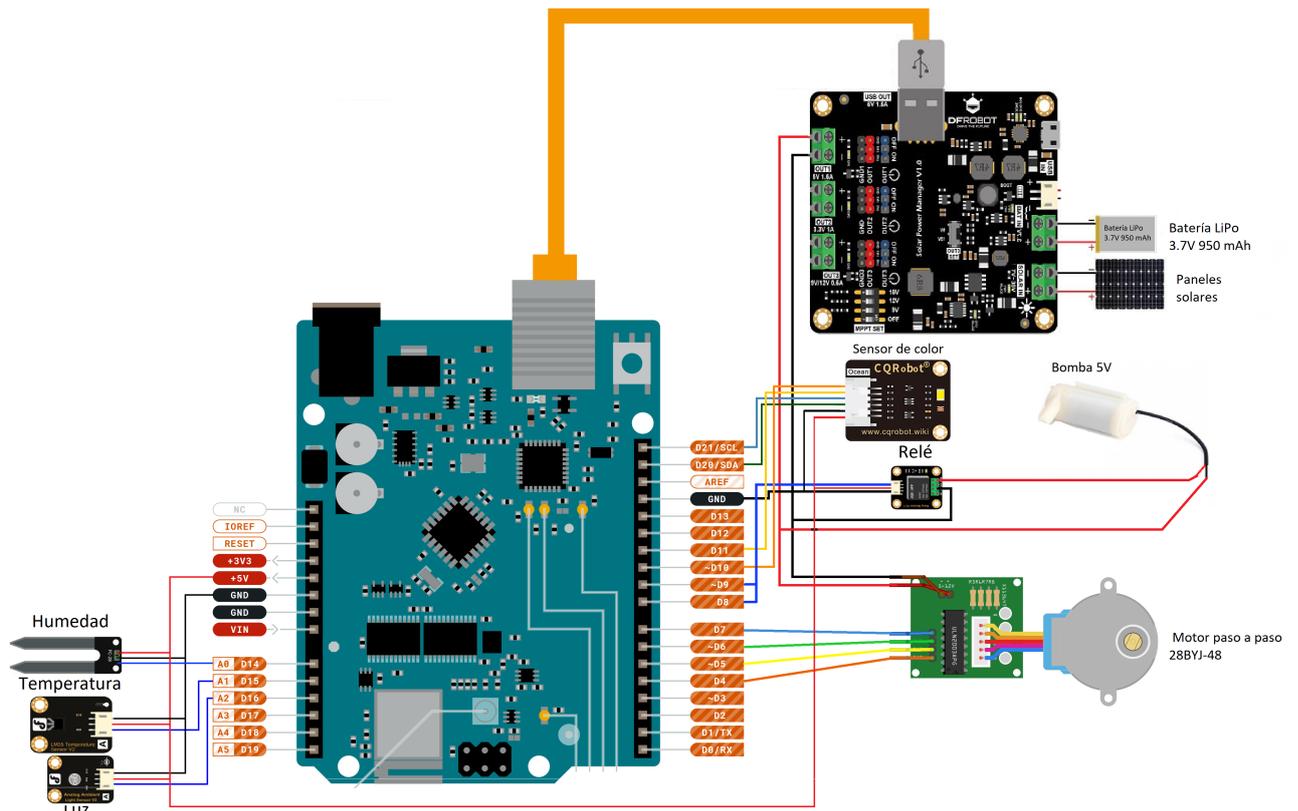


Figura 15: Arduino completo con todos los elementos conectados

En la figura 15 se puede observar todas las conexiones necesarias para el sistema completo. Las conexiones de alimentación de 5V son dos debido a las interferencias que surgen debido al motor y las bombas. Para simplificación la figura tan solo se ha visualizado un relé, sin embargo dos conexiones diferentes en digital son representadas, esto es porque son dos relés distintos cada uno con una bomba diferente pero con las mismas conexiones de alimentación.

En cuanto a los pines usados se dispone de la siguiente tabla para poder visualizar más fácilmente los pines que se están usados.

Tabla 18: Pines de conexión usados

Pin	Dedicado
Pines analógicos usados: 3	Humedad: 1
	Temperatura: 1
	Luz: 1
Pines digitales usados: 8	Motor paso a paso: 4
	Relés: 2
	Sensor de color : 2
Pines especiales	SDA y SCL
Pin de alimentación	1
Pin GND	1

Teniendo este conocimiento se puede proceder a las implementaciones de los sensores.

5.4 Implementación sensor de humedad

La implementación del sensor de humedad se debe colocar en la tierra donde se encuentre la planta a monitorizar, para obtener un mejor resultado hay que disponerlo a una distancia aproximada de 10 cm desde la base de la tierra. Esto es debido a que la humedad es preferible que llegue a las raíces de la planta para que estas puedan absorberlo mejor y que el crecimiento sea el adecuado. Se conecta a la placa Arduino al pin A0 para poder calcular los valores. El valor obtenido del pin A0 es una tensión variable que depende de la resistencia que el sensor genera en la tierra. Para obtener ese valor de tensión se debe usar el siguiente trozo de código:

```
//Humedad
humedad = analogRead(A0);
```

Figura 16: Código de cálculo de humedad

En este código se crea una variable llamada humedad en la que se hace una lectura del pin A0 con el comando analogRead obteniendo la tensión deseada. La tensión varía entre 0 y 950 donde cero es un ambiente seco y 950 es el de estar sumergido en agua. Esta medida se usa en varios apartados del código como variable para llevar un registro de la humedad máxima del día. Este registro es usado para saber si ya se ha regado o llovido el día en el que se hace el chequeo, para así saber si se necesita regar o no. Para ello se usa el siguiente trozo de código:

```
//////////HumedadMaximaDía//////////
if ((dia - duc == 0) && (humedaddia < Hmdfinal)) {
    humedaddia = Hmdfinal;
}
```

Figura 17: Humedad máxima del día

Mediante la librería TimeLib.h que permite obtener un cálculo aproximado del tiempo que a transcurrido desde el comienzo del sistema mediante un cálculo simple, se sabe si ha pasado un día desde el último chequeo y el día actual (duc = día último chequeo) y se comprueba si la humedad que tiene al final del chequeo es mayor que la que ya se había obtenido el mismo día. En caso de haber pasado un día se genera la variable humedad del día anterior (hda) para ser utilizada en el sistema de riego que se explica después. El código utilizado es el siguiente:

```
//////////HumedadDiaAnterior//////////
if (dia - duc == 1) {
    hda = humedaddia;
    humedaddia = Hmdinicial;
}
```

Figura 18: Modificación variable humedad del día anterior (Hmdinicial = humedad inicial; humedaddia = humedad máxima del día)

Con la humedad se han realizado 3 grupos para facilitar el control del sistema de riego:

- H ↑ : Se encuentra húmedo
- H ~ : Ha llovido ayer
- H ↓ : No ha llovido ayer

Esta clasificación, con la de temperatura e iluminación, permiten crear una tabla de la verdad con los casos de uso del proyecto. El límite que se usa para el sistema de riego es 700.

5.5 Implementación sensor temperatura

El sensor encargado de calcular la temperatura es el LM35, este sensor calcula la temperatura ambiente con una precisión de $\pm 0.5^{\circ}\text{C}$. Está compuesto por el circuito integrado LM35, de consumo pequeño y poco autocalentamiento. Tiene un rango de medición adecuado para el proyecto ($0 \sim 150^{\circ}\text{C}$), ya que no se va a necesitar medir más de 60°C y a partir de los 0°C difícilmente se llega en primavera o en verano. Este sensor tiene un consumo muy reducido, ayudando así a mantener más fácilmente la independencia eléctrica. Para conectarse se utiliza el pin A1 y se alimenta con 5V y GND. Obtener la medida de la temperatura se realiza mediante la siguiente manera:

```

////////////////////////////////////Temperatura////////////////////////////////////
temperatura = analogRead(A1);
millivolts = (temperatura / 1023.0) * 5000;
celsius = millivolts / 10;

```

Figura 19: Cálculo de la temperatura

El valor que el sensor obtiene del pin A1 se encuentra codificado en los 10 bits por el conversor analógico-digital que proporciona la placa de Arduino por eso el valor obtenido es una asignación de la tensión de funcionamiento a un número entero entre 0-1023. Para obtener la tensión se debe utilizar la ecuación que se presenta en la Figura 19, el valor de 1023 viene de $(2^n - 1)$ donde $n = 10$ es en número de bits del conversor y 5000 es la tensión de referencia en milivoltios. Obtenidos los milivoltios si se busca en el datasheet [8] del sensor se sabe que tiene una sensibilidad de $1^{\circ}\text{C}/10 \text{ mV}$. Haciendo uso de esta relación se puede obtener la temperatura en grados Celsius como se hace en la Figura 19. Teniendo conocimiento de la temperatura se usa en varios apartados del código para diferentes tareas, estas tareas son las de sistema de regado y control de iluminación.

La temperatura se divide en 3 grupos diferentes para reducir los casos de uso a un número asequible de opciones, los grupos son los siguientes:

- T ↑ : Temperatura alta ($+40^{\circ}\text{C}$)
- T ~ : Temperatura media ($30\text{-}40^{\circ}\text{C}$)
- T ↓ : Temperatura adecuada (-30°C)

Esta clasificación, con la de humedad e iluminación, permiten crear una tabla de la verdad con los casos de uso del proyecto. Para generar esta clasificación en código se ha realizado de la siguiente forma:

```

//Temperatura en rangos
if (celsius < 30) {
  tmpbaja = true;
  tmpmedia = false;
  tmpalta = false;
} else if (celsius > 30 && celsius < 40) {
  tmpbaja = false;
  tmpmedia = true;
  tmpalta = false;
} else {
  tmpbaja = false;
  tmpmedia = false;
  tmpalta = true;
}

```

Figura 20: Rangos de temperatura en código (tmp = temperatura)

El hecho de separarlos así permite realizar las condiciones del sistema de riego con mayor facilidad como se ve en el apartado del sistema de control de riego mas adelante.

5.6 Implementación sensor de luz ambiental

La implementación del sensor de luz permite calcular la luz incidente, variable necesaria para el sistemas de control de riego e iluminación. El sensor está compuesto por un fototransistor PT550F, el cual dependiendo de la luz incidente genera un valor de resistencia diferente, obteniendo del pin A2 el valor de tensión consecuente. Está tensión no es una unidad de medida internacional por ello se ha buscado en diferentes datasheet el modo de conversión para obtener en unidades de luxes, sin embargo la imposibilidad de comprender el datasheet [10] al completo por el hecho de estar en un lenguaje desconocido como es el Chino y la carencia de datasheets en algún idioma comprensible con dicho procedimiento [11] se ha procedido de la siguiente manera. La tensión obtenida es análoga a la tensión del sensor de temperatura así que para obtener la tensión real se debe desasignar el valor obtenido mediante el conversor analógico-digital y obtener una tensión con la que poder obtener un valor en unidades con las que poder trabajar. Igual que en el sensor de temperatura se multiplica por la tensión de referencia en milivoltios (5000 milivoltios en este caso) y dividirlo por 1023. Obtenida la tensión para obtener los luxes se debe realizar una regla de tres donde para 5V se obtienen 6000 luxes, así pues, mediante esta regla de tres se obtiene el número de luxes captado. En código se procede de la siguiente manera:

```

////////////////////Luz////////////////////////////////////
luz = analogRead(A2);
voltiosluz = (luz / 1023.0) * 5000;
Luxes = voltiosluz*6000/5000;

```

Figura 21: Código sensor de luz ambiental

Como los otros sensores ya implementados, esta medida se usa para diferentes tareas como la del control de riego e iluminación. Para facilitar su uso se clasifican en tres grupos:

- L ↑ : Luz incidente
alta
- L ~ : Luz incidente
media
- L ↓ : Luz incidente
baja

Esta clasificación, con la de humedad y temperatura, permiten crear una tabla de la verdad con los casos de uso del proyecto. Para generar esta clasificación en código se ha realizado de la siguiente forma:

```

if (Luxes <= 3000) {
  luzbaja = true;
  luzmedia = false;
  luzalta = false;
} else if (Luxes > 3000 & Luxes <= 5500) {
  luzbaja = false;
  luzmedia = true;
  luzalta = false;
} else {
  luzbaja = false;
  luzmedia = false;
  luzalta = true;
}

```

Figura 22: Rangos de iluminación en código

Los rangos se han clasificado de esta forma a través de ir probando a diferentes horas del día que iluminación recibía el sensor y se ha llegado a la conclusión de que estos rangos pueden ser una de las combinaciones posibles para el proyecto.

5.7 Implementación del relé

Los relés utilizados en este proyecto son: el relé digital 5A con conexión gravity de DFRobot y un relé sin amplificación de corriente. El relé de DFRobot tiene incluida la protección y amplificador de corriente necesario para conectar a la placa Arduino. En el caso de no tener este tipo de relé se necesitaría una resistencia de $1K\Omega$, un diodo 1N4007 [1] y un transistor NPN 2N2222 [13]. Estos elementos son necesarios para aumentar la corriente que se le proporciona al relé para poder activarse. Para ello se deben colocar de la siguiente manera:

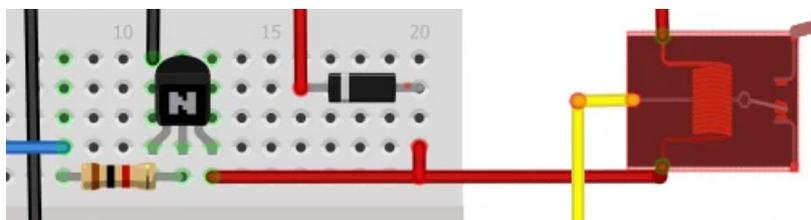


Figura 23: Conexiones para aumentar corriente en relé

Como se puede observar en la Figura 23 las conexiones son las siguientes: el diodo 1N4007 se conecta a la bobina electromagnética del relé, con esto conseguimos conducir la corriente en inversa cuando el transistor es apagado generando una protección de picos de tensión o flujo contra corriente; el colector del transistor al relé, el emisor a tierra y base mediante una resistencia de $1K\Omega$ al pin digital que se desee, el pin D8 en este caso.

El sistema de flujo de corriente que el relé permite deja conectar el actuador de dos formas diferentes NC (normalmente cerrado) o NO (normalmente abierto). Para ello tiene tres conectores NC, NO y COMmon, mediante estas conexiones se puede conectar de forma que mientras este el relé desactivado funcione, para ello se hace uso de NC y COM o en caso de querer activarlo cuando se accione el relé sería con NO y COM.

En este proyecto se va a hacer uso de los relés para el sistema de riego. El riego se divide en dos modos: riego para refrescar y riego de formación. Para ello se utilizan dos bombas diferentes y un relé para cada una (también se podría hacer uso de una electroválvula pero tiene mayor coste y consumo).

Las conexiones que se deben realizar para controlar las bombas son en normalmente abierto así que se conecta: COM a la tierra de la bomba, la alimentación de la bomba a 5V y pin NO a tierra. Se usa una fuente de alimentación independiente para no interferir con la fuente interna del Arduino y así no añadir ruido a esta, pues genera un ruido bastante alto e interfiere con los demás sensores.

5.8 Implementación de la bomba

Como ya se ha comentado en la implementación del relé, la bomba va conectada al relé para poder activarse dependiendo del sistema de riego que se necesite. Para ello se usa la bomba de características anteriormente mencionadas. En el prototipo la bomba se dispone en un tanque con agua cerca del recipiente donde se encuentra la planta para permitir el riego y facilitar al usuario rellenar el recipiente. La conexión de la bomba es la siguiente: COM a la tierra de la bomba, la alimentación de la bomba a 5V y pin NO a tierra. La alimentación usada en las bombas y el motor es una obtenida directamente del gestor de energía solar para reducir ruido inducido por estas. Para bombear el agua para regar se usan tubos transparentes de 5.54 mm de diámetro interior, se realizan unas pequeñas incisiones a lo largo del tubo y se tapa la salida para mantener una presión constante, haciendo así que de las incisiones supuren agua de forma controlada siendo este el primer modo de riego; el segundo modo es otro canal de agua en el que se insertan pequeños nebulizadores para generar una niebla suficiente para humedecer las plantas en momentos de temperatura media e incidencia solar media o alta.

El código necesario para encender la bomba es el siguiente código:

```
digitalWrite(9, HIGH);  
digitalWrite(9, LOW);
```

Figura 24: Código necesario para controlar la bomba

La primera línea permite encender el relé de forma que se encienda la bomba previamente dispuesta en NO y la segunda línea desactiva el relé apagando así la bomba.

Sabiendo el funcionamiento de la bomba se puede programar el código del sistema de riego. Hay dos códigos diferentes dependiendo del grupo elegido, regidos por la tabla 14. El código es el siguiente:

```

//////////////////////////////////// Riego A //////////////////////////////////////
if (grupoA && (regarA || !SeHaRegado) && ((luzbaja)
|| (luzalta && tmpbaja) || (luzmedia && tmpbaja))) {
  while (humedad < humedadlimite) {
    digitalWrite(9, HIGH);
    delay(1000);
    humedad = analogRead(A0);
  }
  digitalWrite(9, LOW);
  delay(1000);
  SeHaRegado = true;
} else if (grupoA && (regarA || !SeHaRegado) &&
((luzalta && tmpmedia) || (luzmedia && tmpmedia))) {
  digitalWrite(8, HIGH);
  delay(1000);
  digitalWrite(8, LOW);
  delay(1000);
} else if (grupoA && !regarA) {
  digitalWrite(9, LOW);
  delay(1000);
}
}

```

(a)

```

//////////////////////////////////// Riego B //////////////////////////////////////
if (grupoB && (regarB || !SeHaRegado) && ((luzbaja)
|| (luzalta && tmpbaja) || (luzmedia && tmpbaja))) {
  while (humedad < humedadlimite) {
    digitalWrite(9, HIGH);
    delay(1000);
    humedad = analogRead(A0);
  }
  digitalWrite(9, LOW);
  delay(1000);
  SeHaRegado = true;
} else if (grupoB && (regarB || !SeHaRegado) &&
((luzalta && tmpmedia) || (luzmedia && tmpmedia))) {
  digitalWrite(8, HIGH);
  delay(1000);
  digitalWrite(8, LOW);
  delay(1000);
} else if (grupoB && !regarB) {
  digitalWrite(9, LOW);
  delay(1000);
}
}

```

(b)

Figura 25: (a) Código grupo A (b) Código grupo B

Seguendo la Tabla 14 se sabe si hay que regar mucho, como en el primer segmento de ambos códigos, solo un tiempo establecido, como en el segundo fragmento o no regar como en la última parte.

En el primer segmento se activa la bomba y se deja un segundo hasta comprobar la humedad de nuevo, se repite hasta superar el límite de humedad y se convierte el bool `SeHaRegado` a `true`, considerando así que no se volverá a pasar por este ni por el segundo proceso. En el segundo fragmento se enciende la bomba y se espera cierto tiempo para que los mini-aspersores puedan humidificar el ambiente sin regar completamente la plantación. Por último tan solo se comprueba que la bomba esté apagada.

5.9 Implementación de motor paso a paso 28BYJ-48

El motor paso a paso 28BYJ-48 se usa para controlar el toldo o sistema de protección solar de la plantación. El motor está conectado al controlador ULN2003AN para poder controlar las cuatro bobinas del interior. Este está incrustado en una placa con los conectores necesarios para conectar el motor a ella y con resistencias como protección, además tiene los pines necesarios para la alimentación. El controlador es alimentado por una fuente de alimentación independiente, en este caso se usa la misma fuente que las bombas. La conexión de los cuatro pines permite controlar las cuatro bobinas y generar pasos de 1/4 de vuelta, para la conexión con Arduino se utilizan los pines 1N1, 1N2, 1N3 y 1N4 conectados respectivamente a los pines D4, D5, D6 y D7. Cada pin permite controlar cada una de las bobinas mediante código y permite al usuario dar saltos entre bobinas. En la parte de código se ha de hacer unas preparaciones previas para comenzar a utilizar el motor. Estas preparaciones son las siguientes:

```

#include <AccelStepper.h>
#define motorPin1 4
#define motorPin2 5
#define motorPin3 6
#define motorPin4 7
#define MotorInterfaceType 8
AccelStepper stepper = AccelStepper(MotorInterfaceType, motorPin1, motorPin3, motorPin2, motorPin4);
void setup() {
    stepper.setMaxSpeed(1000);
}

```

Figura 26: Código de preparaciones del motor

En orden según el código de la Figura 26, la primera línea agrega las librerías necesarias para controlar el motor. Las siguientes 4 son las definiciones de los pines que controlan las bobinas. En la línea 6 se elige la interfaz del motor. La séptima línea representa la generación de una instancia para el motor que estamos utilizando los parámetros a suministrar por orden son: interfaz del motor y pines. La línea que se encuentra dentro del setup limita la velocidad máxima del motor. En cuanto al movimiento, se puede mover en ambas direcciones, como se puede ver en la Figura 27. El control del toldo se lleva a cabo mediante el siguiente código:

<pre> //////////////////////////////////Toldo////////////////////////////////// if (toldocerrado && (luzbaja (luzalta && tmpbaja) (luzmedia && tmpmedia) (luzmedia && tmpbaja))) { while (stepper.currentPosition() != -numeropasos) { stepper.setSpeed(-1000); stepper.runSpeed(); } digitalWrite(7,LOW); digitalWrite(6,LOW); digitalWrite(5,LOW); digitalWrite(4,LOW); toldocerrado = false; toldoabierto = true; } else if (toldoabierto && (luzbaja (luzalta && tmpbaja) (luzmedia && tmpmedia) (luzmedia && tmpbaja))) { toldocerrado = false; toldoabierto = true; } </pre>	<pre> } else if (toldocerrado) { toldocerrado = true; toldoabierto = false; } else { while (stepper.currentPosition() != numeropasos) { stepper.setSpeed(1000); stepper.runSpeed(); } digitalWrite(7,LOW); digitalWrite(6,LOW); digitalWrite(5,LOW); digitalWrite(4,LOW); toldocerrado = true; toldoabierto = false; } </pre>
(a)	(b)

Figura 27: (a) Apertura y mantener abierto (b) Cierre y mantener cerrado

Donde si se cumplen las condiciones se abre o cierra el toldo y después se apagan todas las bobinas (pines) para reducir el consumo. También se mantiene en espera mientras no se cumplan las condiciones de la acción contraria a la que se encuentra, es decir si esta cerrado hasta cumplir las de abrir y viceversa.

5.10 Implementación gestor de energía solar

Este dispositivo tiene una implementación rápida. El gestor se conecta mediante un USB tipo B a la placa Arduino para poder alimentarla, se conecta la batería LiPo en el conector BAT IN y la placa solar en SOLAR IN. Esto suministra la corriente necesaria para alimentar los sensores y Arduino a la vez. El panel solar recarga la pila según el proceso comentado en el apartado de Diseño. El conector de 5V se une mediante cables a la protoboard para realizar las conexiones necesarias.

5.11 Implementación de panel solar

Previamente a implementar los paneles en el sistema ambos paneles solares son unidos en paralelo para así aumentar la corriente de carga y aumentar la capacidad de carga de los paneles. Estos son soldados en paralelo y se le añade un cable extra para poder conectarlos al gestor. Para que no haya cortocircuitos se añade un tubo de plástico termorresistente consiguiendo así que no haya un posible escape de corriente. Los paneles solares son conectados al gestor de energía solar mediante la conexión de SOLAR IN, cable negro al negativo y cable rojo al positivo, consiguiendo así que se cargue la batería o que el sistema sea alimentado. El siguiente paso es poner los paneles en una zona en la que la luz de lo máximo posible. Para ello, es mejor colocarlos por encima de la altura máxima de crecimiento de la planta a monitorizar.

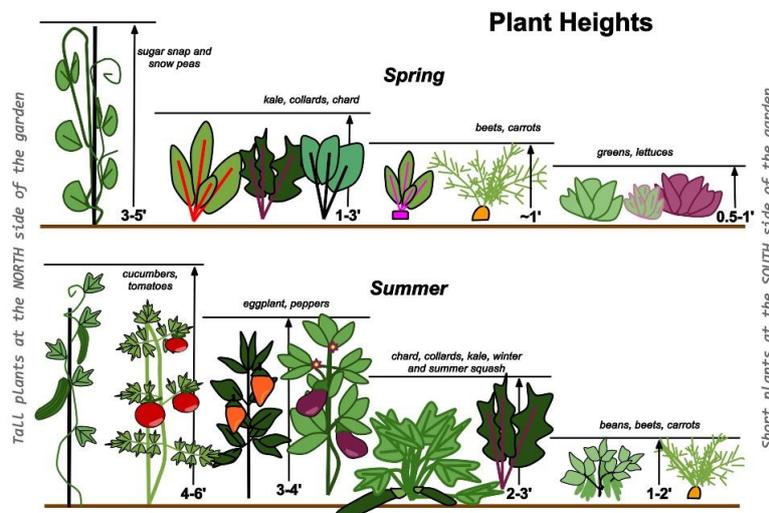


Figura 28: Tamaño de las plantas en pies (pie = 30.48 cm)

5.12 Implementación de la batería LP-523450-1S-3

La alimentación del Arduino es un tema algo complejo debido a las diferentes variables que se pueden ir añadiendo durante un proyecto. Por eso las opciones son amplias y muchas son las baterías que se nos presentan en el mercado. Sin embargo para este proyecto se hizo una valoración de que tipo de batería es la más adecuada, para ello se hizo una diferenciación entre todas las opciones, teniendo en cuenta los siguientes datos:

- Pila 9V : Pila con tensión suficiente para alimentar el Arduino, fáciles de adquirir y usar, aunque baja densidad energética e intensidad de corriente máxima. Esto último genera que el gasto sea rápido y la batería se agote rápidamente. De precio reducido pero no recargables (en general).
- Pilas AA de 1.5V: Al usar cuatro pilas AA en serie se puede llegar a la tensión necesaria para hacer funcionar el sistema, sin embargo aún teniendo mayor densidad de carga que la pila de 9V se sigue agotando rápidamente y en general al no ser recargables genera un problema en el sistema, ya que no lo mantiene activo largos periodos. Sin embargo, es fácil de adquirir, pudiendo usarse para pequeños proyectos.

- Pilas 1.2V recargables: Necesario tener cinco pilas de 1.2V para alimentar el sistema. En este caso al ser recargables aumenta el número de usos de estas y puede mantener el sistema activo mayor tiempo. Respecto al anterior caso el coste sube y adquirirlo sigue siendo relativamente sencillo.
- Banco de baterías USB de 5V: Salida regulada a 5V así que no hay que preocuparse por la necesidad de acoplar alguna otra batería para alimentar el sistema. Permite alimentar tanto a la placa Arduino como los demás sensores. En caso de necesitar alterar la posición de la batería entre proyectos es una opción muy adecuada. Sin embargo el coste aumenta en comparación a las otras opciones.
- Baterías de Niquel-metalhidruro (NiMh): Baterías de Niquel-metalhidruro interconectadas entre ellas y con conector añadido. Pudiendo variar la tensión dependiendo de la batería escogida desde los 6V hasta en general los 9.6V, con estas tensiones se puede alimentar el sistema fácilmente. Tiene una densidad energética media-alta, capacidad desde 300 hasta 5000 mAh. El coste de las baterías es alto comparado con los anteriores, además de la necesidad de un cargador especial de alto coste. Adecuado para proyectos que requieran un consumo energético medio/alto.
- Batería de Polímero Litio (LiPo): Batería compuesta de polímero de Litio con un rango de diferentes tensiones. Densidad de energía más alta de todas, capacidades desde los 500 mAh incluso superando los 12000 mAh. Batería recargable pero de coste alto. Como ventaja, tiene la capacidad de proporcionar intensidades muy altas, para hacer funcionar los motores con mayor necesidad de corriente. Su mayor desventaja viene con su dificultad de manejo debido a que en caso de manipularlo como no es debido pueden ocurrir fallos catastróficos, pero en caso de usarse debidamente es una batería muy efectiva y usada para proyectos más avanzados.

Teniendo todo esto en cuenta las opciones se reducen a las baterías recargables. De entre ellas las más rentables son las de Niquel-metalhidruro y las de LiPo por su media-alta densidad de energía y su alta capacidad. Suponiendo que el consumo puede ser alto en cuanto al uso de motores y bombas debido a los dos tipos de riego y al mecanismo del toldo, se ha llegado a la conclusión de que la batería que mejor se amolda al sistema es la LiPo, por sus altas capacidades e intensidades. Además el sistema elegido para la gestión de la energía cuenta con las protecciones necesarias para controlar la carga de la batería, un problema que requiere de control por su probabilidad a fallar. Por ello, se ha escogido la batería LiPo LP-523450-1S-3 con sus 3.7V de tensión nominal es perfecto para el gestor de energía solar y con su capacidad de 950 mAh se espera un funcionamiento continuo. La conexión al gestor de energía es de la siguiente manera: la batería al contar con tres pines y el gestor solo necesitar dos, se crimpean los cables de forma que el pin rojo de la batería se conecte al pin positivo de BAT IN y el pin negro al pin negativo de BAT IN. Con esto se conecta la batería al gestor y se consigue cargarla mediante las placas solares.

5.13 Implementación de envío de datos

El envío de datos es una aplicación opcional del sistema que es de gran ayuda para el usuario, dando información del sistema al usuario. Varios son los métodos para enviar datos mediante Arduino, para este sistema se considera que los mejores métodos serían GPRS o WiFi. Estos dos métodos permiten mandar los datos mediante mensajes de texto en el caso del GPRS y mediante diferentes métodos a través de la red WiFi. Se ha considerado que el uso de GPRS generaría mayores costes que el caso del WiFi debido a la implicación de un nuevo sensor para poder transmitir los mensajes. La placa Arduino UNO WiFi Rev.2 tiene incluido un módulo WiFi de nombre u-blox NINA-W102 que permite las conexiones tanto por Bluetooth como por redes WiFi. Para este caso solo se usa la conexión a redes WiFi y un método de envío, de extendido uso, muy relacionado con IoT (Internet of Things). Se trata de una red de dispositivos que comparten información normalmente obtenida mediante sensores, llamada MQTT (MQ Telemetry Transport). El método de mensajería MQTT se basa en la tecnología publicador-suscriptor donde el esclavo, el sistema Arduino en este caso, sube los datos a un servidor o broker donde los datos son gestionados y persistidos durante un periodo de tiempo o hasta volver a ser alterados. Este método de mensajería evita la pérdida de datos debido a la desconexión del publicador y permite obtener el valor en cualquier lugar. Los brokers contienen diferentes métodos de almacenamiento de datos de menos a más encriptados, esto permite manejar todo tipo de datos desde los poco relevantes hasta otros de mayor relevancia. Para alterar el nivel de encriptación hay diferentes protocolos enumerados de la siguiente manera:

- **1883** : MQTT, sin encriptar
- **8883** : MQTT, encriptado
- **8884** : MQTT, encriptado, requiere certificación
- **8080** : MQTT mediante WebSockets, sin encriptar
- **8081** : MQTT mediante WebSockets, encriptado

El orden hace referencia a la dificultad de interceptar o alterar los datos. La lista de posibles brokers es amplia, algunos de los más conocidos son: HiveMQ, MyQttHub, EMQX Cloud, Mosquitto, CrystalMQ y Hark Platform. De los brokers anteriormente mencionados, se ha elegido Mosquitto por ser el mejor valorado y fácil de usar. Para utilizar este broker se debe acceder a su página web mediante Arduino.

Para poder comenzar a utilizar el módulo y sus funciones se deben añadir las librerías `ArduinoMqttClient.h` y `WiFiNINA.h`. Finalmente, se debe incluir un archivo en el que añadir el nombre de la red Wifi a utilizar y la contraseña, de modo que se añade una protección extra de privacidad:

```
#define SECRET_SSID "Nombre"  
#define SECRET_PASS "Contraseña"
```

Figura 29: Archivo para nombre y contraseña de la red

La implementación del código en Arduino se divide en dos partes. La primera parte se trata de la conexión del Arduino a la red WiFi y la conexión al broker, el código es el siguiente:

```
#include <ArduinoMqttClient.h>
#include <WiFiNINA.h>
#include "arduino_secrets.h"
char ssid[] = SECRET_SSID;      // Nombre de la red
char pass[] = SECRET_PASS;     // Contraseña de la red
WiFiClient wificlient;
MqttClient mqttClient(wificlient);
const char broker[] = "test.mosquitto.org";
int port = 1883;
const char kmqtfgT[] = "kmqtfgT";
const char kmqtfgH[] = "kmqtfgH";
const char kmqtfgL[] = "kmqtfgL";

void setup() {
  Serial.begin(9600);
  // Intentando conectarse al WiFi:
  Serial.print("Intentando conectar a WPA SSID: ");
  Serial.println(ssid);
  while (WiFi.begin(ssid, pass) != WL_CONNECTED) {
    // Fallado, reintentar
    Serial.print(".");
    delay(5000);
  }
  Serial.println("Estas conectado a la red");
  Serial.println();
  Serial.print("Intentando conectar al MQTT broker: ");
  Serial.println(broker);
  if (!mqttClient.connect(broker, port)) {
    Serial.print("MQTT conexión fallida! Error code = ");
    Serial.println(mqttClient.connectError());
    while (1);
  }
  Serial.println("Estas conectado al MQTT broker!");
  Serial.println();
}
}
```

Figura 30: Código de conexión a WiFi y Broker

Lo siguiente es crear un cliente para poder conectarse a una red WiFi y de la misma manera un cliente para conectarse al broker. Se sigue definiendo el broker al que se va a conectar, en este caso Mosquitto, y el tipo de encriptación deseada, al ser datos de poco valor se ha elegido el puerto 1883. Las siguientes constantes que se crean son los topics o nombres por los que se guarda/busca el dato en el broker, en este caso, al ser temperatura, humedad y luz se han creado los siguientes topics: *kmqtfgT* (Temperatura), *kmqtfgH* (Humedad) y *kmqtfgL* (Luz). Una vez creados y añadidos se procede a conectar el sistema a la red WiFi y al broker. En la primera parte del void setup se conecta al WiFi probando indefinidamente hasta conectarse. En cuanto al broker, se conecta mediante el puerto 1883. Mediante estos pasos se completa la conexión a la red WiFi y al broker.

La segunda parte del código se centra en el envío de los datos:

```

////////Envio del los datos////////
mqttClient.poll();

mqttClient.beginMessage(kmqttfgT);
mqttClient.print(Tmpfinal);
mqttClient.endMessage();

mqttClient.beginMessage(kmqttfgH);
mqttClient.print(Hmdfinal);
mqttClient.endMessage();

mqttClient.beginMessage(kmqttfgL);
mqttClient.print(Luzfinal);
mqttClient.endMessage();

Serial.println();

```

Figura 31: Código necesario para enviar datos

El primer comando *mqttClient.poll()* pone al sistema dispuesto a recibir o mandar diferentes mensajes al broker. Para enviar los datos al broker se debe crear un mensaje mediante *mqttClient.beginMessage(Topic)*, añadir el contenido del dato mediante escritura y terminar el mensaje con *mqttClient.endMessage()*, con esto el dato es enviado al broker y almacenado en el para que un suscriptor pueda acceder a él por su topic.

Por último, para poder visualizar los datos, se hace uso de una aplicación de móvil diseñada para recibir datos de IoT por MQTT, su nombre es "IoT MQTT Panel". Con esta aplicación se pueden visualizar los datos fácilmente mediante gráficos, tablas o imágenes. Para ello se debe seleccionar el broker, "test.mosquitto.org" en este caso, y añadir diferentes paneles con el nombre del dato que se desea monitorizar.

Varias son las opciones de customización y está en constante alerta ante cambios en los datos del broker. Para este caso se utilizan diferentes paneles como pueden ser: un termómetro, un registro de texto, un gráfico de barras para la humedad, botones para realizar acciones (explicadas en la siguiente sección) y un panel que muestra el color.



Figura 32: Paneles aplicación

La configuración del proyecto y de los paneles se realiza de la siguiente manera:

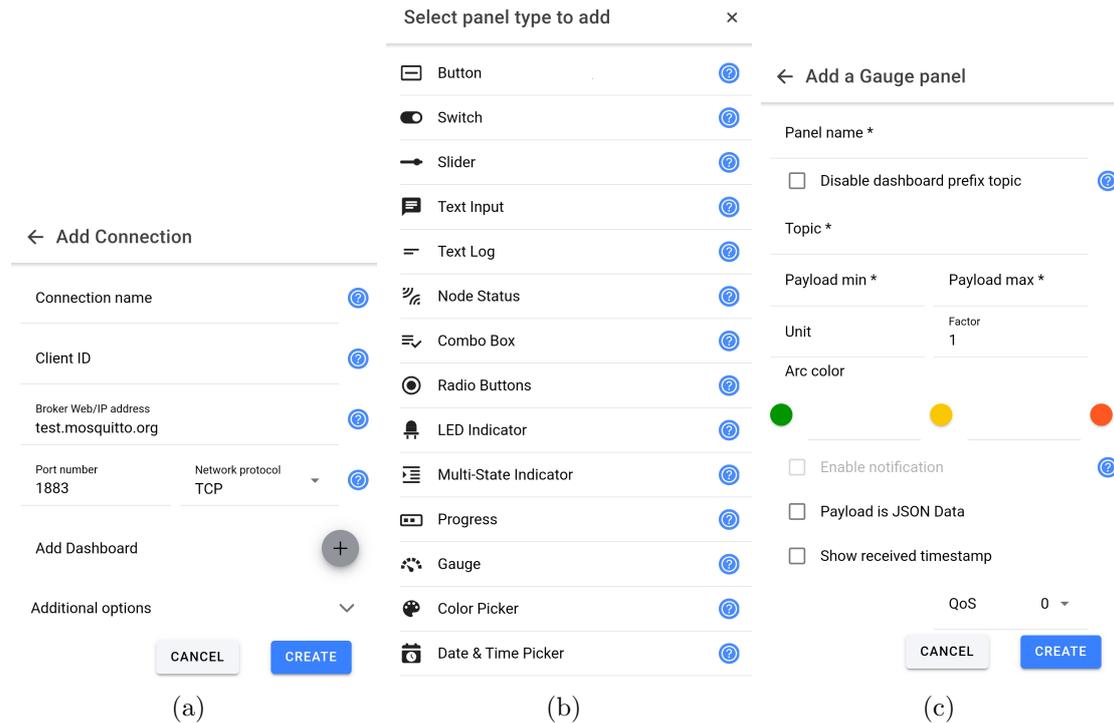


Figura 33: (a) Creación de suscriptor (b) Paneles disponibles (c) Ejemplo de panel

Mediante el código y la aplicación se ha cumplido la relación publicador-suscriptor y permite el envío de datos vía internet.

5.14 Implementación sistema de riego y toldo manual

El sistema control de riego y toldo automatizado convierte al sistema en un dispositivo independiente de todo tipo de interacción humana, sin embargo, no permite al usuario activar el sistema de riego a petición. Con esa idea en mente, se implementa un sistema control de riego y toldo manual mediante comunicación MQTT controlable desde la aplicación.

Para ello se debe crear un dato específico que permita ser alterado, de manera que se pueden variar las acciones de regar y abrir/cerrar el toldo. La comunicación entre el sistema y el usuario se realiza mediante la aplicación IoT MQTT Panel, la cual permite generar un panel que simula el comportamiento de un botón donde el usuario elige el topic a crear y el contenido de este. El topic generado en este caso se trata de *kmqtfgrs*, el cual dependiendo de la acción deseada varía entre 1, 2 o 3. Esos valores se usan para definir que acción se debe realizar: el uno está asociado al sistema de riego y acciona el protocolo de riego predeterminado del grupo elegido A o B, el dos y tres están asociados al sistema de control del toldo y las acciones son cerrar y abrir el toldo, respectivamente. Una vez generados los tres botones que alteran el valor de *kmqtfgrs*.

Realizar la lectura del topic *kmqtfgrs* se consigue mediante la suscripción a él mediante código. Para poder suscribirse, se debe usar *mqttClient.subscribe(topic)*, con este trozo de código realizamos la unión entre el sistema Arduino y el broker, quedando a la espera de un cambio en el topic. Sin embargo la alteración del topic se realiza en cualquier momento de la ejecución del sistema, provocando que no siempre se pueda obtener el dato deseado, para solucionar este inconveniente se utiliza una función capaz de ser ejecutada

cada vez que se de un cambio en el valor de dicho topic, independiente del proceso principal. Esa acción es posible mediante el comando *mqttClient.onMessage(topic)*, este permite al sistema actuar de forma que cuando el proceso principal no se encuentre realizando otra acción, se lleve a cabo la adquisición del dato. Esto se realiza mediante el siguiente fragmento de código:

```
mqttClient.onMessage(MensajeMqtt);
mqttClient.subscribe(kmqttfgRS);
void MensajeMqtt() {
  while (mqttClient.available()) {
    RSMaNUal= String((char)mqttClient.read()).toInt();
    if(RSMaNUal == 1){
      RegarManual = 1;
    }else if(RSMaNUal == 2){
      ToldoManual = 0;
    }else if(RSMaNUal == 3){
      ToldoManual = 1;
    }
  }
}
```

Figura 34: Código de adquisición de datos y acciones

Como se puede apreciar en la Figura 34, se ha creado el proceso MensajeMqtt que se trata de la acción a llevarse a cabo al alterar el dato *kmqtfgrs*. Como anteriormente se ha mencionado, al alterar el dato *mqttClient.onMessage(topic)* da comienzo a la acción y se procede a obtener la información del topic. Se genera una constante llamada RSMaNUal que se utiliza para diferenciar entre la acción a realizar. Para ello se llama al comando *mqttClient.read()*, que devuelve el número contenido, que debe ser transformado a int mediante la función *.toInt* de la clase String. Obtenido el valor, se puede saber la acción deseada y se altera el valor de las constantes RegarManual y ToldoManual. Estas constantes son utilizadas en las siguientes acciones:

```
//////////RegadoManual//////////
if(RegarManual == 1){
  if(grupoA){
    while (humedad < humedadlimite) {
      digitalWrite(9, HIGH);
      delay(1000);
      humedad = analogRead(A0);
    }
    digitalWrite(9, LOW);
    delay(1000);
    SeHaRegado = true;
  }else if(grupoB){
    while (humedad < humedadlimite) {
      digitalWrite(9, HIGH);
      delay(1000);
      humedad = analogRead(A0);
    }
    digitalWrite(9, LOW);
    delay(1000);
    hda = 0;
    SeHaRegado = true;
  }
  RegarManual = 0;
}
```

Figura 35: Código riego manual

El código utilizado en el riego manual no es otro que un fragmento del sistema de riego automático para ambos grupos. Este código se trata de la parte donde se riega hasta detectar la humedad necesaria mediante el sensor de humedad. Se han utilizado estos fragmentos por el hecho de que al realizar el riego de esta manera se consigue que el regado manual y el automatizado coexistan sin problema. Por otro lado el código del sistema de control del toldo manual es el siguiente:

```

////////////////////////////////ToldoManual////////////////////////////////
if(ToldoManual == 1){
  if (toldocerrado) {
    while (stepper.currentPosition() != -numeropasos) {
      stepper.setSpeed(-1000);
      stepper.runSpeed();
    }
    toldocerrado = false;
    toldoabierto = true;
    digitalWrite(7,LOW);
    digitalWrite(6,LOW);
    digitalWrite(5,LOW);
    digitalWrite(4,LOW);
  }
}

}

}
else if(ToldoManual == 0){
  if(toldoabierto){
    while (stepper.currentPosition() != numeropasos) {
      stepper.setSpeed(1000);
      stepper.runSpeed();
    }
    toldocerrado = true;
    toldoabierto = false;
    digitalWrite(7,LOW);
    digitalWrite(6,LOW);
    digitalWrite(5,LOW);
    digitalWrite(4,LOW);
  }
}
}

```

Figura 36: (a) Abrir el toldo manualmente (b) Cerrar el toldo manualmente

Parecido al control de riego manual, el código del toldo ha sido extraído del sistema de control de la versión automática. Se han elegido los fragmentos donde se abre y cierra respectivamente y se activan dependiendo del valor de ToldoManual. Estos dos nuevos modos de toma de acciones permiten al usuario decidir en cualquier momento del proceso de monitorización regular las veces de regado o la luz incidente en la plantación, dando así al usuario la opción a interactuar con el sistema de forma remota.

5.15 Implementación sensor de color TCS34725

El sensor de color sirve para obtener un color aproximado de la parte a monitorizar, en este caso se usa para saber en la mayoría de los casos el color del fruto, con el objetivo de conocer el grado de maduración en el que se encuentra. Las conexiones pueden ser de diferente forma dependiendo de la placa Arduino de la que disponga el usuario. Las placas Arduino UNO en adelante contienen pines dedicados a esta comunicación I2C en la parte final de los pines digitales.

Si se usa una placa Arduino de generaciones más antiguas los pines de SDA y SCL suelen estar en los pines A4 y A5 respectivamente, sin embargo se recomienda buscar en el datasheet y comprobarlo.

En cuanto a las conexiones, el sensor cuenta con 6 pines, alimentación, tierra, SDA, SCL, INT y LED. Alimentación y tierra van conectados a 5v y GND respectivamente, alimentando el sistema. SDA y SCL van conectados a los pines dedicados de la placa Arduino. Por último LED e INT van conectados a los pines D10 y D11 respectivamente, de esta manera se controla el LED que ilumina la zona a monitorizar. Para obtener la medida se enciende el LED para obtener el color aproximado incluso en la oscuridad.

La implementación en código parte del código suministrado por la distribuidora del sensor. Está compuesto por cinco archivos que preparan al sensor para las mediciones, tanto en RGB como en Hexadecimal, sin embargo con la medida de RGB es suficiente,

así que para reducir el tamaño del código se procede a eliminar todo lo relacionado con la medición en hexadecimal. La parte del código con la que se obtiene el color es el siguiente:

```

////////////////////////////////Color////////////////////////////////
digitalWrite(10,HIGH);
delay(500);
rgb=TCS34725_Get_RGBData();
RGB888=TCS34725_GetRGB888(rgb);
int R = RGB888.R;
int G = RGB888.G;
int B = RGB888.B;
String RGBinicial = "rgb(";
String RGBfinal = RGBinicial + R + "," + G + "," + B + ")";
Serial.println(RGBfinal);
digitalWrite(10,LOW);

```

Figura 37: Código para la obtención del color

Para comenzar, se enciende el LED para obtener mayor precisión del color. Después se obtienen los diferentes valores de los fotodiodos, y mediante el siguiente comando lo convertimos a valores de RGB. Para obtener el valor de cada color se accede a la variable *RGB888* y se elige la componente de color deseado (R, G o B). Una vez obtenidos, se disponen los valores de manera sencilla de visualizar y de un modo en el que la aplicación IoT MQTT Panel pueda visualizarlo. La codificación de color aceptada por la aplicación IoT MQTT Panel es: *rgb(R,G,B)*. Este dato se puede enviar a la aplicación y obtener una visualización rápida del color obtenido.

5.16 Consumo

El consumo se ha de tener en cuenta para poder obtener el sistema con independencia energética. Para ello, se calcula el consumo durante una hora, además de la corriente que los paneles solares proporcionan al sistema. El consumo es el siguiente:

Tabla 19: Consumos de acciones

Acción	Consumo	Tiempo en segundos
Arduino solo	80 mA	3550
Realizando mediciones	89 mA	8
Envío datos	162 mA	1
Abrir y cerrar toldo	510 mA	24
Bombas y relé	272 mA	12
Setup del Arduino	110 mA	5

Una aproximación adecuada para saber si el consumo puede ser compensado con las placas solares es realizar la media de consumo por hora. Haciéndolo se obtiene un consumo medio de 82.26 mAh. Para saber si los paneles son suficientes se hace una media de los días soleados, nublados y de noche, que hay al año en el País Vasco, Vizcaya para ser más específico. Se obtiene la siguiente tabla:

Tabla 20: Días según el tiempo

Estado	Número de días
Soleado	62.3
Nublado	166.4
Noche	136.3

Y se han obtenido los siguientes valores para los paneles solares:

Tabla 21: Potencia de los paneles solares

Situación lumínica	Corriente
Luz artificial indirecta	1.05 mA
Luz artificial directa	2.15 mA
Luz natural noche	0.25 mA
Luz natural nublado	50 mA
Luz natural directa	75 mA

Mediante la tabla 21 y 20 se puede obtener una media aproximada de la corriente generada con un valor de 35.69 mAh. Contando con una batería de 950 mAh y con un consumo medio de 82.26 mAh, la batería duraría aproximadamente 11.55 horas. Esto no permitiría al sistema mantenerse encendido en situaciones de día nublado o noche. El valor medio tan bajo de corriente generada por los paneles no permitiría recargar la batería de forma eficiente. Esto depende de muchas variables donde la climatología tiene gran importancia, además Vizcaya es una de las comunidades con menor número de horas de sol, por debajo de Cantabria, así que no favorece al sistema. Por ello, se deben ajustar estos elementos si se desea un sistema independiente. Para el caso de Vizcaya, se recomendaría usar una batería con capacidad para alimentar el sistema al menos 3 días, por las posibles rachas de lluvias persistentes y días nublados. Para ello se necesitaría una batería de alrededor de 6000 mAh, para poder alimentar durante 3 días consecutivos sin necesidad de carga. Para recargar se necesitarían paneles solares de mayor eficiencia y que por lo menos tuvieran 1.5 veces la media del consumo de la batería, es decir, unos 124 mAh, así se conseguiría sustentar el sistema y recargar la batería de forma consistente.

Uno de los objetivos del proyecto es disminuir el consumo del Arduino al máximo, consiguiendo reducir costes en materiales como la batería o paneles. La reducción encontrada consiste en poner el Arduino en deepSleep con una librería específica. Sin embargo, el microcontrolador ATmega4809 del Arduino Uno WiFi Rev.2 no permite poner el sistema en reposo. La complejidad del cambio de la placa Arduino al detectar dicho posible hizo que no fuera viable. Poner en reposo el sistema permitiría reducir en gran medida el consumo, hasta los 16 mA, reduciendo drásticamente la media de consumo de 83.6 mAh a 19.15 mAh. Esta reducción afecta drásticamente a los requisitos para mantener el sistema encendido indefinidamente, en este caso con una batería de 1400 mAh y unos paneles que pudiesen generar 29 mAh se conseguiría el efecto deseado. Con los elementos de los que se dispone, los cuales tienen 950 mAh de capacidad y 35.69 mAh de media de generación de corriente, sería suficiente, ya que aunque la capacidad de la batería sea menor se compensa con el aumento en la generación de carga.

6 Prototipo

Obtenido el sistema, se desea comprobar su funcionamiento mediante la monitorización de un prototipo que simule una plantación de tamaño pequeño. El prototipo ha pasado por diferentes etapas de desarrollo obteniéndose diferentes bocetos. Dichas etapas se pueden ver en las siguientes figuras:

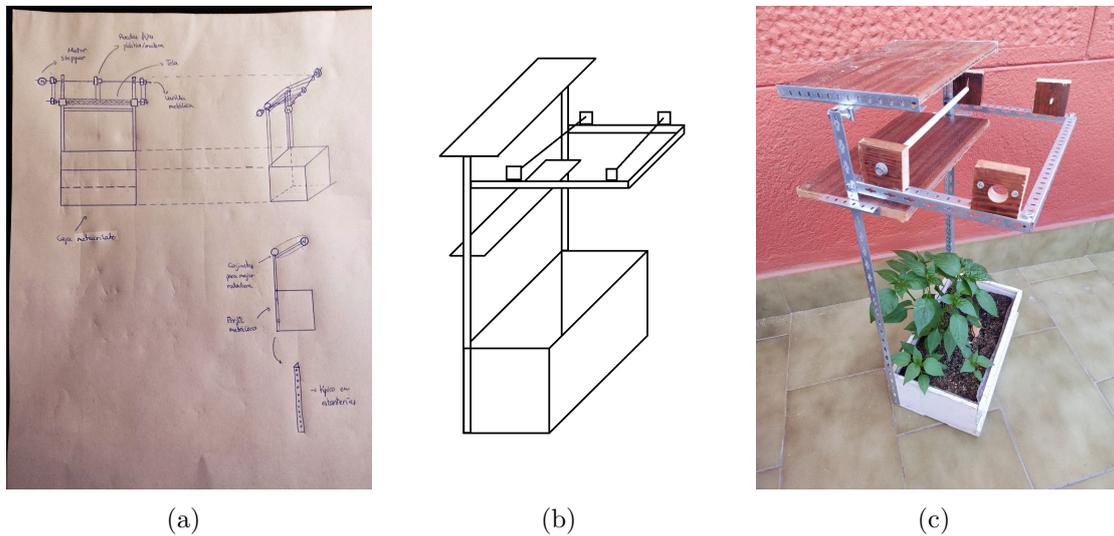


Figura 38: (a) Boceto original (b) Boceto final (c) Prototipo

En el boceto (Figura 38) se puede ver que el sistema está compuesto por una caja en la que disponer la planta con la tierra necesaria, dos perfiles metálicos en las esquinas para suministrar un soporte para estantes para el toldo, los paneles solares y dispositivos varios. Ambos perfiles están unidos por otro perfil para reforzar el armazón. Los paneles solares, el sensor de temperatura y el sensor de luz se disponen en un estante colocado en la parte superior para recibir la luz lo máximo posible. El gestor de energía solar se coloca en una madera en un punto más bajo del prototipo para cubrir lo máximo posible para reducir el contacto con las posibles lluvias.

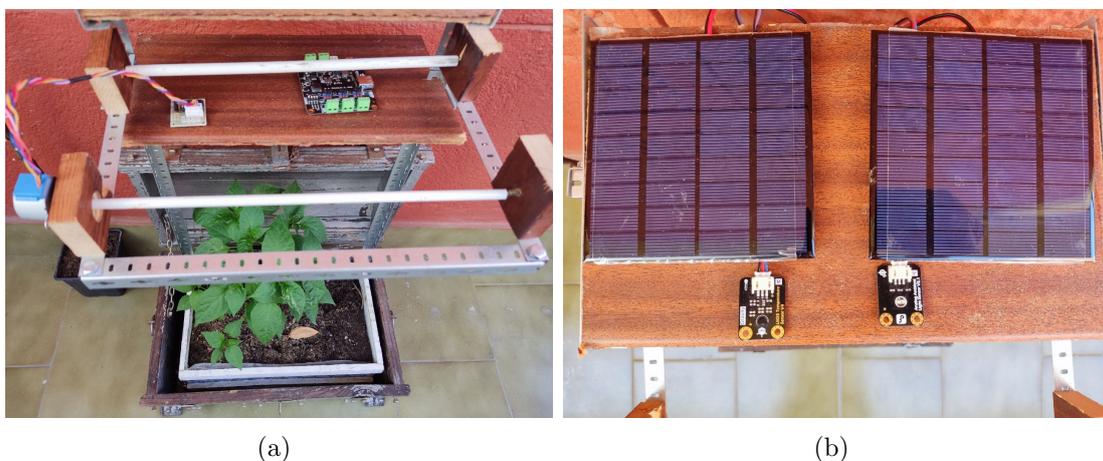


Figura 39: (a) Balda del gestor y motor (b) Paneles solares y sensor de temperatura y luz

Para disponer el toldo se añaden otros dos perfiles metálicos y se unen en su extremo con otro para fijarlos y aumentar la estabilidad. El toldo se dispone encima de los dos primeros perfiles mediante una barra metálica giratoria, mediante un trozo de madera unido al perfil, mientras que en el otro extremo se dispone de una barra metálica unida al motor paso a paso para girar el toldo. Se gira de forma antihoraria para abrirlo y horaria para cerrarlo.



Figura 40: Toldo del prototipo

El sensor de humedad está enterrado en la tierra hasta estar completamente cubierto, de manera que pueda monitorizar la humedad aproximada de las raíces de las plantas.

Por otra parte, el sensor de luz se coloca en la parte superior de la estructura donde no se encuentra ningún problema de interferencias con ningún material o sombra que se pudieran generar, a la vez, se coloca el sensor de temperatura para que reciba todos los rayos solares directamente y así poder calcular la temperatura con mayor precisión.

Para suministrar el agua necesaria, se coloca la bomba de agua en un recipiente que contenga el agua necesaria para regar la planta el tiempo requerido o en caso de no poder ser así un recipiente recargable para que al sistema no le falte el agua. La bomba requiere de un sistema de riego y para ello se usan tubos de PVC para distribuir el sistema de riego en dos formas de riego: a la base de la planta o riego para humedecer la planta por encima de la misma.



Figura 41: (a) Riego directo a la base (b) Riego por encima de la planta

El sensor de color dispone de un cable largo para poder moverlo a gusto y cambiarlo a la parte que se desea saber el color. Para el caso de los frutos se recomienda añadir una malla elástica para ajustarse a la forma del fruto dejándole así crecer sin problemas.

Todos los sensores están conectados al Arduino mediante cables de longitud larga para poder alterar la posición de estos en caso de necesidad y para dar la posibilidad al usuario de monitorizar otras plantas cercanas con mayor facilidad. El prototipo final tiene la siguiente forma:

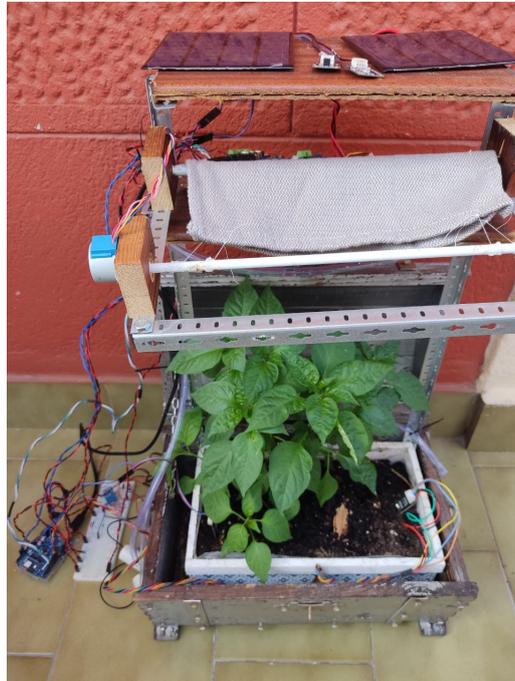


Figura 42: Prototipo completo

7 Costes de proyecto

En este apartado se hace un análisis del coste que conlleva el desarrollo y diseño del proyecto. En él, se tiene en cuenta todos los medios utilizados para dicho desarrollo y los materiales necesarios para construir el prototipo. Varios de los sensores y materiales necesarios han sido suministrados por la universidad y los sensores restantes han sido comprados.

Para los precios de los materiales se obtienen de diferentes distribuidores para observar la forma más barata de obtenerlos:

Tabla 22: Precios de material

Material	Precio
Arduino	46.70 €
Sensor analógico de humedad de suelo	2.69 €
Sensor analógico de luz ambiental	2.59 €
Sensor analógico de temperatura LM35	9.46 €
Módulo relé digital 5A	2.59 €
Motor paso a paso 28BYJ-48	4.19 €
Micro bomba	3.99 €
Gestor de energía solar 1.0V	38.73 €
Cables	13.99 €
Batería LiPo	17.14 €
Sensor de color	14.99 €
Total	157.06 €

El coste total del sistema ascendería a 157.06 €. El coste se puede reducir si se altera el gestor de energía solar 1.0V por Solar Power Manager 5V (DFR0559) donde se reduce de 38.73 € a 7.87€, sin embargo se usó el primero para realizar pruebas. Siempre se puede reducir el coste buscando en otros distribuidores de sensores o eligiendo otro tipo de sensores para realizar las mismas tareas, pero no se asegura que el código o el montaje sea igual.

8 Conclusiones

La elaboración del proyecto del sistema y la elaboración del documento de desarrollo e investigación ha permitido una aproximación a la elaboración de un producto útil y que cumple las características deseadas, además de un acercamiento a la investigación y desarrollo que se llevaría a cabo en una empresa tecnológica. El desarrollo de este proyecto me ha permitido valorar el trabajo que implica realizar un proyecto, desde la consolidación de la idea, hasta lograr un producto finalizado. Además, el proyecto podría contar con varias utilidades de cara al futuro, como pueden ser el empleo del sistema desarrollado en plantaciones o incluso ser utilizado como herramienta de aprendizaje para alumnos con ambiciones de programación en C y de montar un sistema electrónico con sus propias manos. El proceso de documentación llevado a cabo en la memoria me ha permitido observar de primera mano lo importante que es tener un orden y realizar explicaciones claras y concisas del proceso completo si se desea obtener los mismos resultados al replicar el proyecto.

Además, con este trabajo he podido ahondar en el sistema Arduino, que me fue presentado en 2^o año de universidad mediante el grupo IEMakers, donde realizamos un sistema muy simplificado del sistema de riego que ha sido extendido en este TFG. Durante ese año conocí la versatilidad de Arduino y los proyectos que con este se podían llevar a cabo. Por eso el desarrollo a un nivel más complejo me ha permitido conocer en más profundidad dicho sistema y formarme para poder diseñar y desarrollar sistemas más complejos en el futuro.

Por otra parte, la programación del software me ha acercado más a un lenguaje que era desconocido para mí, como es C, sin embargo, al tener nociones de programación en Python, me ha resultado más sencillo de aprender. El software se basa en acciones realizadas mediante actuadores activados según las mediciones de diferentes sensores con diferentes características. La necesidad de buscar información sobre los elementos del sistema en sus datasheets ha afianzado los conocimientos de búsqueda ya adquiridos en la carrera. Los problemas derivados del ruido e interferencias entre componentes, me han servido para entender las diferentes maneras de abordarlos y solucionarlos, tanto en términos de software como en hardware. Incluso el conocer el consumo de los elementos y como se deberían controlar para reducir su consumo al mínimo, me ha permitido aprender sobre un aspecto de la electrónica, que aún entendiendo su importancia, nunca había priorizado.

Finalmente, se ha obtenido un sistema funcional y que cumple los objetivos establecidos al comienzo del proyecto. Completar cada objetivo del sistema uno a uno, superando todos los errores y problemas surgidos durante sus implementaciones, me ha enseñado como todo puede solucionarse con constancia y paciencia. Además, varios objetivos iniciales han sido alterados en el desarrollo en aras de hacerlos más viables y dotarlos de mejor rendimiento/funcionamiento, obteniendo así mayor adaptabilidad y a no caer en el conformismo. También ha permitido conocer en mayor profundidad el sistema Arduino y realizar un trabajo práctico con utilidad para el medio ambiente y el usuario, dueños de plantaciones de pequeña a media escala.

9 Líneas futuras

El trabajo desarrollado en este TFG abarca los campos más prácticos de la electrónica como son la elaboración del sistema eléctrico y la programación de este. El sistema se centra en la gestión del desarrollo agrícola mediante Arduino y sus respectivos sensores. Aún así el desarrollo puede ser extendido por diferentes líneas. Como líneas futuras, se plantea añadir más módulos e incluso explorar otras tecnologías más avanzadas:

- Cámara: Cámara dispuesta con visión completa de la plantación, siendo posible la obtención de imágenes para obtener un seguimiento visual del crecimiento de estas. Capacidad de detectar las posibles enfermedades surgidas por el mal tiempo o por plagas.
- Vial con remedio: Posible vial con remedios para las enfermedades más comunes que sea usado en caso de detectarla para reducir los efectos adversos de ellos.
- Red neuronal para automatización: Implementación de una red neuronal entrenada a base de imágenes de diferentes enfermedades de diferentes plantas y combinarlas con la cámara y el vial para que esté automatizado.

Estas líneas futuras suponen una mejora inmediata (a corto plazo) del sistema que se ha desarrollado, sin embargo, se podrían plantear escenarios más complejos a abordar a más largo plazo, como podría ser la extensión de este proyecto para llevar el control de una plantación de mayor tamaño.

Bibliografía

- [1] *1.0A Rectifier Diode*. 1N4007. Diodes Incorporated. URL: <https://www.diodes.com/assets/Datasheets/ds28002.pdf>.
- [2] *5V, Stepper motor*. 28BYJ-48. Rohs. URL: <https://www.tme.eu/Document/6250c8b5584ec9950520f47b13e256e8/MIKROE-1530-DTE.pdf>.
- [3] *Arduino UNO WiFi Rev.2*. Arduino UNO WiFi Rev.2. Arduino. URL: <https://store.arduino.cc/products/arduino-uno-wifi-rev2>.
- [4] VV Athani. *Stepper motors: fundamentals, applications and design*. New Age International, 1997.
- [5] *Color Light-to-Digital Converter with IR Filter*. TCS34725. AMSCO. URL: <https://pdf1.alldatasheet.com/datasheet-pdf/view/894928/AMSCO/TCS34725.html>.
- [6] *Gravity: Analog Soil Moisture Sensor for Arduino*. SKU:SEN0114. DFRobot. URL: https://wiki.dfrobot.com/Moisture_Sensor__SKU_SEN0114_.
- [7] *Li-polymer Rechargeable Battery*. LP-523450-1S-3. Shenzhen Bak Technology Co.,ltd. URL: <https://www.farnell.com/datasheets/3175990.pdf>.
- [8] *LM35 Precision Centigrade Temperature Sensors*. LM35. Texas Instruments. 2017. URL: <https://www.ti.com/lit/ds/symlink/lm35.pdf>.
- [9] Alfian Ma'arif and Naufal Rahmat Setiawan. "Control of DC Motor Using Integral State Feedback and Comparison with PID: Simulation and Arduino Implementation". In: *Journal of Robotics and Control (JRC)* 2.5 (2021), pp. 456–461. ISSN: 2715-5072.
- [10] *PT550, TO-18 Type Phototransistor with Base Terminal*. PT550. Shenzhen Yongxincheng Technology Co., Ltd. 2008. URL: <https://image.dfrobot.com/image/data/DFR0026/XYC-PT5I850AC-A4xls.pdf>.
- [11] *PT550, TO-18 Type Phototransistor with Base Terminal*. PT550. Sharp Corporation. 2008. URL: <https://media.digikey.com/pdf/Data%20Sheets/Sharp%20PDFs/pt550.pdf>.
- [12] R. Romero et al. "Research on automatic irrigation control: State of the art and recent results". In: *Agricultural Water Management* 114 (2012). For a better use and distribution of water, pp. 59–66. ISSN: 0378-3774.
- [13] *Small Signal NPN Transistor*. PN2222A. DIOTEC [Diotec Semiconductor]. URL: <https://diotec.com/request/datasheet/2n2222a.pdf>.
- [14] *Solar Power Manager v1.0*. SKU:DFR0535. DFRobot. URL: https://wiki.dfrobot.com/Solar_Power_Manager_SKU__DFR0535.
- [15] *Songle Relay, SRD-05VDC-SL-C*. Songle Relay. Ningbo Songle Relay Co.,ltd. URL: <https://pdf1.alldatasheet.com/datasheet-pdf/view/1132028/SONGLERELAY/SRD-05VDC-SL-C.html>.
- [16] *Stand-alone Multiradio modules*. NINA-W10. UBLOX. URL: https://content.arduino.cc/assets/Arduino_NINA-W10_DataSheet_%28UBX-17065507%29.pdf.
- [17] *Subminiature High Power Relay*. HF3FA. Hongfa. URL: https://source.hongfa.com//pdf/web/viewer.html?file=/Uploads/Product/PDF/HF3FA_HF3FA-T_en.pdf×tamp=1687040072.

10 Anexo

El objetivo de este anexo es explicar el montaje del sistema respecto a los pines y alimentaciones, para tener un punto del que partir en caso de desconexión de algún sensor o actuador, además del funcionamiento del código respecto a las diferentes opciones que tiene.

10.1 Instrucciones de montaje

La colocación de los pines es la siguiente:

Tabla 23: Colocación de pines

Sensor/Actuador	Pin
Humedad	A0
Temperatura	A1
Luz ambiental	A2
Motor paso a paso 1	D4
Motor paso a paso 2	D5
Motor paso a paso 3	D6
Motor paso a paso 4	D7
Relé bomba secundaria	D8
Relé bomba principal	D9
Led sensor de color	D10
Sensor de color conexiones I2C	SDA y SCL

La tabla 23 se usa para colocar los pines que están conectados a un pin específico del Arduino.

10.2 Instrucciones código

Para que el código funcione correctamente se necesita de varias librerías para los diferentes sensores o funciones. Las librerías añadidas son las siguientes:

```
#include <AccelStepper.h>
#include <TimeLib.h>
#include <ArduinoMqttClient.h>
#include <WiFiNINA.h>
#include "arduino_secrets.h"
```

Figura 43: Librerías necesarias

Se deben añadir todas estas librerías para que el programa pueda funcionar, para añadirlas se añaden a través de Sketch > Include Library > Manage Libraries y después buscarlas por el mismo nombre de la Figura 43. La última no es una librería sino una tab donde se pone el nombre y contraseña de la red a conectarse para tener una protección más antirobo.

```
//Variables a elegir//  
bool grupoA = false;  
bool grupoB = false;  
bool ConexionWiFi = true;
```

Figura 44: Variables del código

Como opciones a control se daban dos grupos y para ello se deben modificar unas variables en el código, el código a modificar es el siguiente:

Si se desea controlar una plantación del grupo A se debe poner `grupoA = true` y si se quiere control un grupo B se debe poner `grupoB = true`. Para que el sistema no funcione mediante internet para los usuarios que solo deseen controlarlo y no recibir ningún dato se debe poner `ConexionWiFi = false` y `true` en caso contrario.

10.3 Sensor de humedad

En el caso de no utilizar el sensor de humedad de resistencia y usar el de capacidad se recomienda alterar el límite del trigger del sistema de riego debido a que los datos obtenidos de un sensor a otro varían. Se debe variar `humedadlimite = 350` y alterar en el sistema de riego todas las clausulas que contengan `humedad < humedadlimite` por `humedad > humedadlimite`. Con esto el sistema funcionaría para el sensor de humedad basado en capacidad.