



**Universidad**  
Zaragoza

# Trabajo Fin de Grado

## **Huerto Inteligente**

Smart Gardening

Autor:

Joaquín Martínez Belenchón

Director:

Bonifacio Martín del Brío

Escuela de Ingeniería y Arquitectura  
2021



Escuela de  
Ingeniería y Arquitectura  
**Universidad** Zaragoza



## DECLARACIÓN DE AUTORÍA Y ORIGINALIDAD

(Este documento debe entregarse en la Secretaría de la EINA, dentro del plazo de depósito del TFG/TFM para su evaluación).

D./D<sup>a</sup>. \_\_\_\_\_, en  
aplicación de lo dispuesto en el art. 14 (Derechos de autor) del Acuerdo de 11 de  
septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el  
Reglamento de los TFG y TFM de la Universidad de Zaragoza,  
Declaro que el presente Trabajo de Fin de (Grado/Máster)  
(Título del Trabajo)

es de mi autoría y es original, no habiéndose utilizado fuente sin ser  
citada debidamente.

Zaragoza,

Fdo:

# RESUMEN

Este trabajo consiste en un estudio acerca de la automatización de un huerto familiar situado en una zona rural de Teruel, que se encuentra alejado de la población y, por ello, sin acceso a suministro eléctrico ni a internet. Para dar solución a problemas como los relacionados con el riego y la gran distancia a la vivienda habitual, planteamos diferentes soluciones tecnológicas para implantar en el huerto con el fin de lograr su gestión automatizada y posibilitar monitorización y control a distancia.

Para ello, se hará uso de dos placas Arduino para realizar el control del riego de las dos parcelas que constituyen el huerto, controlando dos bombas y una electroválvula contando tanto con la información aportada por dos sensores de nivel a instalar en los depósitos de agua del huerto, como de sensores de temperatura, humedad relativa y humedad del suelo. Se realizarán los programas para controlar los tiempos de riego, los cuales podrán modificarse a través de un módulo GSM que permite establecer una comunicación a distancia con el sistema electrónico que controla el huerto. Además, se incorporan los dispositivos necesarios para poder realizar su manejo directo cuando se está físicamente en el huerto, haciendo uso de encoders rotativos junto con una pantalla LCD.

Por último, se alimentará toda la instalación mediante el uso de placas solares junto con una batería, para dotar de cierta autonomía tanto por la noche como los días no soleados.

# **Contenido**

Capítulo 1. Introducción .....	8
1.1. Contexto .....	8
1.2. Objetivos .....	10
1.3. Cronograma .....	13
Capítulo 2. Análisis de posibles soluciones.....	15
2.1. Diagrama de bloques.....	15
2.2. Sensor de nivel .....	18
2.3. Sensor de humedad del suelo .....	21
2.4. Sensor de temperatura y humedad .....	22
2.5. Electroválvula .....	24
2.6. Bomba hidráulica .....	27
2.6.1. Bomba de llenado de bidones.....	28
2.6.2. Bomba para riego del circuito inferior .....	29
2.7. Módulo para transmitir potencia a la electroválvula y a las bombas .....	31
2.8. Control y visualización .....	34
2.8.1. Control del circuito superior.....	35
2.8.2. Control del circuito inferior.....	37
2.9. Elección de los microcontroladores.....	38
2.9.1. Microcontrolador para el circuito superior.....	38
2.9.2. Microcontrolador para el circuito inferior.....	40
Capítulo 3. Diseño del hardware .....	40
Capítulo 4. Comunicaciones .....	42
4.1. Introducción al GSM/GPRS.....	43
4.2. Microcontrolador con comunicación GSM/GPRS.....	43
Capítulo 5. Alimentación mediante placas solares .....	44
Capítulo 6. Desarrollo del software .....	47
6.1 Software del circuito superior .....	47
6.1.1. Interrupción del encoder rotativo .....	48
6.1.2. Control de la electroválvula.....	49
6.1.3. Llenado de los bidones .....	50
6.2 Software del circuito inferior .....	51
6.2.1. Modos de funcionamiento .....	51

6.2.2. Modo automático.....	52
6.2.3. Modo manual.....	53
6.2.4. Control del riego mediante la bomba .....	53
6.2.5. Control mediante pantalla LCD y encoder rotativo .....	55
6.2.6. Control mediante página WEB.....	57
<b>Capítulo 7. Prototipo.....</b>	<b>60</b>
7.1. Interrupción Sensor de Nivel .....	61
7.2. Interrupción de reloj.....	63
7.3. Programa completo .....	67
7.4. Pruebas del prototipo.....	67
7.5. Implantación del prototipo .....	68
<b>Capítulo 8. Conclusiones .....</b>	<b>71</b>
<b>Capítulo 8. Bibliografía .....</b>	<b>73</b>

## Índice de Figuras

Figura 1. Vista del depósito y de los bidones .....	8
Figura 2. Vista del huerto tomada con un dron .....	9
Figura 3. División del huerto en dos circuitos de riego.....	10
Figura 4. Cronograma de organización del proyecto .....	14
Figura 5. Diagrama de bloques circuito superior .....	16
Figura 6. Diagrama de bloques circuito inferior.....	17
Figura 7. Tipos de sensores de nivel.....	18
Figura 8. Sensor de nivel magnético .....	20
Figura 9. Conexionado sensor de nivel .....	20
Figura 10. Sensor de humedad del suelo capacitivo.....	22
Figura 11. Conexionado sensor de humedad del suelo capacitivo .....	22
Figura 12, Sensor de temperatura y humedad DHT22 .....	23
Figura 13. Conexionado sensor de temperatura y humedad DHT22 .....	24
Figura 14. Válvula 2/2 NC accionada por electroimán y retorno por muelle .....	25
Figura 15. Válvula 2/2 NO accionada por electroimán y retorno por muelle .....	25
Figura 16. Válvula 2/2 de latch .....	26
Figura 17. Electroválvula VLN Baja Presión 9V 1" Rain Bird.....	27
Figura 18. Bomba sumergible para llenado de bidones.....	29
Figura 19. Bomba Seaflo serie 41 .....	30
Figura 20. Características técnicas de la bomba Seaflo serie 41 .....	30
Figura 21. Temperatura de funcionamiento bomba.....	30
Figura 22. Conexionado del control de una carga con un relé .....	31
Figura 23. Conexionado del control de una carga con un transistor MOSFET.....	33
Figura 24. Conexionado del control de una carga con un MOSFET de nivel lógico.....	34
Figura 25. Funcionamiento del encoder rotativo.....	35
Figura 26. Visualización de la salida en cada uno de los canales del encoder.....	35
Figura 27. Encoder rotativo KY-040 .....	36
Figura 28. Conexionado KY-040 .....	37
Figura 29. Conexionado LCD 20x4 con módulo KY-040 .....	38
Figura 30. Arduino Nano Every .....	39
Figura 31. Obtención de los diferentes voltajes .....	41
Figura 32. Diseño completo del circuito inferior .....	41
Figura 33. Diseño completo del circuito superior .....	42

Figura 34. Arduino MKR GSM 1400.....	44
Figura 35. Instalación fotovoltaica .....	45
Figura 36. Pack placa solar, regulador y batería.....	47
Figura 37. Diagrama de flujo encoder rotativo.....	48
Figura 38. Diagrama de flujo electroválvula .....	49
Figura 39. Diagrama de flujo llenado de los bidones .....	50
Figura 40. Diagrama de flujo de los modos de funcionamiento.....	52
Figura 41, Funcionamiento modo automático .....	53
Figura 42. Diagrama de flujo del funcionamiento de la bomba .....	54
Figura 43. Diagrama de flujo del funcionamiento de la pantalla LCD con el selector ..	55
Figura 44. Pantallas de los cuatro estados de la LCD .....	56
Figura 45. Pantalla principal página WEB .....	57
Figura 46. Pantalla para modificar el programa de riego .....	57
Figura 47. Diagrama de flujo modificación de programa de riego .....	58
Figura 48. Pantalla del historial de los sensores.....	59
Figura 49. Diagramas de flujo historial de sensores.....	59
Figura 50. Modos de funcionamiento de la Interrupción .....	61
Figura 51. Rebotes en la conmutación del sensor de nivel.....	62
Figura 52. Filtrado de los rebotes mediante condensador .....	62
Figura 53. Mensajes mostrados al cambiar el estado de los sensores de nivel.....	63
Figura 54. Valor del contador de ticks en cada temporización .....	64
Figura 55. Registro TCCR1B .....	65
Figura 56. Fórmula para calcular el registro de comparación del Timer .....	65
Figura 57. Registro TIMSK1 .....	65
Figura 58. Interrupción cada 4 segundos.....	66
Figura 59. Temporizaciones repetitivas de 12 segundos.....	67
Figura 60. Simulación del funcionamiento sin agua .....	68
Figura 61. Simulación del funcionamiento con el nivel alto.....	68
Figura 62. Conexionado del sensor del depósito.....	69
Figura 63. Acabado final del sensor del depósito.....	69
Figura 64. Montaje del sensor en uno de los bidones.....	70
Figura 65. Acabo final del sensor del bidón.....	70
Figura 66. Conexión de ambos sensores con Arduino .....	71
Figura 67. Acabado final del prototipo.....	71

## **Índice de Tablas**

Tabla 1. Equivalencias tareas cronograma .....	13
Tabla 2. Comparación entre los sensores DHT11 y DHT22.....	23
Tabla 3. Equivalencias de las posiciones de los encoders.....	37
Tabla 4. Comparación placas Arduino .....	39
Tabla 5. Consumos del circuito superior .....	46
Tabla 6. Consumos del circuito inferior .....	46

## **Anexos**

1. Cálculo de la instalación de riego para la electroválvula.....	1
2. Cálculos de la instalación de riego para seleccionar la bomba.....	4
3. Selección de los transistores MOSFET y cálculo de la potencia disipada.....	5
4. Código del prototipo.....	7
4.1. Interrupción Sensor de Nivel.....	7
4.2. Interrupción de reloj.....	9
4.3. Programa completo.....	10
5. Presupuesto.....	11
6. Datasheets.....	12

# Capítulo 1. Introducción

## 1.1. Contexto

Desde siempre mi familia se ha dedicado a cultivar las tierras en un pequeño pueblo de la provincia de Teruel, Guadalaviar, antes como modo de vida y ahora como *hobby* y para autoconsumo. Actualmente, por motivos laborales residimos en Teruel capital y tan solo vamos al pueblo los fines de semana y los periodos de vacaciones. Es por ello que únicamente cultivamos un pequeño huerto para el consumo familiar de frutas, verduras y hortalizas.

El huerto se encuentra alejado del núcleo urbano, por lo que está aislado de cualquier tipo de suministro eléctrico y, obviamente, de conexión a internet. Además, se encuentra al final de los últimos cultivos del pueblo siguiendo el curso del río, por lo que en verano suele faltar agua para el riego, ya que los vecinos que se encuentran antes la agotan. Por ello, aprovechando un pequeño manantial existente en la parte alta, que se ve incrementado cuando riegan los huertos situados en la parte superior, diseñamos un sistema de riego muy eficiente que permite poder regar durante toda la temporada de cultivo.



*Figura 1. Vista del depósito y de los bidones*

El agua de este manantial es almacenada en un depósito de 13.000 litros (Figura 1) y, posteriormente, distribuida mediante el sistema de goteo, mucho más eficiente y

productivo que el riego a manta. La parcela donde se encuentra el huerto es de  $500\text{ m}^2$  (Figura 2) y, por la diferencia de altura, está dividida en dos partes de cultivo, separadas estas por una acequia que en verano se encuentra casi siempre seca.



*Figura 2. Vista del huerto tomada con un dron*

A lo largo del trabajo, por el mero hecho de diferenciarlas, denominaremos a la parte donde se encuentra el depósito “parcela superior” y a la restante “parcela inferior”.

El depósito se haya a la misma altura, incluso, un poco más bajo que la parte cultivable de la parcela superior, lo que imposibilita su riego directo por gravedad. Por ello, hemos instalado dos bidones de 1000 litros cada uno (Figura 1), conectados por abajo (vasos comunicantes) y que se encuentran a una altura de 2 metros y medio respecto del suelo y cuyo llenado se realiza actualmente mediante una bomba de gasolina desde el depósito.

En el caso del riego de la parcela inferior, existe una diferencia de altura de 50 centímetros, pero, debido a la gran distancia desde el depósito hasta el invernadero (30 metros), la presión que llega es insuficiente para que el riego se realice correctamente, provocando así que el caudal de los últimos goteros del circuito de riego sea prácticamente nulo.

Con los motivos expuestos, queda evidente la separación del proyecto en dos circuitos de riego y, por tanto, en dos sistemas de control totalmente independientes (Figura 3).



*Figura 3. División del huerto en dos circuitos de riego*

Desde que recurrimos al riego por goteo hasta el día de hoy, utilizamos programadores a pilas para controlar el tiempo de riego que abren o cierran el circuito de riego mediante el giro de una bola. Debido a la baja presión de la que disponemos y a la cal existente en los tubos, cada temporada nos vemos obligados a cambiar los programadores por unos nuevos, ya que los que tenemos empiezan a fallar a la hora de conmutar y no cierran totalmente el paso en estado cerrado, de tal forma que se producen pérdidas de agua, encharcando la zona de cultivo provocando que ciertos alimentos se pudran.

## **1.2. Objetivos**

En definitiva, los objetivos principales del Trabajo son los siguientes:

- Obtener una solución funcional para el problema del riego en el huerto con el fin de dejar de utilizar programadores ineficientes.
- Automatizar determinadas tareas debido a la imposibilidad de realizar un seguimiento diario físicamente en el huerto.

### **Solución para el circuito de riego de la parcela inferior**

Respecto al primer problema, la única solución reside en incrementar la presión de riego. Para ello es necesario una bomba que sea capaz de suministrar el mismo caudal a todos los goteros por igual, evitando que los últimos del circuito se queden sin agua.

Cuando empezamos a investigar bombas para solventar el problema, nos encontramos con que, al tener un gran circuito de riego con muchos goteros y una gran distancia desde el depósito hasta estos, la bomba que necesitábamos debería ser muy potente, impidiéndonos alimentarla con pilas o baterías pequeñas.

Al encontrarnos aislados de suministro eléctrico, en este punto surge otro de los grandes objetivos del trabajo: la obtención de electricidad mediante una instalación aislada de paneles solares.

### **Automatización de diferentes tareas**

El segundo tema a tratar está relacionado con diferentes tareas cuya automatización mejoraría la situación del huerto. Actualmente, en la temporada de cultivo, cada domingo cuando volvemos a Teruel seleccionamos la frecuencia y la duración de riego que queremos utilizar durante la semana.

Este sistema no se adecúa a las condiciones óptimas de riego, principalmente porque no tiene en cuenta los cambios meteorológicos, pero nos soluciona parcialmente los problemas de abastecimiento de agua entre semana. Entonces, la solución planteada a este problema pasa por establecer comunicación a distancia con el huerto, de tal forma que seamos capaces de modificar variables del riego, así como conocer determinadas magnitudes físicas del terreno mediante sensores. El motivo de esta decisión se basa en dos situaciones reales con las que contamos actualmente:

- **Necesidad de realizar cambios en el riego.** El cuidado del huerto es complejo y no consiste únicamente en las tareas de plantación y riego, conlleva muchos más trabajos. Para argumentar tal afirmación vamos a utilizar uno de los muchos ejemplos que nos exigen cambiar el riego entre semana:

Dos o tres veces cada temporada tenemos que “cavar los surcos de patatas” (quitar a mano las hierbas malas que salen cerca de las raíces de las patatas para que no consuman agua del riego y así conseguir que las patatas crezcan más).

Esta tarea supone menor esfuerzo físico cuando la tierra está seca. Podemos ejemplificarlo de manera que, si se quiere, por ejemplo, hacer un fin de semana, lo ideal sería dejar de regar el miércoles para el sábado poder hacerlo cómodamente y una vez hecho retomar el riego.

- **Variación de las condiciones meteorológicas:** Por mucho que cada domingo miremos el tiempo que va a hacer a lo largo de la semana, este puede cambiar. Por ello, si una semana llueve más de lo esperado, se sobrecarga el cultivo de agua, corriendo el riesgo de que ciertos alimentos se pudran. Por el contrario, si una semana es más calurosa, el riego fijado puede no ser suficiente provocando que se sequen ciertos cultivos.

Estos dos acontecimientos nos han ocasionado problemas en la recolección en los años anteriores, por ello su corrección como una tarea importante.

Por último, otra funcionalidad que hemos decidido automatizar es la derivada del riego del circuito superior. En la actualidad se ejecuta desde los bidones con un programador, llenando estos desde el depósito con una bomba de gasolina únicamente los fines de semana.

Por tanto, la idea es realizar el llenado con una bomba eléctrica que podamos manejar con un controlador y utilizar sensores de nivel que permitan que la bomba funcione de forma automática en función del nivel del depósito y los bidones.

Una vez dispongamos de agua en los bidones, realizaremos el control de los tiempos de riego mediante una electroválvula, ya que, al estar más altos que el circuito de riego, en este caso no tenemos problemas de presión debido a la diferencia de altura.

Para realizar el control del proyecto, hemos decidido utilizar tarjetas Arduino por las facilidades a la hora de conectar los diferentes componentes, por la sencillez de su uso y por la gran documentación de la que dispone.

Con todo, podemos destacar que la iniciativa para realizar el trabajo surge de los problemas reales comentados en este apartado. Por ello, el objetivo principal no es únicamente realizar un estudio, sino implantar progresivamente las soluciones propuestas en función de la disponibilidad de tiempo y económica (se trata de un desarrollo ambicioso que excede con mucho los créditos asignados a un trabajo fin de grado).

### 1.3. Cronograma

En la tabla que se presenta a continuación podemos ver las equivalencias entre las tareas a realizar y una referencia que se ha asignado a cada una de ellas, con el fin de ahorrar espacio en el diagrama de Gantt y apreciar mejor los periodos.

Equivalencias	Tareas realizadas
T1	Primeras reuniones para delimitar los contenidos del proyecto
T2	Estudio de los diagramas de bloques necesarios
T3	Desglose de cada bloque del circuito
T4	Diseño completo del hardware
T5	Investigación de las posibles comunicaciones
T6	Estudio de la comunicación mediante Arduino MKR 1400 GSM
T7	Análisis de la alimentación eléctrica mediante paneles solares
T8	Toma de decisiones acerca del funcionamiento deseado
T9	Realización del software de control
T10	Construcción de prototipo demostrativo
T11	Cálculo de coste total del proyecto
T12	Obtención de conclusiones
T13	Realización de la memoria final del TFG
T14	Últimas correcciones con el tutor antes de depositar

*Tabla 1. Equivalencias tareas cronograma*

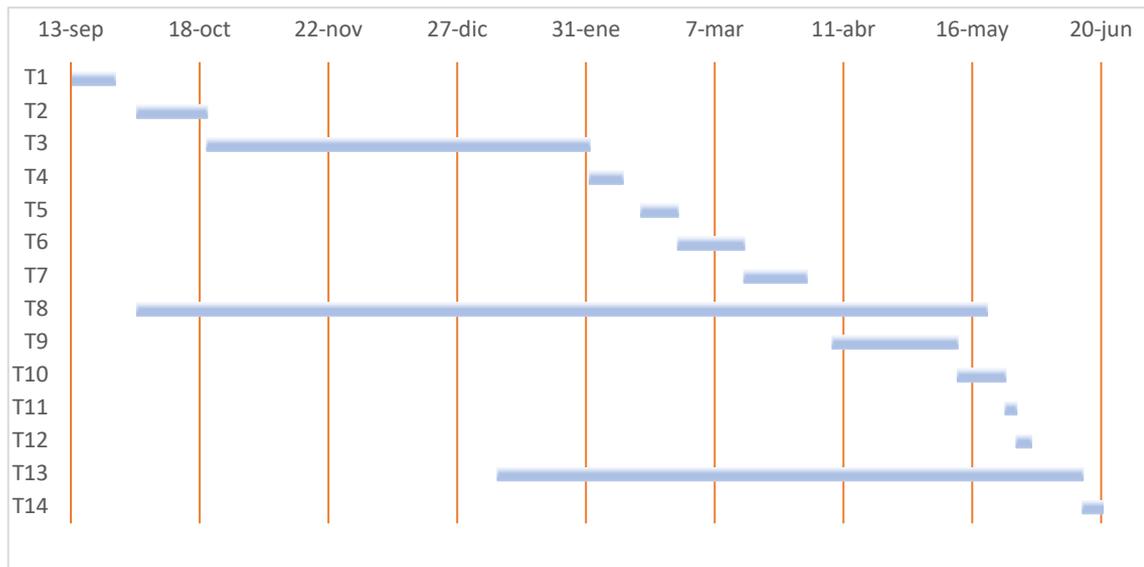


Figura 4. Cronograma de organización del proyecto

Como podemos observar en la Tabla 1, hemos separado todas las tareas en varios bloques, que pasamos a describir.

El primero de ellos, comprende las reuniones iniciales y el diseño completo del hardware, el cual, como apreciamos en la Figura 4 ha sido una de las tareas más costosas. Este bloque se trata en los Capítulos 2 y 3:

Los siguientes dos bloques tratan las comunicaciones y la alimentación de la instalación mediante placas solares y se tratan en los Capítulos 4 y 5 respectivamente.

A continuación, tenemos el bloque de software, que, como podemos ver en el cronograma, tiene una duración similar a la total del proyecto. Esto es debido a que comprende todas las tomas de decisiones del funcionamiento deseado de la instalación, las cuales se han ido modificando y perfeccionando durante todo el proyecto. Este bloque se trata en el Capítulo 6.

La siguiente tarea es la 10, que hace referencia a la construcción de un prototipo demostrativo del proyecto y se trata en el Capítulo 7.

Finalmente, tenemos el cálculo de presupuesto, la obtención de conclusiones del proyecto y la realización y corrección de errores de la memoria, la cual se ha realizado de forma bilateral al resto de tareas. El presupuesto se localiza en el documento Anexos (Capítulo 5) junto con otros cálculos realizados en el proyecto, y las conclusiones corresponden al Capítulo 7.

Como podemos apreciar en el diagrama, se trata de un proyecto muy amplio. Esto ha provocado que, aunque las previsiones eran tener determinadas partes implantadas en el huerto a finales de abril para el inicio de la temporada de cultivo, la magnitud del trabajo y la falta de tiempo lo han impedido.

Únicamente hemos podido construir el prototipo de una de las funcionalidades estudiadas como una pequeña demostración del proyecto.

A pesar de esto, se espera que, con fecha máxima de abril de 2022, todos los componentes estudiados en el proyecto estén instalados y funcionando en el huerto de forma que no tengamos que preocuparnos más por los problemas existentes actualmente.

## **Capítulo 2. Análisis de posibles soluciones**

### **2.1. Diagrama de bloques**

Como hemos comentado en la introducción, vamos a dividir el proyecto en dos circuitos independientes con el objetivo de realizar un mejor control.

#### I) Circuito Superior

Lo primero es plantear las necesidades de este circuito. Queremos realizar el control del riego de esa parcela con el agua proveniente de los bidones que están situados a una altura de 2 metros y medio por encima. Para ello hemos pensado en utilizar una electroválvula pilotada eléctricamente que mediante una señal de Arduino podamos modificar su estado de abierta a cerrada controlando así los tiempos de riego.

El siguiente problema por tratar sería cómo llenar los bidones desde el depósito de 13.000 litros, que lo solventaremos con una bomba de agua. Para saber cuándo tenemos que activar la bomba necesitamos saber el nivel que tienen tanto el depósito como los bidones, por ello necesitaremos dos sensores de nivel.

Por último, es fundamental un sistema de control y visualización que nos permita modificar los parámetros que controlan el riego.

Esta división del terreno dispone de una tierra de peor calidad y por tanto menos productiva, por lo que la reservamos para la cultivación de especies que requieren menos

cuidados, como por ejemplo una regulación menos estricta de la cantidad de agua o una frecuencia de riego menor. En este terreno solemos cultivar cebollas, puerros, coles, patatas o diferentes frutales como manzanos, perales, ciruelos...

Debido a este motivo, hemos decidido que en esta parte de cultivo vamos a establecer un riego regulable sólo mediante dos seleccionadores de frecuencia y duración, ya que no requiere mayor cuidado en función de los diferentes factores meteorológicos.

Por tanto, el diagrama del bloque para este circuito de acuerdo con nuestras necesidades puede verse en la Figura 5.

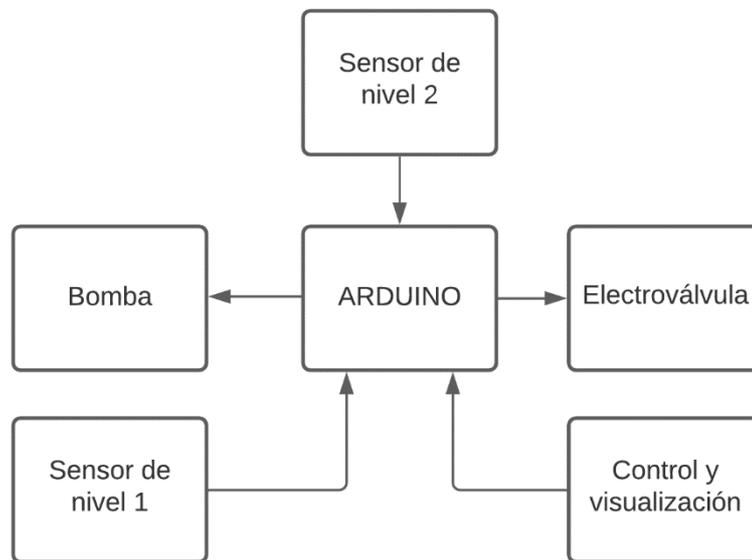


Figura 5. Diagrama de bloques circuito superior

## II) Circuito Inferior

En este caso, como el circuito de riego está 50 cm más bajo que el depósito, la presión no es suficiente para que el riego por goteo se realice de forma homogénea tanto al principio como al final de los circuitos. Por ello introducimos una bomba que incremente el caudal y la presión homogenizando el riego en todos los circuitos.

Este circuito va a controlar el riego de dentro y fuera del invernadero, donde plantamos principalmente lechugas, tomates, pimientos, calabacines, borraja, etc. Por tanto, para esta parte sí es necesaria una atención especial.

Por ello, introducimos dos sensores de humedad del suelo y de temperatura y humedad relativa que nos informarán acerca de las condiciones meteorológicas y el correspondiente estado de la tierra, ayudándonos a tomar decisiones relativas al riego.

También añadimos un módulo GSM (*Global System for Mobile communications*) que posibilita las comunicaciones necesarias para poder controlar el riego a distancia, uno de los principales problemas a resolver.

Finalmente, al igual que en el caso anterior, utilizamos un bloque de control y visualización, pero que en este caso no está formado por dos selectores, sino que está compuesto por un selector y una pantalla LCD (*Liquid Cristal Display*).

Esta decisión es debida a que en el primer circuito sólo se puede modificar el riego mediante los selectores, permitiéndonos éstos ver en qué estado nos encontramos. En cambio, en el segundo circuito, como el programa de riego se puede modificar a distancia, necesitamos un método de visualización que nos muestre la descripción del programa en el que nos encontramos.

De acuerdo con lo comentado, el diagrama de bloques del segundo circuito podemos verlo en la Figura 6.

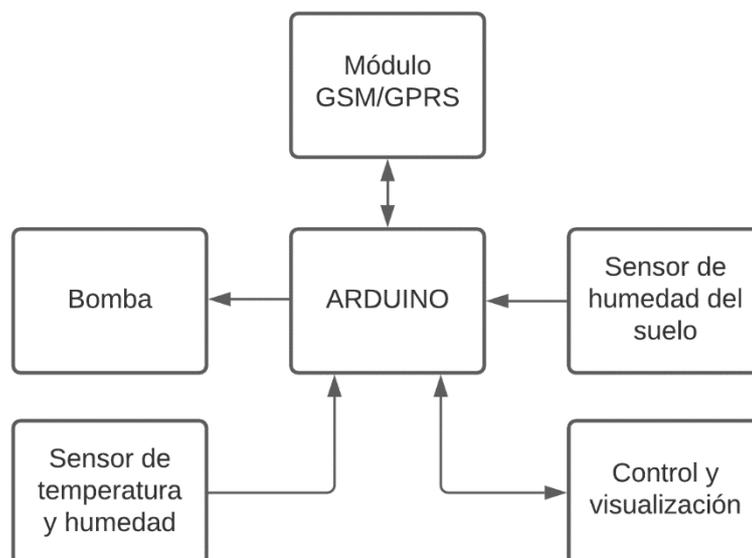


Figura 6. Diagrama de bloques circuito inferior

## 2.2. Sensor de nivel

Un sensor de nivel es un dispositivo que mide el nivel de líquido de un tanque o un depósito. El nivel se puede medir de dos maneras [1]:

- **Mediciones continuas:** Muestran el nivel continuo del líquido, así como cuando sube o baja el nivel. Estos sensores nos dan una salida analógica que nos indica dentro del rango de medición qué nivel tiene el líquido dentro del tanque o recipiente.
- **Medidas del punto de nivel:** Indican cuando el líquido está en un cierto punto. Su principal función es actuar de *switch* (interruptor), ya que conmutan cuando el líquido llega a un cierto nivel fijado anteriormente.

En [1] y [2] se tratan los diferentes tipos de sensores de nivel y su división en función de los dos sistemas explicados anteriormente (Figura 7).

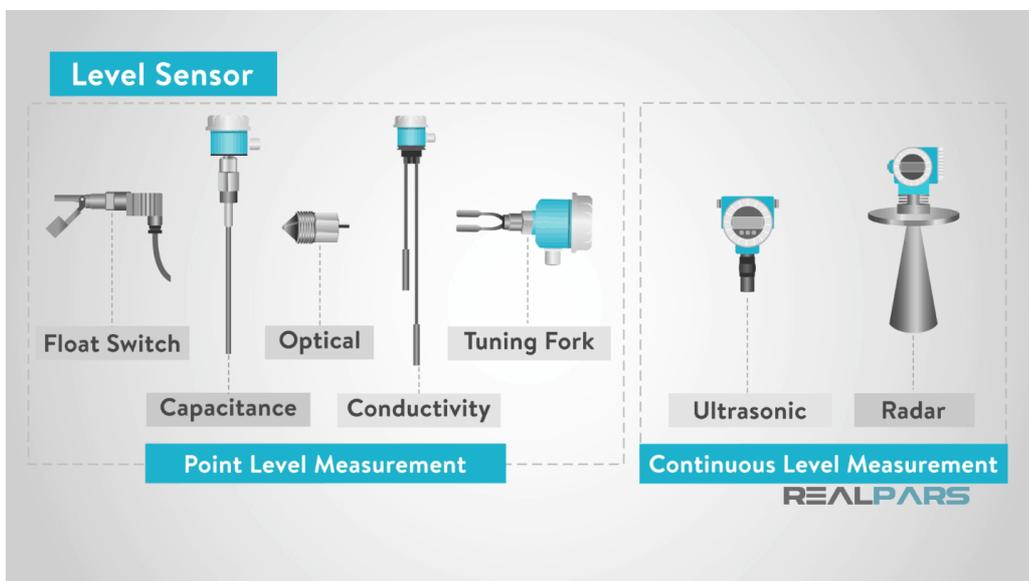


Figura 7. Tipos de sensores de nivel

- **Float Swicth (Interruptor por flotador):** Es un flotador magnético que se encuentra en un cierto nivel. Cuando el líquido está por debajo del sensor, éste se encuentra en una posición (cerrado o abierto dependiendo de la colocación del sensor) y cuando llega a dicho nivel el líquido mueve una pieza provocando su conmutación al estado opuesto y por tanto la detección del nivel.

- **Capacitance (Capacitivo):** Se basan en el principio de que los líquidos tienen diferentes constantes dieléctricas que el aire. Por ello miden el cambio de capacitancia entre dos placas producidas cuando el líquido entra en contacto con el sensor.

- **Optical (Óptico):** Su funcionamiento se basa en los cambios de transmisión en una luz infrarroja emitida por un LED (*Light-emitting diode*). El sensor está formado por un LED y un fototransistor. Cuando el sensor está en contacto con el aire, toda la emisión de luz del diodo se refleja debido a la forma del prisma y vuelve al fototransistor. Cuando el nivel del líquido supera el sensor la luz deja de reflejarse completamente y se refracta al líquido con la consiguiente pérdida de luz en el fototransistor y la obvia detección del nivel del líquido.

- **Conductivity (Por conducción):** Se usa para líquidos que sean conductivos (como el agua) o que sean altamente corrosivos. Su funcionamiento consiste en dos probetas metálicas, una larga y una corta. En la larga se aplica un pequeño voltaje y cuando el nivel del líquido llega hasta la probeta corta, se cierra el circuito y se detecta entonces que el líquido ha llegado al nivel que se estaba estudiando.

- **Tuning fork (Por vibración):** El sensor está formado por dos palas que vibran a su natural frecuencia de resonancia. Cuando el nivel del líquido sube hasta el sensor, su frecuencia cambia y conocemos el nivel.

- **Ultrasonic (Ultrasónico):** Este sensor emite una onda ultrasónica que rebota en el líquido y vuelve al receptor del sensor. Midiendo el tiempo que tarda esta onda desde que sale hasta que es recibida, el sensor es capaz de calcular la distancia del sensor al nivel del líquido y así el nivel absoluto del depósito.

- **Radar (Ondas microondas):** El funcionamiento es similar al del sensor ultrasónico, pero con la diferencia de que la frecuencia ahora es de ondas microondas en vez de ultrasónicas.

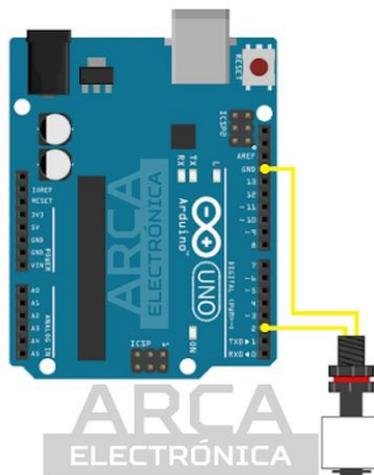
Una vez explicados los diferentes sensores disponibles tenemos que elegir uno para nuestro proyecto. Primero nos decantamos por un sensor de punto de nivel ya que es más fácil de evaluar y al tratar grandes cantidades de agua no es necesario conocer el nivel de

ambos depósitos en todo momento. Dentro de este grupo hemos decidido utilizar el sensor de flotador magnético (Figura 8) debido a que es el sensor más básico y barato que satisface nuestras necesidades. El precio de este sensor ronda el valor de 1€ en páginas como Aliexpress.



*Figura 8. Sensor de nivel magnético*

Respecto a su conexionado con Arduino, este sensor tiene dos cables que conectaremos a un pin con capacidad de interrupción (Figura 9) y a masa, ya que su funcionamiento idóneo es que se genere una interrupción cada vez que el nivel del depósito supere el nivel en el cual se ha colocado el sensor.



*Figura 9. Conexionado sensor de nivel*

## 2.3. Sensor de humedad del suelo

Un sensor de humedad del suelo es un instrumento que nos aporta información acerca del suelo que se estudia. De acuerdo con los artículos [3] y [4], en función del tipo de sensor podemos medir dos variables diferentes:

- **Tensión matricial**

Miden la cantidad de esfuerzo que tiene que realizar la raíz para absorber el agua. Tienen como unidad de medida el bar y cuanto menor sea el esfuerzo necesario, mayor cantidad de humedad tendrá el suelo. Existen dos tipos, tensiómetros y sensores de matriz granular.

- **VWC (*Volumetric Water Content* - Contenido del agua en el suelo)**

Miden la proporción de volumen de agua que hay respecto al volumen total del estudio y lo muestra en porcentaje. Existen dos tipos de sensores que miden el contenido del agua en el suelo, los resistivos y los capacitivos

El método que utilizan los sensores resistivos [5] consiste en colocar dos electrodos separados por una distancia, aplicar un voltaje y medir la intensidad que atraviesa por los electrodos para ver la resistencia que ofrece el suelo y por tanto su humedad. Como inconveniente podemos destacar que pueden oxidarse con el tiempo.

Los capacitivos [5] funcionan de forma similar, pero en este caso miden la capacitancia, la cual depende del área, del dieléctrico (material entre las placas conductoras, en nuestro caso el suelo) y de la distancia. Como el área y la distancia van a ser constantes, la capacitancia solo dependerá de los cambios de la constante dieléctrica y podremos conocer la humedad.

Para nuestro proyecto nos interesa medir el contenido del agua en el suelo, ya que nos ofrece una información más fácil de evaluar. Y dentro de los dos tipos de sensores que miden esta variable hemos elegido el sensor capacitivo, ya que es mucho más exacto que el sensor resistivo, no presenta problemas de oxidación y desde el punto de vista económico la diferencia no es notoria, ya que su valor es de 1.5€.



Figura 10. Sensor de humedad del suelo capacitivo

Para comunicarse con Arduino, este sensor tiene tres pines, alimentación, masa y una salida analógica que es la que ofrece el valor de la medición, por lo que su conexión se muestra en la Figura 11.

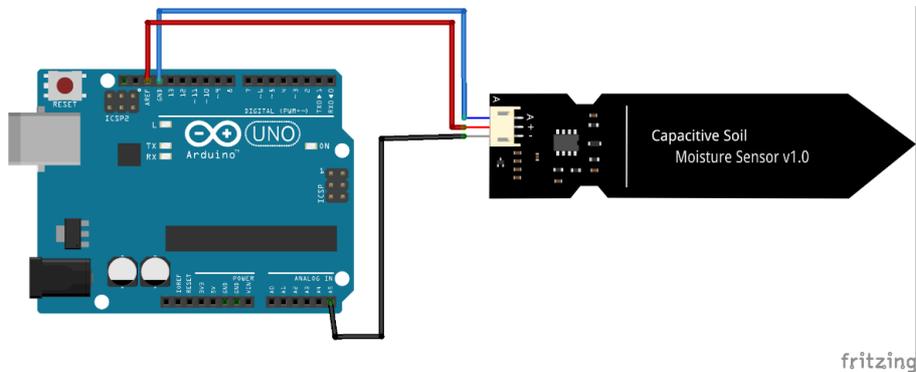


Figura 11. Conexionado sensor de humedad del suelo capacitivo

Para evitar que el microcontrolador esté constantemente comprobando las salidas del sensor, vamos a tratarlo mediante interrupciones de reloj, de forma que cada cierto tiempo que determinaremos más adelante, el microcontrolador realizará la medida del sensor. Así conseguimos reducir el consumo de forma importante.

## 2.4. Sensor de temperatura y humedad

Un sensor de temperatura permite medir la temperatura e informarnos de esta a través de una señal eléctrica. Por otro lado, un sensor de humedad nos informa acerca de la cantidad de vapor de agua que existe en el aire, expresándose esta como un porcentaje.

Existen diferentes sensores tanto de temperatura como de humedad en función de la técnica utilizada para realizar la medición. Pero hay dos sensores muy utilizados (por su sencillez y su precio) en proyectos similares con Arduino que nos permiten medir ambas propiedades a la vez, por lo que decidimos centrarnos en ellos.

Estos sensores son el DHT11 y DHT22 y están formados [6] por un termistor NTC (*Negative Temperature Coefficient*) para medir la temperatura, un sensor de humedad y un microcontrolador que nos permite obtener los valores analógicos de estas variables en formato digital. Tienen un funcionamiento muy similar pero el DHT22 tiene mayor precisión a costa de tener un periodo de medición de 2 segundos en vez de 1 segundo que tiene el DHT11. Esto no presenta un problema para nuestro proyecto, ya que el este sensor se va a tratar por interrupción de reloj, de tal forma que el microcontrolador solicitará información al sensor en un periodo mucho mayor que 2 segundos.

En la siguiente tabla podemos observar alguno de los parámetros de medición de los dos sensores.

Modelo	DHT11	DHT22
Rango de medición de humedad	20-90 % HR	0-100 % HR
Rango de medición de temperatura	0 hasta 50 °C	-40 hasta 80 °C
Precisión de temperatura	±2 °C	±0.5 °C
Precisión de humedad	±5 % HR	±2 % HR

Tabla 2. Comparación entre los sensores DHT11 y DHT22

Como vemos nos permite medir un rango más amplio de humedad y temperaturas y con una precisión mucho mayor y debido a que su precio es levemente superior (1€ en páginas como aliexpress), hemos decidido utilizar el sensor DHT22.

En la Figura 12 podemos ver la correspondencia de cada pin y en la Figura 13 su conexionado con el microcontrolador, usando una resistencia de *pull-up* de 10kΩ como recomienda el fabricante.

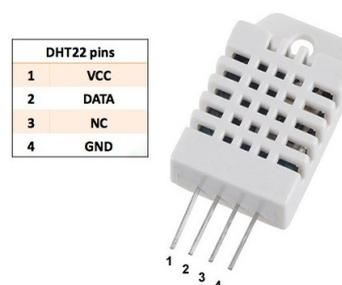


Figura 12. Sensor de temperatura y humedad DHT22

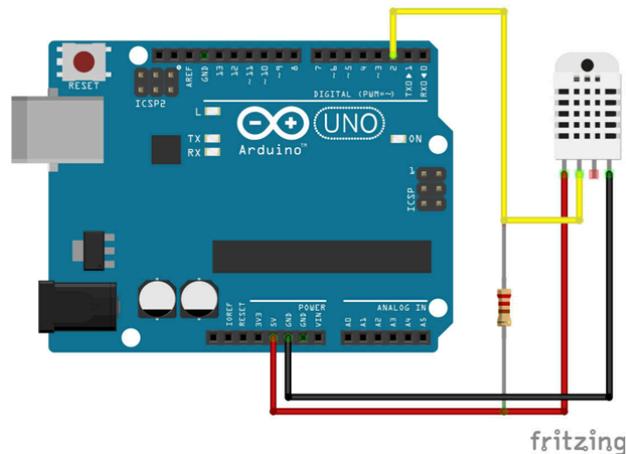


Figura 13. Conexión sensor de temperatura y humedad DHT22

## 2.5. Electroválvula

Una electroválvula es un dispositivo electromecánico que se encarga de controlar el flujo que circula por un conducto, abriendo o cortando el paso de este. Podemos encontrar electroválvulas motorizadas (precio muy elevado), de globo (prácticamente en desuso) y las de membrana. En estas últimas, el control de la apertura del paso se realiza mediante la presión que ejerce el propio fluido sobre una membrana flexible. Las electroválvulas de membrana también se denominan electroválvulas de solenoide y nos vamos a centrar en ellas para nuestro propósito.

Las electroválvulas de solenoide las podemos clasificar de acuerdo a diferentes criterios [7] [8]:

### Según su funcionamiento

Según su funcionamiento podemos encontrar dos tipos de electroválvulas:

- De tipo directo: No requieren una diferencia de presión para operar ya que la bobina del solenoide actúa directamente sobre la membrana.
- Asistidas: Estas válvulas no están controladas directamente por el solenoide, si no que cuando recibe corriente eléctrica cierra o abre un piloto y el fluido es el encargado de cerrar o abrir con su propia presión. Utilizan la diferencia de presión del fluido para abrirse y cerrarse. Se suele utilizar para aplicaciones de gran caudal.

Por tanto, debido a que no disponemos de gran diferencia de presión entre la entrada y la salida, vamos a utilizar las electroválvulas de tipo directo.

### Según su forma

En función de su forma, las electroválvulas pueden tener 2,3, 4 o 5 vías. En nuestro caso como se va a utilizar únicamente de llave de paso puesto que sólo tenemos un circuito de riego, es obvio que vamos a utilizar la válvula de 2 vías, la cual permite o impide el paso del agua por la tubería.

### Según su alimentación

Respecto a su alimentación tenemos dos tipos:

- Corriente alterna: Se controlan con 24-230V de alterna. Pueden ser normalmente abierta (NO – *Normally Open*) o normalmente cerradas (NC – *Normally Closed*) (Figura 13 y 14), es decir, tienen una posición fija y cuando se le suministra corriente cambia de posición el tiempo que permanezca la señal eléctrica y una vez se corta la señal de pilotaje, el muelle, que estaba presionado, retorna a la válvula a su posición fija. Tienen consumos elevados debido al suministro continuo de electricidad para su funcionamiento.

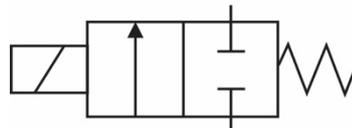


Figura 14. Válvula 2/2 NC accionada por electroimán y retorno por muelle



Figura 15. Válvula 2/2 NO accionada por electroimán y retorno por muelle

- Corriente continua: Funcionan con corriente continua y pueden ser de dos tipos: de *latch* (impulso) o continua. Las de continua tienen un funcionamiento similar a las de corriente alterna, pudiendo ser estas normalmente abiertas o cerradas.

Por el contrario, las de *latch* [9] son electroválvulas biestables (Figura 16), es decir, sólo necesitan un impulso para cambiar de posición. Estas válvulas están

continuamente desenergizadas y cuando reciben un impulso cambia de estado, se abren y el solenoide se engancha en esta posición gracias a un imán que lo retiene de forma permanente. Para cerrar la válvula es necesario otro impulso que es el encargado de crear una fuerza contraria que se encarga de neutralizar la fuerza del imán y permite que el actuador vuelva a su posición normal.

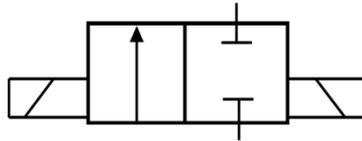


Figura 16. Válvula 2/2 de latch

De acuerdo con nuestra situación de alimentación del sistema mediante placas solares, resulta evidente que las electroválvulas con corriente continua de tipo *latch* son las más adecuadas ya que tienen muy bajo consumo respecto a las otras opciones.

### **Selección de la electroválvula más adecuada en el mercado**

Una vez decidida la electroválvula deseada, hay que elegir una dentro de la gran variedad que ofrece el mercado [10]. Para ello hemos de tener en cuenta varios parámetros, como el caudal, la presión y el diámetro.

Para el cálculo del caudal y la presión recurrimos a las fórmulas de instalaciones de fluidos en conductos estudiadas en la asignatura de Mecánica de Fluidos [11]. Analizando los resultados obtenidos en el documento Anexos (Capítulo 1), tenemos un caudal de 37.4 l/min y una presión en la salida de 0.23 bar y observamos que la mayoría de las electroválvulas de solenoide de 9V tipo *latch* requieren una presión mínima de 0.7bar, por lo que no podemos utilizarlas para nuestro proyecto.

Existen unas electroválvulas de baja presión que nos ofrecen la posibilidad de trabajar en el rango de 0.05 – 1 bar, por lo que escogemos este modelo para nuestro diseño. Una opción sería la “Electroválvula VLN Baja Presión 9V 1" Rain Bird” la cual podemos adquirirla por 55€ (Figura 17).



*Figura 17. Electrovalvula VLN Baja Presión 9V 1" Rain Bird*

Respecto al conexionado, la electrovalvula, al igual que las bombas, consumen mucha mayor intensidad de la que es capaz de suministrar Arduino, por lo que requerirán una etapa de potencia intermedia que se estudiará posteriormente en un apartado exclusivo.

## 2.6. Bomba hidráulica

Una bomba hidráulica es una máquina que transforma la energía suministrada para mover un fluido. Podemos clasificarlas de acuerdo con diferentes criterios [12][13]:

Según el **tipo de accionamiento** de la bomba, podemos dividirlos en manuales, neumáticas, de accionamiento hidráulico, motobombas y electrobombas. Para nuestro proyecto las bombas tienen que ser de accionamiento eléctrico (electrobombas) ya que son las únicas que nos permitirán hacer un control de ellas mediante señales eléctricas.

Dentro de las bombas eléctricas, estas pueden estar alimentadas con corriente alterna o continua. Las bombas de corriente continua consumen mucha menos energía que las de alterna, pero por contrapartida cabe destacar que las bombas de continua requieren mayor mantenimiento y tienen más desgaste. Con el objetivo de minimizar los consumos hemos decidido que vamos a usar bombas de corriente continua.

Otro criterio de clasificación es su **funcionamiento**, en función del cual tenemos dos tipos [14]:

- **Bombas volumétricas:** El principio de funcionamiento de esas bombas consiste en disminuir el volumen de la cámara de almacenamiento del fluido, consiguiendo así que se incremente la presión.

- Bombas rotodinámicas: Su funcionamiento consiste en la transmisión de energía al fluido mediante rotación, es decir mediante el giro de unos alabes incrementamos la presión del fluido.

Las bombas volumétricas se utilizan sobre todo para fluidos con una gran densidad, mientras que las rotodinámicas son las más adecuadas para bombear fluidos, por lo que nos decantamos lógicamente por las bombas rotodinámicas. Las bombas rotodinámicas podemos diferenciarlas según su **uso** [15]:

- Bomba sumergible: Se sumergen en el líquido totalmente y permiten evacuar agua de un pozo de gran profundidad.
- Bomba de superficie: Se colocan en la superficie y extraen el agua mediante una manguera de aspiración.
- Bomba presurizadora: Se colocan a la salida del depósito y se usan para incrementar la presión y a su vez el caudal.

Por último, podríamos decir que existe otra clasificación en función de las partículas sólidas diluidas en el líquido que distingue entre agua limpia y sucia. En nuestro caso estaríamos usando agua sucia, pero utilizando un filtro eliminamos las partículas sólidas, por lo que podríamos usar ambos tipos de bombas.

Como hemos comentado en el diagrama de bloques, necesitamos una bomba para cada uno de los circuitos, las cuales tienen diferentes funcionalidades. Por ello, una vez explicados todos los tipos, vamos a estudiar por separado cada uno de los casos.

### **2.6.1. Bomba de llenado de bidones**

Para el circuito superior, la funcionalidad deseada es la de bombear agua desde el depósito a los dos bidones con una diferencia de altura de apenas 4 metros. Los bidones poseen una capacidad de 1000 litros cada uno, pero están unidos con un tubo por la parte de abajo, de tal manera que funcionan como si fuesen un único bidón de 2000 litros, lo que nos permite usar un único sensor de nivel para detectar el nivel total.

Partiendo de una bomba eléctrica alimentada con corriente continua y rotodinámica hay que decidir en función de su uso de entre las tres explicadas. Para este caso queda obviamente excluida la tercera opción y nos centramos en las bombas de superficie o

sumergidas. De acuerdo con [16], nos decantamos por una bomba sumergible, ya que requiere menos mantenimiento, ofrece mayor presión y en menor tiempo.

Una vez seleccionada el tipo de bomba, necesitamos saber qué características requiere. Por ello comenzamos a buscar una bomba que proporcione esos requisitos, minimizando el consumo y el precio. Seleccionamos la siguiente bomba (Figura 18).



*Figura 18. Bomba sumergible para llenado de bidones*

Esta bomba se alimenta a 12 Voltios y nos ofrece un caudal de 1320 litros/hora en vacío y 1000 con un desnivel de 3,5m con un consumo de 50W por un precio de 39€. Además, nos dice que su funcionamiento idóneo es el de funcionar 30 minutos y dejarla descansar 30 minutos, requisito que deberemos tener en cuenta en la parte de software.

### **2.6.2. Bomba para riego del circuito inferior**

La segunda bomba se usa para incrementar el caudal y la presión para el sistema de riego de la parcela inferior. Por ello resulta evidente que vamos a utilizar una bomba eléctrica, rotodinámica y presurizada. Sin embargo, para elegir esta bomba es más importante el caudal y la presión que en el caso anterior, ya que una presión elevada podría dañar los elementos del circuito de riego. Por tanto, vamos a realizar un estudio del caudal y la presión necesaria para que el riego funcione correctamente y así poder elegir la bomba.

De acuerdo con el análisis realizado en el documento Anexos (Capítulo 2), necesitamos una bomba que nos proporcione como mínimo una presión de 0,36 bares con un caudal de 780 litros por hora (13litros por minuto). Con los datos de nuestro estudio la bomba que más se acerca a nuestras condiciones es la bomba Seaflo serie 41 que consume cerca de 130W y tiene un precio de 90€ (Figura 19).



Figura 19. Bomba SeaFlo serie 41

Con la Figura 20 analizamos su hoja de características y vemos que con una presión un poco superior a la que necesitamos ofrece un caudal de algo más de 13 litros por minuto con un consumo de 9.93 Amperios, lo que encaja en nuestro caso.

PRESSURE		FLOW		CURRENT
PSI	BAR	GPM	LPM	APMS
0	0	4.33	16.40	5.9
20	1.38	3.49	13.20	9.93
25	1.73	3.43	13.00	10.48
30	2.07	2.59	9.80	10.99
35	2.42	2.06	7.80	11.49
40	2.76	0	0	12.13

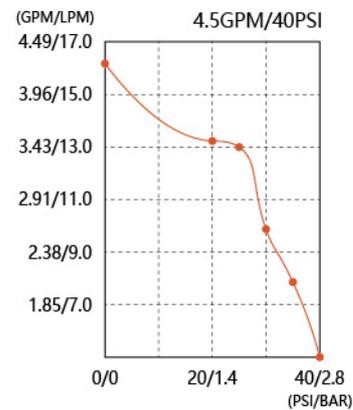


Figura 20. Características técnicas de la bomba SeaFlo serie 41

Por último, una gráfica interesante de su hoja de características la apreciamos en la Figura 21, que nos informa de la temperatura que adquiere la bomba mientras trabaja.

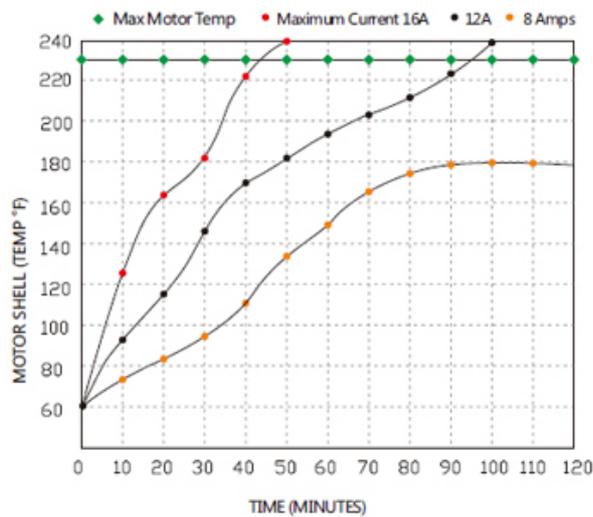


Figura 21. Temperatura de funcionamiento bomba

En esta gráfica vemos que trabajando a una intensidad de 12 Amperios (intensidad superior a la real con la que trabajaremos) la bomba llega a su temperatura máxima en 95 minutos, dato que tendremos en cuenta en la parte de control.

## 2.7. Módulo para transmitir potencia a la electroválvula y a las bombas

Arduino trabaja con 5V de tensión máxima y con una intensidad de salida por cada uno de sus pines de unos 20mA de media y de 40mA de máxima y de 200mA por la de alimentación (Vcc).

Tanto la electroválvula como las bombas trabajan a una tensión de 9 y 12 Voltios y con unas intensidades de unos pocos Amperios. Por tanto, resulta obvia la necesidad de utilizar un componente capaz de realizar el control de esas cargas a través de Arduino. Las soluciones estudiadas para solventar este problema son tres:

- Relé

Un relé es un interruptor mecánico que podemos accionar eléctricamente [17]. La intensidad necesaria para hacer conmutar el relé es de 500mA, mucho más elevada de la que es capaz de suministrar Arduino, por ello se usa un transistor BJT (transistor de unión bipolar) para conseguir la intensidad necesaria (Figura 22).

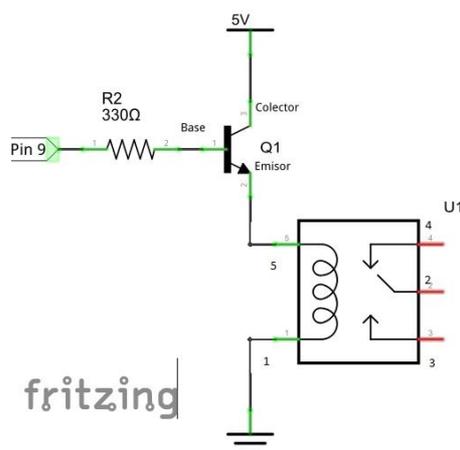


Figura 22. Conexión del control de una carga con un relé

Como este circuito es muy utilizado para realizar el control de grandes cargas, existen módulos (*drivers*) que incorporan el circuito expuesto y por tanto simplemente hace falta conectarlos al pin del microcontrolador y a la carga que queremos controlar.

Los relés tienen la ventaja de que pueden trabajar con cargas de voltajes elevados, incluso con corriente alterna, pero al ser dispositivos mecánicos tienen un cierto retardo (0.2-2s) y desgaste asociado a cada conmutación y su vida útil normalmente es del orden de 100.000 conmutaciones.

Además, tienen un elevado consumo de intensidad en estado ON provocando una gran disipación de potencia, por lo que los descartamos debido a nuestra intención de minimizar lo máximo posible el consumo.

- Relé de estado sólido

Un relé de estado sólido o SSR (*Solid State Relay*) [18] es un dispositivo basado en semiconductores que tiene un funcionamiento parecido al relé ya explicado. No tiene partes móviles ya que está formado únicamente por componentes electrónicos y por tanto su velocidad de conmutación es del orden de milisegundos. No sufre problemas de desgaste y permite trabajar con cargas de 220V en alterna y con voltajes de continua, pero si trabajamos una intensidad elevada la potencia se disipa en calor y necesitaríamos incluir un disipador de calor.

Al igual que el relé convencional necesita una etapa anterior para poder conseguir la intensidad necesaria para conmutar. Pero su gran inconveniente es su precio, por lo que descartamos su uso en este proyecto.

- Transistor

Un transistor es un dispositivo electrónico que modifica una señal eléctrica como respuesta a una señal de entrada. Podemos diferenciar dos tipos de transistores: bipolar (BJT) y los de efecto campo (FET) [19].

A diferencia del relé, el transistor es un elemento electrónico, por tanto, no tiene ningún tipo de desgaste por lo que puede conmutar infinitas veces. Por otro lado, su conmutación es prácticamente instantánea pero solo nos permite trabajar con 30 Voltios de continua, lo que no presenta un problema en nuestro proyecto.

Los transistores BJT tienen una gran caída de tensión entre colector y emisor y por tanto una gran disipación de potencia si la intensidad del colector es elevada. Pero la problemática principal de los transistores bipolares es que la intensidad máxima de colector no es muy elevada y por tanto limita las cargas que se puedan conectar. De ahí surge la necesidad de usar los transistores MOSFET (transistor de efecto campo metal-óxido-semiconductor), que son una subfamilia de los transistores FET. Estos transistores operan de forma similar a los BJT, pero con alguna diferencia como, por ejemplo: en modo saturación se comportan como una resistencia muy pequeña, lo que provoca pérdidas muy pequeñas de potencia al conectar grandes cargas. Además, por lo general, suelen ser más rápidos y son menos sensibles a las temperaturas.

El funcionamiento deseado del transistor es el de funcionar como interruptor, que deje pasar o no la corriente en función de la salida del microcontrolador. Para ello el transistor conmutará entre los estados de corte y saturación. Sin embargo, MOSFET tienen un inconveniente al usarlo con Arduino [20] ya que para usarlos en modo saturación necesitan una tensión de puerta (*gate*) de 10V y Arduino únicamente puede alimentar a 5 Voltios. Para solucionar este problema tenemos dos posibles soluciones:

- Usar una etapa intermedia con un transistor BJT (igual que planteábamos en el caso del relé) que aumente los niveles de salida del microcontrolador, obteniendo así los 10V y poder usar un MOSFET normal.

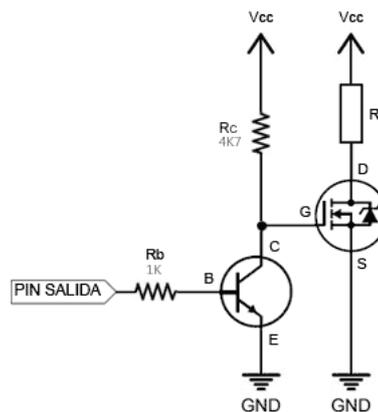


Figura 23. Conexión del control de una carga con un transistor MOSFET

- Usar un MOSFET de potencia para nivel lógico, los cuales tienen una baja tensión de puerta y por tanto pueden ser utilizados directamente por un microcontrolador.

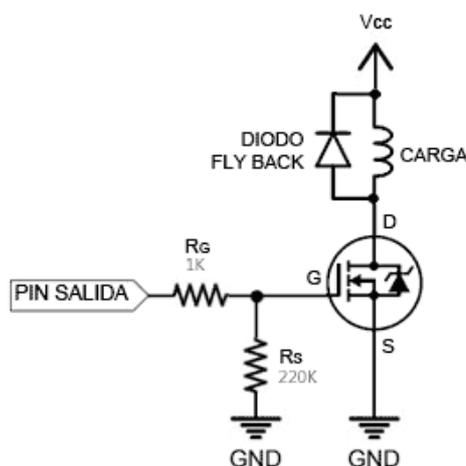


Figura 24. Conexión del control de una carga con un MOSFET de nivel lógico

Por tanto, decidimos que un transistor MOSFET es la mejor solución para el problema, y nos decantamos por utilizar de nivel lógico para ahorrarnos una etapa.

Respecto a las resistencias de la Figura 24,  $R_G$  de  $1k\Omega$  se utiliza para limitar la corriente de salida de Arduino y para asegurar que el transistor conmute correctamente de saturación a corte. Por otro lado, la resistencia  $R_S$  con un valor de  $220k\Omega$  se utiliza para conectar el pin de salida del microcontrolador con GND (*ground*, tierra) para evitar que en caso de que por el pin de salida tuviésemos un estado de alta impedancia se activase el MOSFET.

Por último, el diodo en paralelo con la carga se utiliza cuando esta es inductiva para evitar corriente inversa en el transistor.

Entonces, una vez elegido el tipo de componente que vamos a utilizar para solventar el problema de potencia de Arduino, tenemos que elegir qué modelo usar para cada uno de los circuitos en función de las intensidades necesarias. De acuerdo con el estudio realizado en el documento Anexos (Capítulo 3), vamos a utilizar tres transistores MOSFET FDP8874 iguales debido a su baja disipación de potencia que adquiriremos por Mouser por un precio total de 5€.

## 2.8. Control y visualización

Como hemos comentado, para ambos circuitos necesitamos los bloques de control y visualización, aunque para cada uno de los circuitos va a ser diferente.

### 2.8.1. Control del circuito superior

En el caso del primer circuito, no es necesario un gran control de los parámetros del riego. Por ello hemos considerado que la mejor solución es utilizar dos encoders rotativos que nos permitan seleccionar la frecuencia y la duración con la que queremos regar. Este es un sistema que usan muchos programadores de riego, dado a su bajo precio y alta eficiencia. Estos encoders a su vez funcionan como visualizadores, ya que nos permiten conocer en cada momento, los parámetros con los que estamos trabajando.

Un encoder rotativo [21][22] es un elemento electromecánico que permite codificar el movimiento mecánico y conocer así su posición angular. Están formados por un disco con diferentes posiciones y dos patillas metálicas que cuando se colocan en una posición cierran el contacto (Figura 25). Tienen dos *switches*, uno formado entre el pin A y el pin C, y el otro en el B y el C.

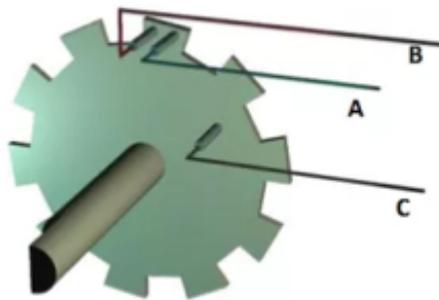


Figura 25. Funcionamiento del encoder rotativo

En cada posición los dos conectores están cerrados o abiertos, de forma que un movimiento en alguna dirección cambiará el estado de los dos, así que tenemos que estudiar cuál ha conmutado antes para conocer el sentido de giro.

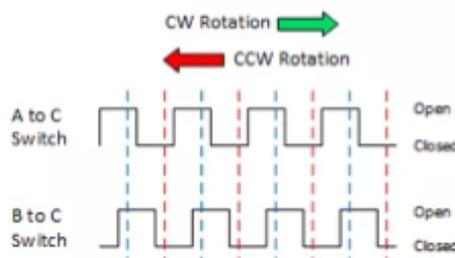


Figura 26. Visualización de la salida en cada uno de los canales del encoder

Como vemos en la Figura 26, si conmuta primero el A sabremos que estamos girando en sentido horario, y si conmuta primero B en antihorario. De esta forma partiendo de la

posición inicial a 0°, y conociendo cada vez que gira y en que dirección, podemos saber su posición absoluta y por tanto el parámetro al que representa ese ángulo de giro.

El encoder lo vamos a tratar por interrupción, para que el microcontrolador no tenga que estar comprobando constantemente. Tenemos dos opciones, conectar sólo un pin a interrupción, de forma que cuando salte una interrupción tendremos que comprobar el estado del otro canal para conocer el sentido de giro o conectar los dos de forma que en función de en qué pin salte la interrupción conoceremos el sentido de giro, lo que añade mayor precisión. Esta decisión la tomaremos en la elección del microcontrolador en función de los pines que posibilitan utilizar interrupciones y las necesidades del sistema.

En el mercado estos encoders suelen estar disponibles en módulos adaptados para usarlos directamente por microcontroladores por un precio de 1€. Así, una buena opción para nuestro propósito es el encoder KY-040 (Figura 27), que dispone de 12 posiciones y tiene 5 pines: + (alimentación) y GND (*Ground*), CLK (*Clock*) y DT (*Data*) que hacen referencia a los dos canales ya explicados y SW (*Switch*) que permite su uso como pulsador en caso de conectarlo, pero que para este primer caso vamos a prescindir de su uso.



Figura 27. Encoder rotativo KY-040

El conexionado del encoder KY-040 lo podemos ver en la Figura 28, aunque como ya hemos comentado para este caso el pulsador no lo vamos a utilizar.

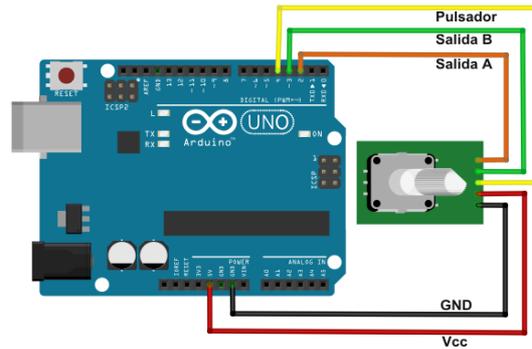


Figura 28. Conexionado KY-040

Una vez explicado el funcionamiento del encoder rotativo hemos asignado un valor a cada posición de los dos encoders (frecuencia y duración) para establecer el programa de riego en función de los valores introducidos:

Posición	0	1	2	3	4	5	6	7	8	9	10	11
Frecuencia(h)	1	2	3	4	5	6	8	12	24	48	72	Semanal
Duración(min)	OFF	1	3	5	10	15	20	30	40	50	60	ON

Tabla 3. Equivalencias de las posiciones de los encoders

De acuerdo con la Tabla 3 se podrá elegir cualquier combinación posible, siendo las más restrictivas la posición OFF y ON del selector de duración, permitiendo abrir o cerrar de forma permanente hasta el próximo cambio.

### 2.8.2. Control del circuito inferior

Para el segundo circuito, el programa de riego se va a modificar por otros componentes aparte del bloque de selección manual, por ello no podemos utilizar el sistema de control anterior ya que no nos ofrece la visualización necesaria para conocer en qué programa nos encontramos en caso de que este haya sido modificado posterior al cambio manual.

Por ello la decisión que hemos tomado es introducir un encoder rotativo y una pantalla LCD. De esta forma, en la pantalla se mostrará el programa en el que nos encontramos y con el encoder podremos movernos por un menú programado.

Para la selección de la pantalla hemos decidido usar una LCD 20x4 ya que disponemos de cuatro filas en vez de dos de la más conocida 20x2, porque consideramos que para usarlo como seleccionador de opciones ofrece más facilidades.

En este caso sí vamos a usar el *switch* conectándolo a un pin con capacidad de interrupción, de tal forma que la pantalla permanecerá en modo de bajo consumo hasta que se pulse el switch, momento en el que se encenderá y podremos navegar por el menú programado.

La pantalla LCD se conectará a Arduino a través de un módulo que posibilita la comunicación I2C (*Inter-Integrated Circuit*) que es una comunicación en serie con el microcontrolador que nos permite utilizar únicamente dos pines para alimentación y masa y otros dos pines analógicos para comunicación, en vez de los ocho que utilizaríamos si la usásemos en paralelo.

En el mercado existen bloques formados por la pantalla LCD 20x4 con el módulo integrado que permite conectarla mediante el protocolo I2C por un precio de 6€.

Por último, el conexionado de este bloque lo podemos ver en la Figura 29.

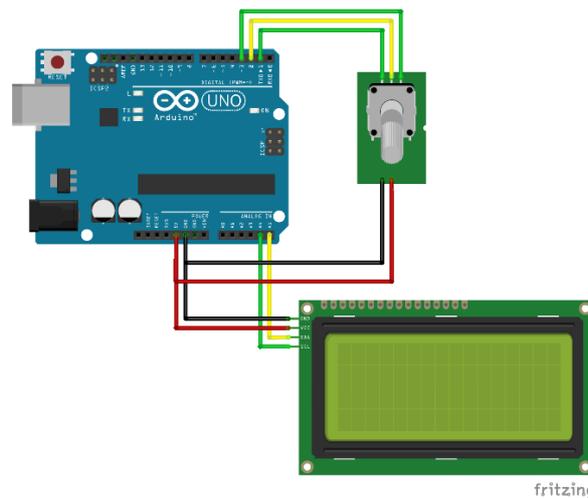


Figura 29. Conexionado LCD 20x4 con módulo KY-040

## 2.9. Elección de los microcontroladores

Al tener dos circuitos independientes, resulta obvia la necesidad de utilizar dos microcontroladores. Por tanto, de la misma manera que los anteriores apartados, vamos a estudiar cada caso de forma independiente.

### 2.9.1. Microcontrolador para el circuito superior

Primero de todo analizamos que componentes vamos a conectar al microcontrolador. De acuerdo con el estudio realizado en los apartados anteriores de cada uno de los bloques

del diagrama de bloques planteado al inicio del Capítulo 2 tenemos: dos sensores de nivel, una bomba, una electroválvula y dos encoders rotativos.

La bomba y la electroválvula necesitan cada una de ellas un pin digital simple, sin embargo los sensores de nivel los vamos a tratar por interrupción, al igual que los encoders, con la diferencia de que estos podemos conectarlos con precisión simple (una interrupción para los dos canales) o doble (dos interrupciones, una para cada canal).

De esta forma vamos a necesitar una placa Arduino que tenga como mínimo 4 pines con interrupción si usamos precisión simple o 6 pines si usamos precisión, ya que en ningún caso vamos a utilizar el *switch* de los encoders.

Por tanto, realizamos un estudio de las placas Arduino que cumplan nuestros requisitos (sin excederse con funcionalidades innecesarias para nuestro propósito como, por ejemplo: Wifi, Bluetooth, etc.) y obtenemos que las posibilidades son las siguiente:

Placa	Pines digitales con interrupción	Número de pines digitales	Precio
Nano Every	Todos los pines digitales	14	11€
Micro, Leonardo	0, 1, 2, 3, 7	20	18€

Tabla 4. Comparación placas Arduino

Analizando los datos de la Tabla 4, hemos decidido utilizar la placa Nano Every ya que las otras placas estudiadas tienen únicamente 5 pines con interrupción impidiéndonos usar la precisión doble con los encoders rotativos y además por un precio mayor.

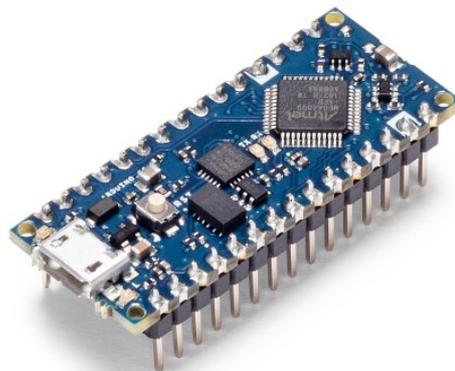


Figura 30. Arduino Nano Every

Arduino Nano Every (Figura 30) es el microcontrolador de la marca más pequeño, siendo la evolución del Nano, pero con un procesador más potente, ATmega4809. Como

dato importante para nuestro proyecto podemos destacar que su tensión de funcionamiento es de 5 Voltios, aunque permite alimentarlo con un voltaje de entre 7 y 21 Voltios.

### **2.9.2. Microcontrolador para el circuito inferior**

Respecto al circuito inferior, tal y cómo hemos planteado el diagrama de bloques, tenemos un bloque para la comunicación GSM y otro para el microcontrolador. Sin embargo, como estudiaremos en el Capítulo 4 dedicado a las comunicaciones, la empresa Arduino nos ofrece un microcontrolador que además incluye las herramientas necesarias para comunicarnos mediante GSM ahorrándonos así el uso de una *shield* (placa complementaria al microcontrolador para añadir determinadas funcionalidades).

## **Capítulo 3. Diseño del hardware**

Una vez estudiado cada bloque por separado de ambos circuitos realizamos el diseño completo del hardware con las conexiones de cada uno de los componentes. Primero, analizamos los voltajes que necesitamos para cada componente ya que ahora mismo únicamente disponemos de 12 Voltios que obtenemos de las placas solares. Tanto la bomba del circuito inferior como del superior trabajan a 12 Voltios, por lo que se pueden conectar directamente. Sin embargo, la electroválvula del circuito superior requiere de un voltaje de entrada de 9 Voltios.

Por otro lado, la placa Arduino Nano Every necesita una tensión de entrada de entre 7 y 21 Voltios, por lo que podríamos alimentarla a 12 Voltios, pero la placa Arduino MKR 1400 GSM utilizada en el circuito inferior, únicamente puede alimentarse con 5 Voltios y tiene una tensión de funcionamiento de 3.3 Voltios.

Por tanto, necesitamos una tensión de 12, 9 y 5 Voltios. Como la diferencia de voltaje es pequeña la solución más eficiente es utilizar reguladores lineales para obtener los valores fijos de tensión. Para el primer caso vamos a utilizar el regulador lineal 7809 y para el segundo 7805 ambos con sus respectivos condensadores como recomienda el fabricante (Figura 31). Además de lo comentado, para evitar dañar componentes del circuito, hemos decidido utilizar un magnetotérmico de 12 Voltios de corriente continua de 20 Amperios que colocaremos a la salida del regulador.

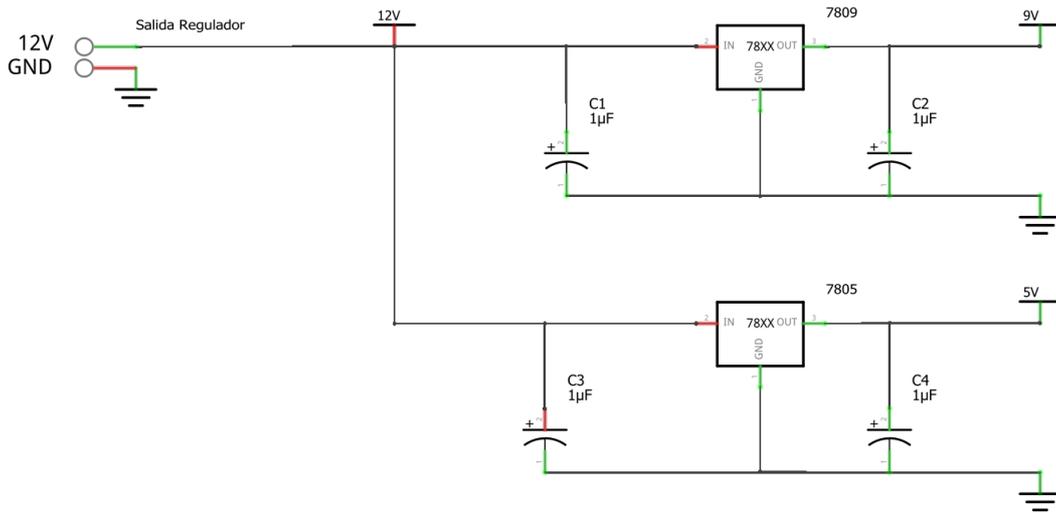


Figura 31. Obtención de los diferentes voltajes

Como he comentado, el microcontrolador del circuito inferior tiene una tensión de funcionamiento de 3.3V. Esto inicialmente parece un problema a la hora de conectar los sensores, pero analizando la hoja de características de cada componente, comprobamos que todos pueden trabajar a 3.3V. Esto supone una ventaja ya de esta manera reducimos la potencia necesaria. De esta forma, el circuito general (Figura 32) queda formado por el sensor de humedad del suelo, el sensor de humedad relativa y temperatura, el encoder rotativo junto con la pantalla LCD 20x4 conectada a través del protocolo I2C y la bomba de agua activada gracias a un MOSFET de nivel lógico.

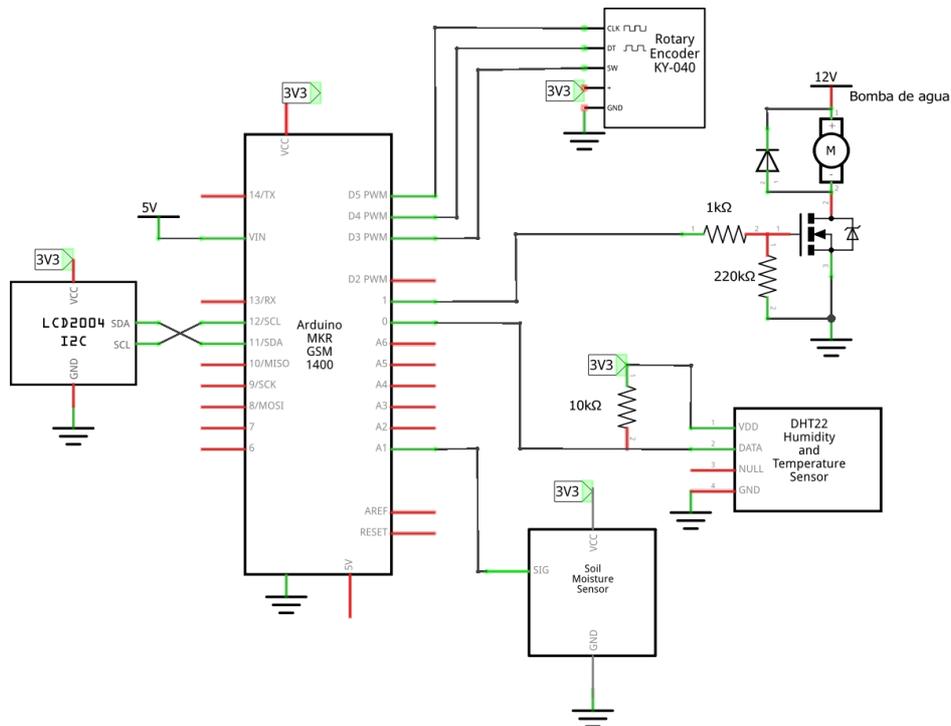


Figura 32. Diseño completo del circuito inferior

Respecto al circuito superior, aunque Arduino Nano Every nos ofrece la posibilidad de trabajar a 5 Voltios, con los encoders rotativos también vamos a trabajar a 3.3 Voltios para disminuir el consumo.

Por lo demás, el circuito detallado (Figura 33) queda formado por los dos encoder rotativos para la selección de frecuencia y duración, los dos sensores de nivel (uno en el depósito y otro en el bidón), la bomba de llenado de los bidones y la electroválvula para el riego que ambas trabajan con la misma etapa de potencia del circuito anterior.

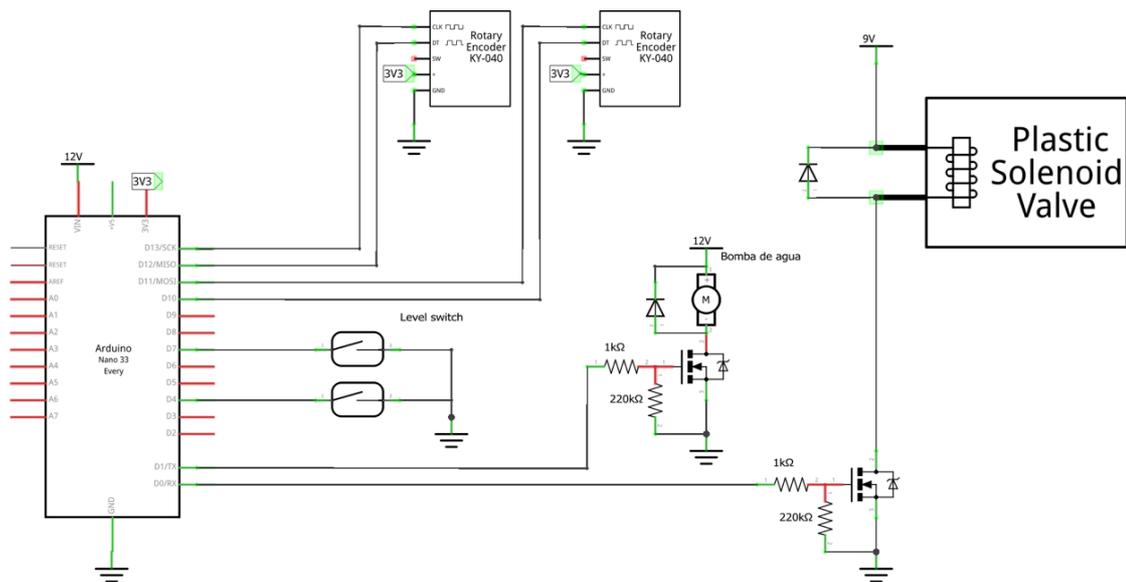


Figura 33. Diseño completo del circuito superior

## Capítulo 4. Comunicaciones

Este apartado está dedicado a establecer el tipo de comunicación que permitirá conectarnos con el microcontrolador situado en el huerto sin la necesidad de estar físicamente allí. El objetivo principal es poder modificar el programa de riego a distancia, aunque además añadiremos más funcionalidades como poder conocer las temperaturas ambientes o de la tierra cada cierto tiempo, realizar un historial de las temperaturas y de la humedad o ver los valores máximo y mínimos alcanzados por cada sensor en un determinado periodo.

Cuando empezamos a plantear esta idea, lo primero que pensamos fue en comunicarnos con WIFI, pero como el huerto no se sitúa cerca del núcleo urbano esta opción quedó totalmente descartada. A continuación, buscamos diferentes alternativas y

de todas las que encontramos, vimos que una destacaba por encima de las demás en proyectos similares y además se adecuaba perfectamente a nuestras necesidades.

## 4.1. Introducción al GSM/GPRS

La comunicación de la que hablamos se denomina GSM (Sistema global para las comunicaciones móviles) y se define como “un estándar de comunicación para la telefonía móvil, implementado mediante la combinación de satélites y antenas terrestres” [23][24].

Su función principal fue establecer comunicación telefónica entre distintos usuarios, pero también permitía la transmisión de datos. Los móviles mediante una tarjeta SIM (*Subscriber Identity Module*) actúan como estación móvil y envían datos a una estación fija que es controlada por las diferentes operadoras para establecer las comunicaciones. A esta tecnología se le denominó 2G, pero poco tiempo después se desarrolló el sistema GPRS (*General Packet Radio Service*) conocido como el 2.5G.

Este sistema de comunicación móvil [25] utilizaba las redes del GSM, pero con la novedad de enviar paquetes de datos de forma discontinua. Esto supuso un gran avance ya que permitía la conexión a internet de los dispositivos móviles con una velocidad de transmisión de datos mucho mayor que la del GSM (hasta diez veces más).

## 4.2. Microcontrolador con comunicación GSM/GPRS

Una vez explicadas las comunicaciones GSM y GPRS, buscamos que posibilidades nos ofrece Arduino para poder hacer uso de ellas. Hasta hace poco tiempo lo más común era usar una placa de Arduino básica añadiéndole una *shield* que fuese la encargada de establecer las comunicaciones, pero en 2017 Arduino implementó un microcontrolador que permite esta comunicación con una única placa base. Este modelo es la placa Arduino MKR GSM 1400 (Figura 34) y es una placa similar a Arduino, pero con un módulo integrado (u-blox SARA – U201) que permite la conexión a la red GSM.



*Figura 34. Arduino MKR GSM 1400*

Con esta placa podemos hacer el control de todo el circuito inferior además de realizar la comunicación GSM. En la tienda oficial de Arduino la podemos encontrar por 60€ con una antena y una tarjeta SIM gratuita durante 3 meses y con un recargo mensual posterior de 1.5€ mensuales. Se alimenta con una tensión de 5 Voltios, pero tiene una tensión de trabajo de 3.3 Voltios, aspecto que hemos debido tener en cuenta a la hora de diseñar hardware de este circuito.

Otro aspecto importante de esta placa es su consumo, ya que cada vez que establecemos comunicación mediante el módulo u – blox SARA – U201, tenemos un consumo pico de 2 Amperios. Este valor no presenta un problema ya que la placa se conecta directamente a los 12V provenientes de las placas solares los cuales no tiene problema de suministrar corriente.

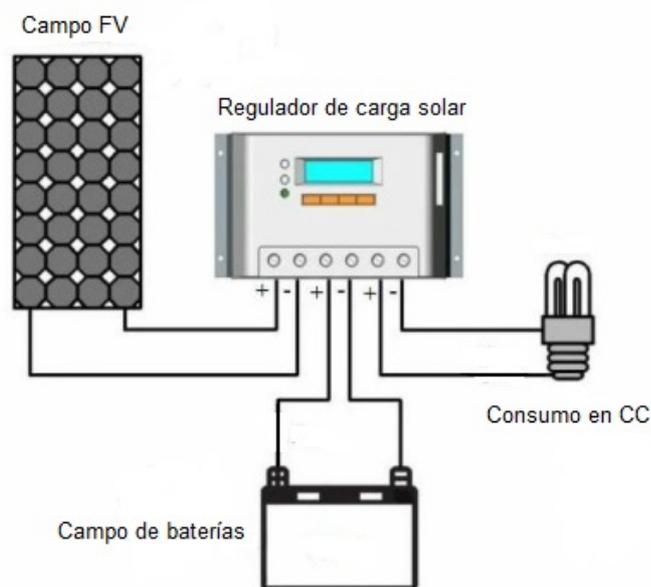
## **Capítulo 5. Alimentación mediante placas solares**

La necesidad de usar placas solares surge de la falta de suministro eléctrico en el huerto para alimentar a las componentes que requieren un elevado consumo de potencia (bombas y electroválvula). El objetivo de la instalación es proporcionar suministro eléctrico a los diferentes componentes de ambos circuitos, los cuales se han escogido a conciencia con una tensión nominal máxima de 12 Voltios. Al haber elegido todos los componentes para funcionar en corriente continua podemos obtener la tensión directamente de las placas solares y no necesitamos un inversor CC/CA ya que no tenemos ningún componente que utilice corriente alterna [26].

Tanto las bombas como la electroválvula tendrán un horario programado para funcionar, el cuál puede ser a cualquier hora del día, incluso de noche. Esto implica que por la noche la instalación debe ser capaz de suministrar la corriente requerida por los componentes. De ahí surge la necesidad de utilizar baterías para que durante el día se vaya almacenando energía en ellas y que por la noche si fuese necesario estas proporcionasen la corriente correspondiente.

Por último, necesitamos un componente que controle qué energía obtenida por las placas solares se consume directamente por el circuito y cual se almacena. Este componente es el regulador.

Por tanto, la instalación (Figura 35) queda compuesta por las placas solares, cuya energía pasará al regulador que será el encargado de distribuirla entre las baterías y los componentes del circuito.



*Figura 35. Instalación fotovoltaica*

Ahora que ya conocemos cómo será la instalación necesitamos dimensionar las placas solares y las baterías para poder buscar en el mercado las más adecuadas. Partimos de que la temporada de cultivo únicamente será desde finales de mayo a septiembre. Este hecho es una ventaja ya que únicamente necesitamos hacer uso de la energía de las placas solares durante esos meses, en los cuales la cantidad lumínica es mayor y por tanto las placas necesitadas para la misma potencia serán menores.

Lo primero es realizar un estudio aproximado de la potencia que necesitamos analizando cada bloque por separado en el peor caso. Utilizamos la palabra bloque para referirnos al componente en sí además de otros que son necesarios para su funcionamiento (por ejemplo, en la bomba de agua está incluido el consumo de los transistores necesarios).

### Consumos circuito superior

Componente	Consumo (W)	Número de horas/día	Consumo Wh/día
Arduino Nano 33 Every	0,05	24	1,2
Electroválvula	3 (cada conmutación)	12 veces	36
Bomba de agua	50,2	2	100,4
Sensor de nivel (x2)	0,005	24	0,12
Encoder rotativo (x2)	0,066	24	2,4
	Consumo total (KW/día)		140,12

Tabla 5. Consumos del circuito superior

### Consumos circuito inferior

Componente	Consumo (W)	Número de horas/día	Consumo (Wh/día)
Arduino MKR GSM 1400	6,6	3	19,8
Sensor de humedad del suelo	0,0165	1	0,0165
Sensor de humedad y temperatura	0,00825	1	0,00825
Bomba de agua	120,8	2	241,6
Encoder rotativo	0,033	24	1,2
LCD 20*4	0,425	1	0,625
	Consumo total (KW/día)		263,25

Tabla 6. Consumos del circuito inferior

Entre los dos circuitos tenemos un consumo máximo aproximado de 403,4 Watios hora/día, al cual le vamos a aplicar un rendimiento del 85% debido a pérdidas en las placas solares, el regulador y la batería, de la tal forma que la potencia necesaria será de 474Wh/día.

Una vez conocemos la potencia necesaria lo más conveniente para el proyecto es buscar en el mercado un pack formado por las placas solares, el regulador de carga y la batería que se acerque a las condiciones que necesitamos y la solución que mejor se adapta a nuestras necesidades es la siguiente:

🏠 Kit Solar Iluminación 12V 500Whdia



**291,99€** SIN IVA 241,31 € ★★★★★ [2 Opiniones](#)

Envío: Entrega gratuita en España!

Entrega: **Recíbelo el martes 27 de abril en pedidos antes de las 15h** >

Fabricante: Must Solar | Cod. Artículo: 4002005

Cantidad:  - +

Financiación: [Calcular cuota](#)

Servicios:  Instalación 871,20 € >

**AÑADIR AL CARRITO** **COMPRAR**

Figura 36. Pack placa solar, regulador y batería

El único problema que tiene este kit es que el regulador que incorpora tiene una intensidad máxima de salida de 10 Amperios y ya sólo la bomba de riego consume cerca de ese valor, por tanto, necesitamos uno de mayor capacidad. La empresa permite modificar ese regulador y coger el siguiente modelo de hasta 20A incrementando el precio en sólo 10€ y así el pack total quedaría por 300€.

## Capítulo 6. Desarrollo del software

Al igual que hemos realizado en la parte de hardware, vamos a estudiar el desarrollo del software de forma independiente para cada uno de los circuitos.

### 6.1 Software del circuito superior

Podemos describir el funcionamiento general del circuito con tres diagramas de flujo.

- El primero muestra las tareas a realizar en la interrupción producida al mover uno de los dos selectores.
- El segundo es para controlar el riego mediante la electroválvula.
- El tercero realiza el control del llenado de los dos bidones desde el depósito mediante la bomba de agua.

### 6.1.1. Interrupción del encoder rotativo

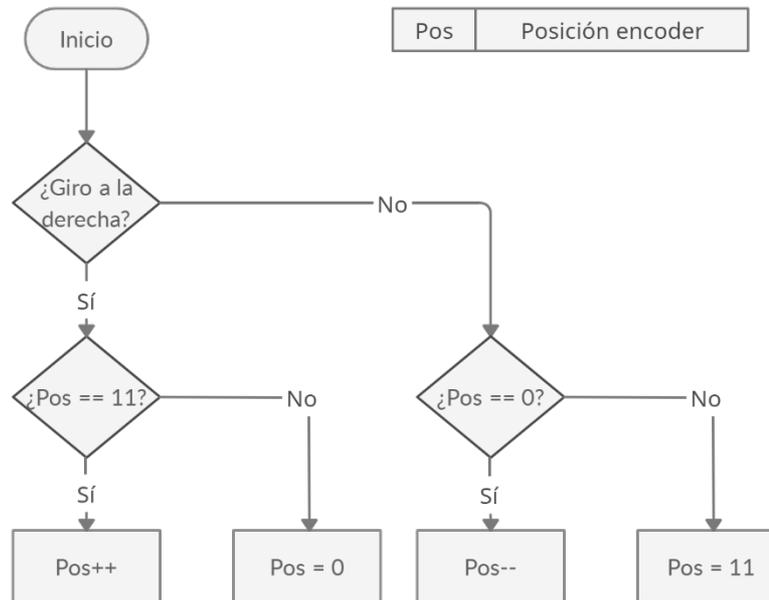


Figura 37. Diagrama de flujo encoder rotativo

La Figura 37 muestra la programación de la interrupción producida al modificar la posición de cualquiera de los dos selectores. Tendremos dos variables diferentes, una para el selector de duración (posDur) y otra para el de frecuencia (posFrec) que se modificarán cada una cuando se modifique la posición de su respectivo encoder rotativo. De esta forma conociendo el valor de cada una de las posiciones de duración y frecuencia podremos gestionar el programa de acuerdo con las equivalencias de la Tabla 3 expuestas anteriormente en el Capítulo 2.

En la pregunta “¿Giro a la derecha?” el controlador estará comprobando qué canal conmuta primero para conocer así el sentido de giro como ya se ha explicado en el apartado respectivo al encoder.

### 6.1.2. Control de la electroválvula

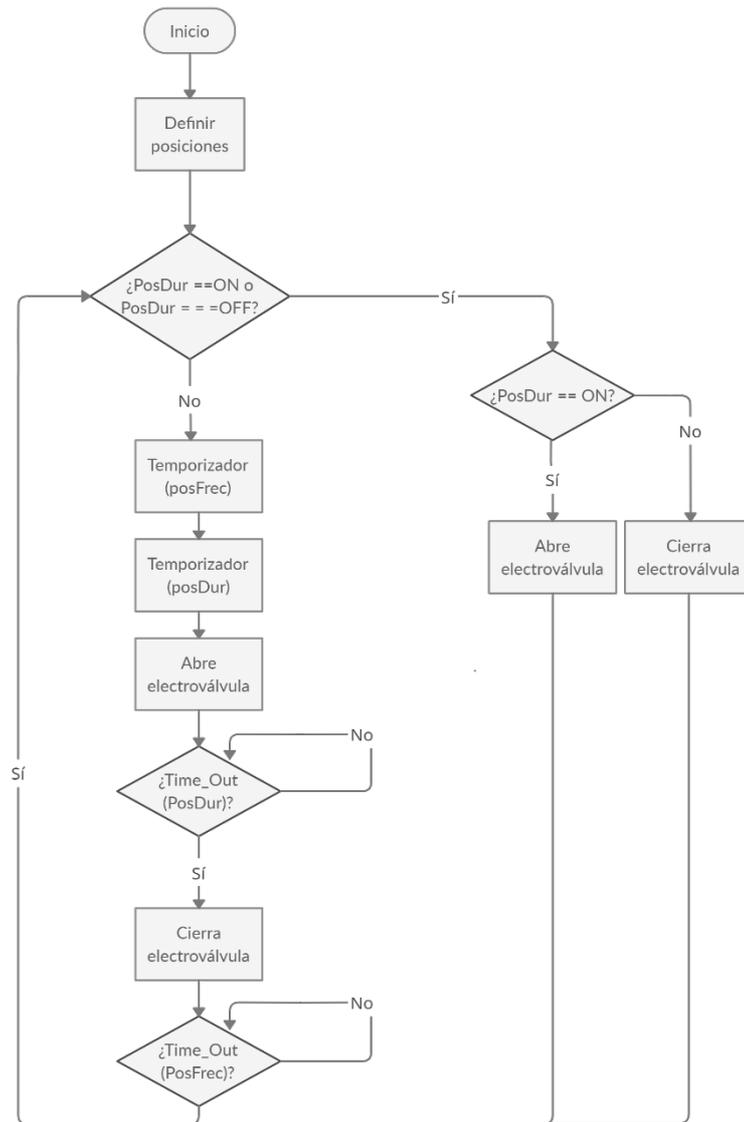


Figura 38. Diagrama de flujo electroválvula

En este caso de acuerdo con la Figura 38 gestionamos cuando se tiene que abrir y cerrar la electroválvula. Tal como comentamos en el apartado de selección de la electroválvula, las posiciones ON y OFF son las más restrictivas y así se muestra en el diagrama. Esta función se repite constantemente en el tiempo hasta que salta una interrupción en uno de los selectores, modificando así la variable correspondiente, y una vez finalizada la interrupción se continúa con el programa principal.

### 6.1.3. Llenado de los bidones

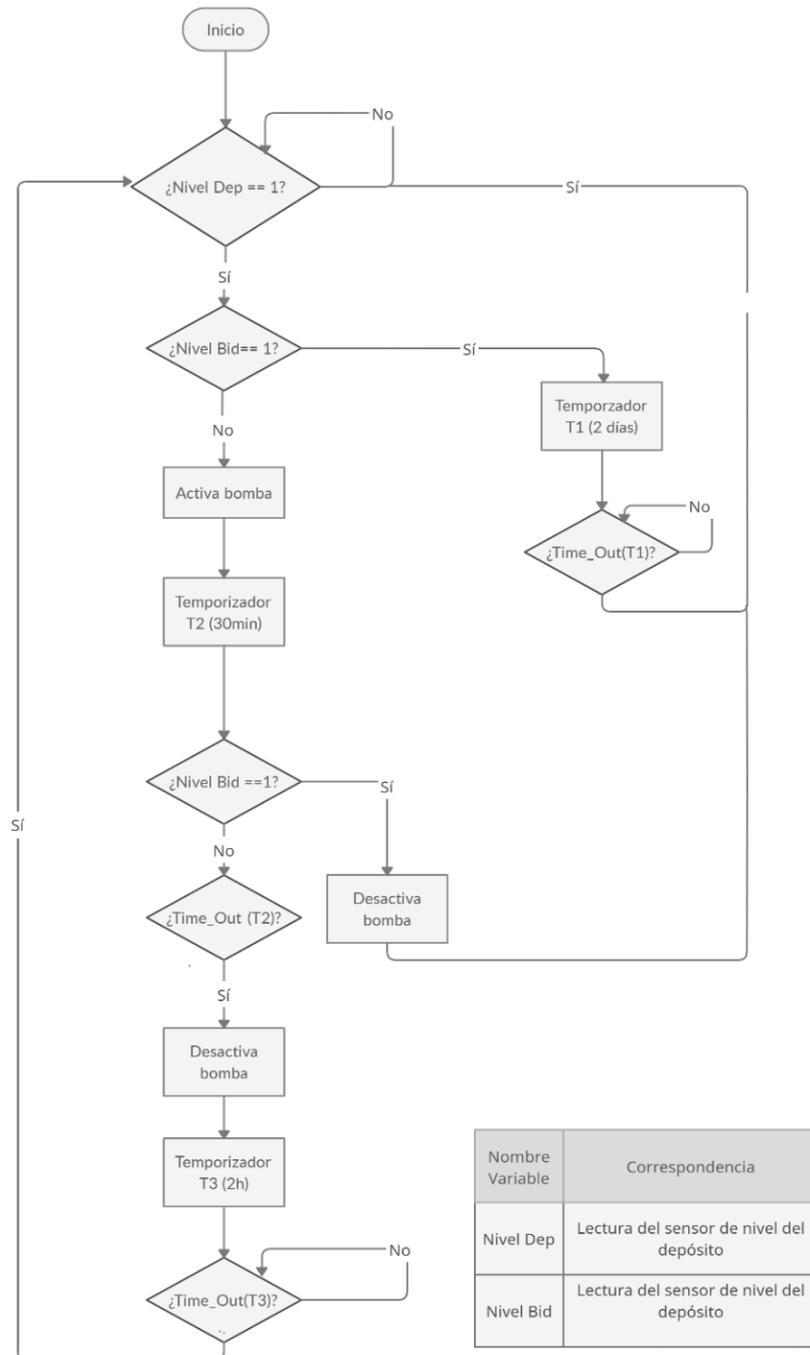


Figura 39. Diagrama de flujo llenado de los bidones

Este último diagrama gestiona el llenado de los bidones desde el depósito mediante la bomba. Primero comprobamos que el depósito esté por encima del valor establecido al colocar el sensor de nivel y que los bidones no estén llenos. En caso de que estén llenos, aplicamos un temporizador de 2 días que va a realizar una función de histéresis por programa. Esto lo utilizamos para evitar que cada vez que se desactive el *switch* debido a

una leve bajada del nivel se vuelva a activar la bomba. De esta forma una vez estén los bidones llenos esperaremos 2 días hasta volver a llenarlos. Hemos decidido fijar ese periodo ya que como hemos comprobado años anteriores con riego manual, durante este tiempo es suficiente para mantener los cultivos bien regados en la época más calurosa sin agotar toda el agua de los bidones.

Por el contrario, si no están completamente llenos, activaremos la bomba hasta que se llenen los bidones o pase un tiempo máximo de 30 minutos para evitar dañarla tal y como nos indica el fabricante. Si en este tiempo no se han llenado, esperaremos 2 horas para que repose la bomba y a continuación retomaremos el proceso de llenado. Si por el contrario sí le da tiempo a llenarse, se desactivará la bomba y volveremos al temporizador de 2 días para realizar el próximo ciclo de llenado.

## **6.2 Software del circuito inferior**

Primero plantearemos los diferentes modos de funcionamiento y explicaremos las características de cada uno de ellos. A continuación, trataremos el riego del circuito mediante la bomba y finalmente analizaremos la manera de manipular el programa de riego a través del encoder y la pantalla LCD y de la página WEB.

### **6.2.1. Modos de funcionamiento**

El riego del circuito inferior se puede controlar de tres formas diferentes: con los sensores, con el bloque de control y visualización (encoder rotativo y LCD) o a través de la página WEB. Por ello hemos considerado que la mejor opción para controlar su funcionamiento es dividir en dos modos de funcionamiento: automático y manual.

Utilizaremos una variable para almacenar el modo de funcionamiento actual del programa. Esta variable se puede modificar mediante el selector o a través de la página WEB, de tal manera que cada cambio sobrescribirá el anterior. De esta forma el programa se iniciará en automático y cada vez que se modifique el modo a través de la página o mediante el selector en el huerto, se guardará el nuevo valor y el programa actuará en consecuencia.

Para explicar este funcionamiento vamos a utilizar el diagrama de flujo de la Figura 40.

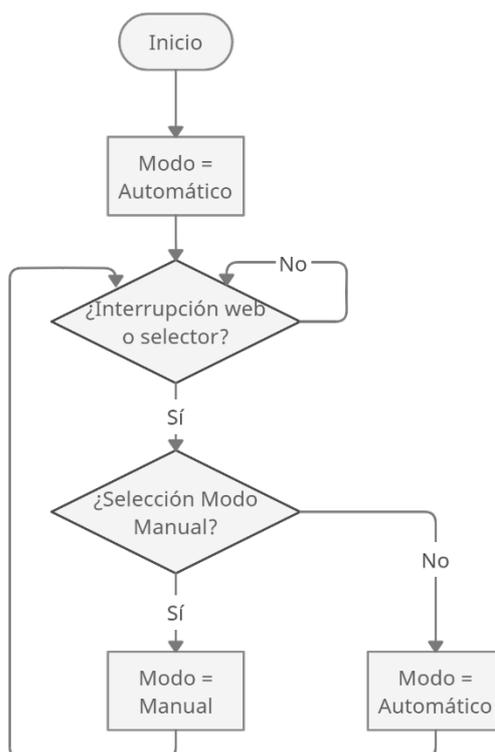


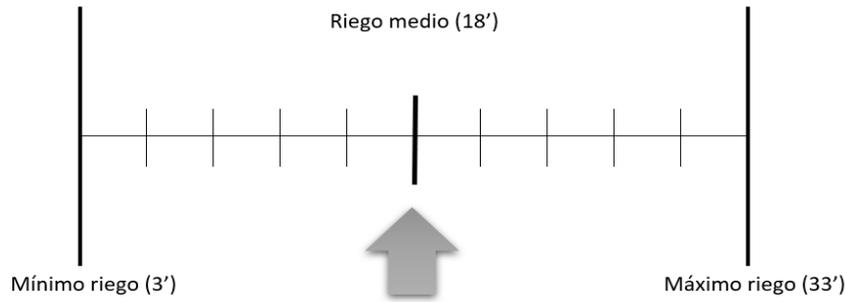
Figura 40. Diagrama de flujo de los modos de funcionamiento

Una vez explicado el funcionamiento del cambio de un modo a otro, vamos a profundizar un poco más en los dos modos:

### 6.2.2. Modo automático

Este modo tiene un programa inicial de riego decidido de acuerdo con un estudio estadístico de la frecuencia y duración adecuada que se ha realizado durante los años anteriores. De acuerdo con este estudio, el programa que más se acerca al óptimo para este circuito es regar 18 minutos cada 6 horas.

Con el objetivo de facilitar el programa, hemos decidido modificar únicamente el parámetro de duración. Por tanto, en función de las mediciones de los sensores, el programa automáticamente podrá cambiar la duración del riego de cada ciclo dentro de unos valores predeterminados. Para ello, este modo establece esas características como punto de partida y realiza medidas de los sensores de humedad del suelo, de temperatura y de humedad del aire justo antes de cada ciclo de 6 horas a través de las interrupciones de reloj. La idea que expresa el funcionamiento de este modo es la siguiente:



*Figura 41, Funcionamiento modo automático*

Partimos del riego medio que es 18 minutos cada 6h. En función de los valores de los sensores, el parámetro duración variará en alguna de las diferentes posiciones de la Figura 41, siendo el máximo 33 minutos y el mínimo 3 minutos.

Inicialmente estableceremos unos valores de humedad del suelo y de humedad y temperatura del ambiente para cada uno de los diferentes tramos que nos permitan variar el parámetro de la duración. Por ejemplo: cuando la humedad del suelo es superior a 80% y la temperatura es inferior a 20°C, fijamos la duración en 3 minutos.

Para ajustar correctamente los valores de los sensores a partir de cuales modificaremos la duración del programa, consideramos que la mejor solución es realizar un estudio a largo plazo del valor obtenido por estos sensores comparándolos con la situación real de la tierra y del ambiente. De esta forma se espera ir modificando estos valores en función del estudio hasta encontrar los que más se acerquen al funcionamiento óptimo de un modo automático.

### **6.2.3. Modo manual**

Este modo es igual tanto para la página WEB como para el control del menú con el selector, su funcionamiento consiste en elegir la frecuencia y la duración de riego y regar en función de sus valores.

### **6.2.4. Control del riego mediante la bomba**

Una vez explicados ambos modos vamos a estudiar el funcionamiento del riego mediante la bomba, el cuál va a ser el mismo para los dos modos ya que este programa únicamente decide si activar o no la bomba en función a los parámetros de frecuencia y duración (ya sean pasados por los sensores o por nosotros manualmente).

Este funcionamiento se parece mucho al de la electroválvula del circuito superior, a diferencia de una restricción más que hemos añadido cuando el parámetro frecuencia se encuentra en el valor ON. Esto es debido a que la electroválvula podía estar en modo ON todo el tiempo ya que una vez abierta no requería más corriente y no había peligro de dañarla. Sin embargo, la bomba, al igual que la bomba de llenado de los depósitos no puede estar indefinidamente encendida. Como estudiamos en el apartado de la bomba, esta puede funcionar 90 minutos hasta llegar a la temperatura máxima del motor. Para evitar llegar a ese punto hemos decidido que, si utilizamos el modo ON para regar de forma continua, vamos a regar 60 minutos y a continuación la dejaremos reposar otros 60 minutos. El funcionamiento se puede ver en el diagrama de flujo de la Figura 42.

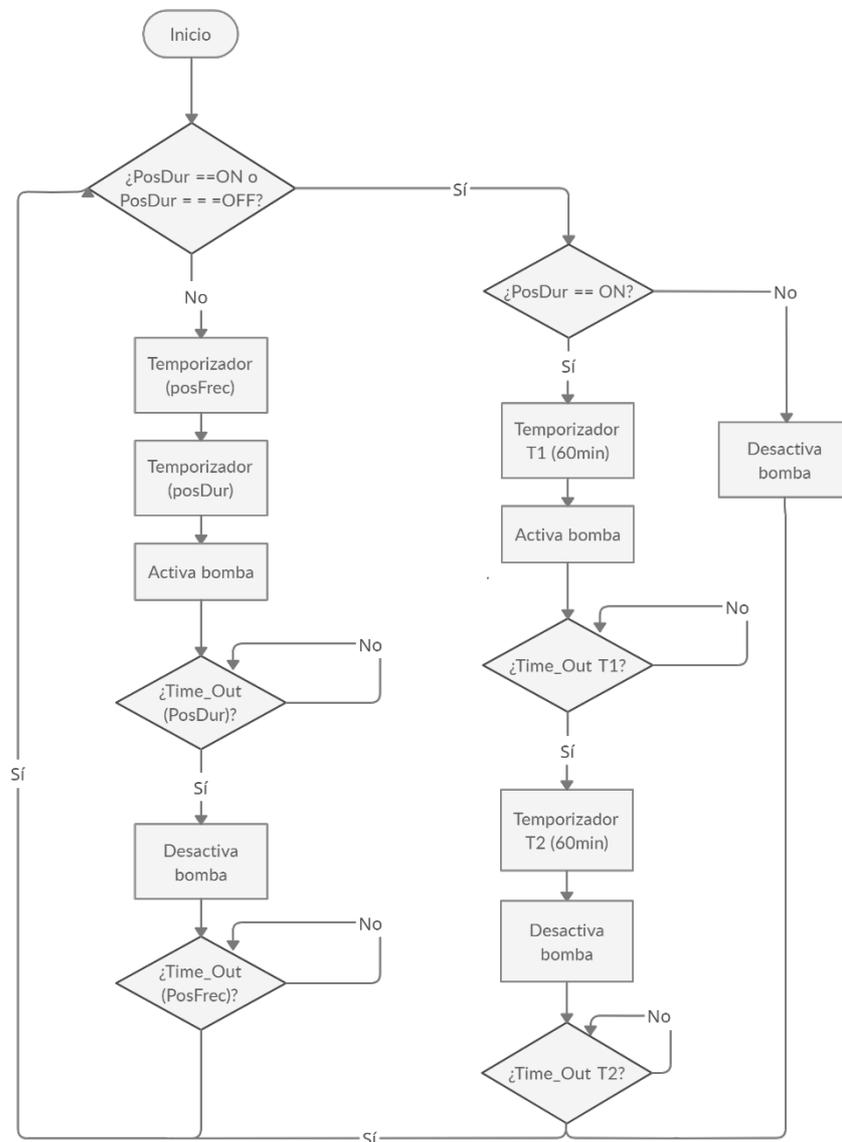


Figura 42. Diagrama de flujo del funcionamiento de la bomba

## 6.2.5. Control mediante pantalla LCD y encoder rotativo

En el diagrama de flujo de la Figura 43 vamos a explicar el uso de la pantalla LCD 20x4, encoder rotativo y máquina de estados que permite realizar los cambios necesarios.

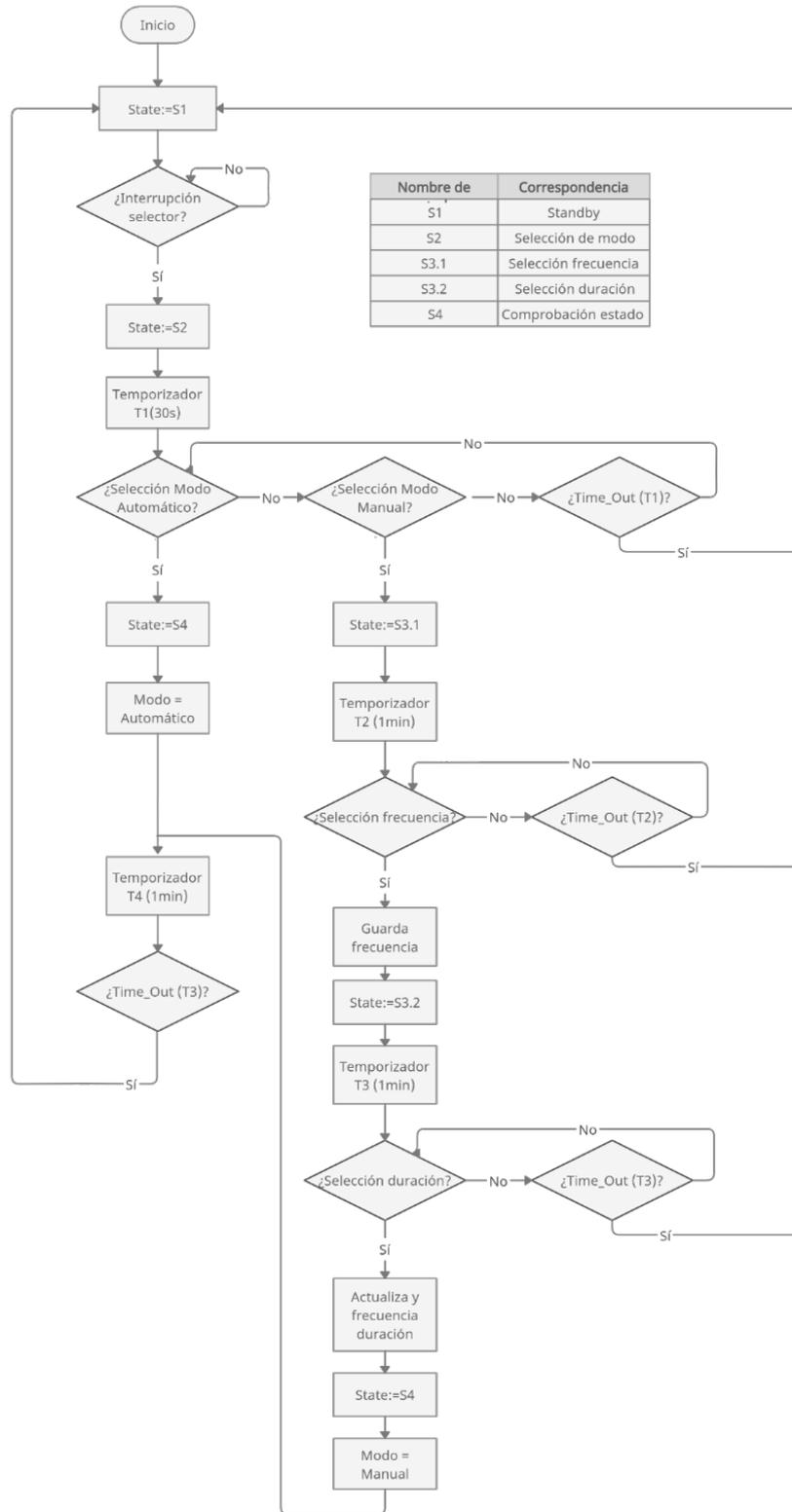


Figura 43. Diagrama de flujo del funcionamiento de la pantalla LCD con el selector

Como vemos en la tabla de la Figura 43 tenemos 5 estados diferentes, el primero (S1) es *Standby* y lo utilizamos para dejar la pantalla apagada en *Standby* y así reducir el consumo (únicamente consumirá el tiempo que se esté utilizando para realizar cambios).

El segundo estado es el S2, que primero nos informará del modo que está funcionando actualmente y de la frecuencia y la duración con la que se está regando. A continuación, nos dará la posibilidad de cambiar a uno de los dos modos. Si pulsamos el modo manual nos mandará al estado S3.1 donde elegiremos entre las frecuencias ofrecidas, y a continuación al estado S3.2 para escoger la duración. Una vez elegidos los dos parámetros modificará el valor de “Modo” y pasará al estado S4.

Si por el contrario elegimos el modo automático, cambiará el valor de “Modo” y nos mandará directamente al estado S4. En este estado nos dirá que el programa se ha modificado y nos ofrecerá información de los cambios realizados. Por último, hay que añadir que en cada decisión a tomar se ha programado un temporizador de 1 minuto para que en caso de que no se termine de realizar correctamente el proceso se vuelva al estado *Standby* apagando la pantalla y ahorrando energía.

En el diagrama podemos ver que en el paso del estado 3.1 al 3.2, la variable frecuencia se guarda en una variable local y después de seleccionar la duración ambas se actualizan en las variables de programa al mismo tiempo. Esta decisión se hace para evitar que, si en el momento de elegir la duración se nos acaba la temporización, se actualice la variable frecuencia únicamente, sin modificar la duración y el modo.

Con el objetivo de tener una mejor visión del funcionamiento, hemos simulado los mensajes que aparecerán en la pantalla LCD en los diferentes estados:

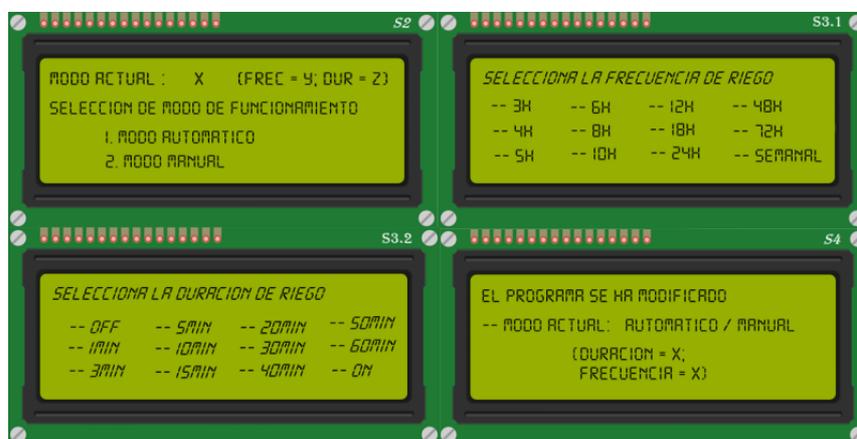


Figura 44. Pantallas de los cuatro estados de la LCD

## 6.2.6. Control mediante página WEB

El control mediante la página WEB se realiza gracias a la comunicación a distancia establecida mediante el módulo GSM tratado en el Capítulo 4. Con la página se pretende realizar dos funciones principalmente:

- La primera consiste en modificar el programa de riego, ya sea para cambiar de modo de funcionamiento (de automático a manual o viceversa) o para dentro del programa manual modificar los valores de duración y frecuencia.
- La segunda idea es realizar un historial de los valores recibidos por los sensores, de tal forma que, por ejemplo, podamos ver cuál ha sido la temperatura máxima o mínima de la temporada o conocer la humedad de la tierra cada cierto tiempo.

De acuerdo a esta división de tareas, al entrar en la página WEB la primera interfaz será la siguiente:



Figura 45. Pantalla principal página WEB

Si nos metemos en la primera categoría accederemos a la siguiente ventana:



Figura 46. Pantalla para modificar el programa de riego

En esta pantalla seleccionaremos entre modo automático y en modo manual y para la frecuencia y la duración del modo manual hemos decidido mostrar unos desplegables con los valores posibles con el objetivo de evitar valores erróneos. El funcionamiento del programa para esta funcionalidad lo podemos ver en el diagrama de flujo de la Figura 47.

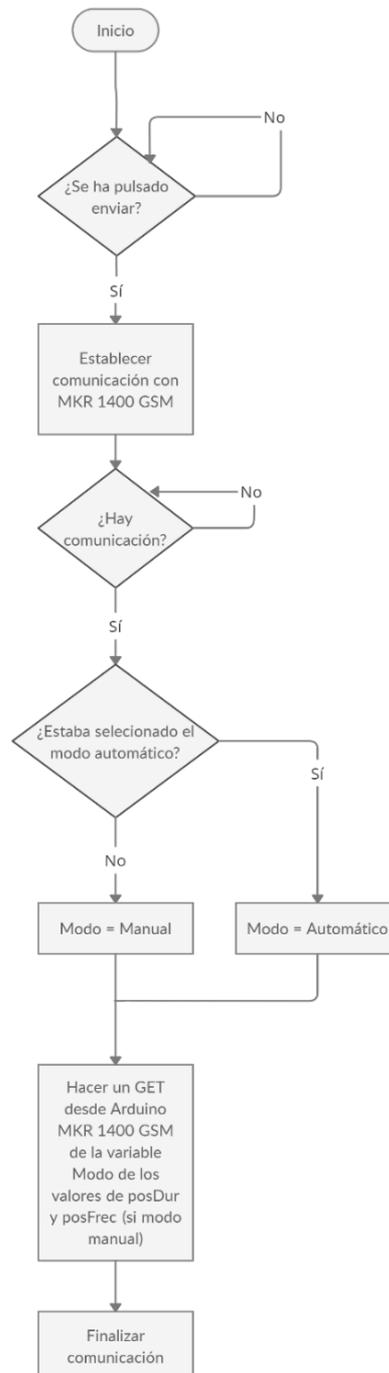


Figura 47. Diagrama de flujo modificación de programa de riego

Por otro lado, accediendo a historial de sensores la interfaz será la siguiente:



Figura 48. Pantalla del historial de los sensores

Como podemos ver en la parte superior, tenemos tres pestañas, una igual para cada magnitud. En cada sección tendremos una tabla con el historial de cada sensor con el día, la hora y el valor enviado y en el lado opuesto unos recuadros con los valores máximos y mínimos. Los diagramas de flujo de la Figura 49 muestran el funcionamiento para la temperatura, aunque sería igual para las tres variables.

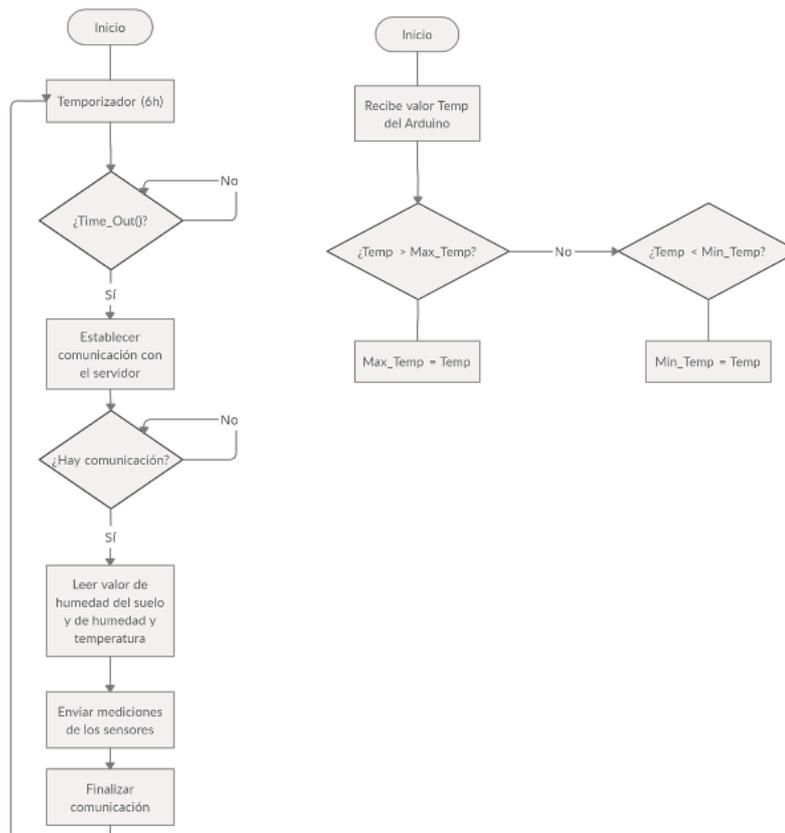


Figura 49. Diagramas de flujo historial de sensores

## Capítulo 7. Prototipo

Como se ha comentado en la introducción, las bases sobre las que se crea este proyecto es un problema real del riego de un huerto familiar. Para ello se decide realizar un estudio ambicioso acerca de la automatización de diferentes tareas con el objetivo de realizar un prototipo de todo el proyecto que sea funcional y solucione los problemas comentados. A medida que avanza el trabajo, se aprecia que el estudio teórico se vuelve cada vez más extenso, excediendo ampliamente el tiempo en créditos en el que un TFG es valorado. Aunque la idea inicial era implementar todos los bloques estudiados en el proyecto, con el fin de cerrar mi etapa personal dedicada al TFG (y a mis estudios), se decide realizar únicamente un prototipo de muestra. Para ello, hemos seleccionado el programa de llenado de bidones expuesto en el apartado 1.3 del Capítulo 6.

Este circuito consiste en dos sensores de nivel (uno en el depósito y otro en uno de los bidones) que en función del nivel encienden o apagan una bomba. Aunque en el estudio del control de este circuito se seleccionó una placa Arduino Nano Every, este prototipo voy a realizarlo con Arduino UNO ya que disponía de la placa con anterioridad y para la demostración he considerado que no era necesario comprar la otra placa.

Como todavía no tenemos alimentación a 12 Voltios proveniente de las placas solares, para esta demostración hemos decidido prescindir de la bomba ya que desde el punto de vista del programa no influye en nada que tengamos o no conectada la bomba.

Vamos a dividir el capítulo en diferentes apartados que están relacionados con la forma en la que se ha realizado el prototipo. Primero me he centrado en hacer funcionar el sensor de nivel con Arduino mediante interrupciones hardware (7.1), a continuación, en configurar temporizaciones mediante interrupciones de reloj (7.2) y, una vez funcionaba cada apartado por separado, en unificarlos siguiendo el diagrama de flujo del circuito de llenado de los bidones (7.3).

En los dos últimos apartados se muestra documentación fotográfica del prototipo tanto en fase de pruebas como ya fijado en su lugar de trabajo, a espera de completarse por el resto de componentes del proyecto (7.4 y 7.5).

## 7.1. Interrupción Sensor de Nivel

Como se ha explicado durante la memoria, el sensor de nivel va a trabajar mediante interrupción ya que estar comprobando constantemente su estado es inviable [27]. Estas interrupciones se denominan ISR (*Interrupt Service Routines*) [28] y nos permiten crear una función que sea llamada cada vez que se produzca un cambio en un pin físico.

Arduino UNO dispone de 2 interrupciones por hardware en el pin 2 y 3 y la sintaxis para utilizarlas es la siguiente:

```
“ attachInterrupt(digitalPinToInterrupt(pin), ISR, mode) “
```

Básicamente añadimos una interrupción de hardware al pin que tenemos conectado el sensor, hacemos que cada vez que salta la interrupción llame a la función ISR() y por último elegimos el modo de funcionamiento de la interrupción dentro de los que nos ofrece Arduino:

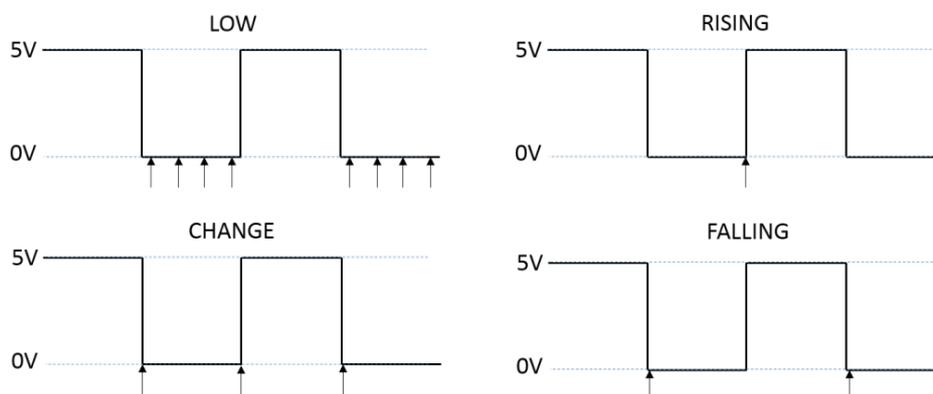


Figura 50. Modos de funcionamiento de la Interrupción

Como vemos en la Figura 50, Arduino permite generar interrupción cuando el pin está en *LOW* (estado bajo), *RISING* (pasa de 0 a 1), *CHANGE* (cada vez que cambia de estado de 0 a 1 o viceversa) y *FALLING* (conmuta de 1 a 0). En nuestro caso vamos a elegir *CHANGE* ya que nos interesa saber en cada momento que nivel de agua tenemos en los depósitos.

Una vez definida la interrupción solo faltaría definir la rutina a realizar, es decir la función ISR(), en la cual únicamente invertiremos el valor de una variable que se corresponderá con el estado del sensor. Esta variable se debe definir como *volatile* ya que su valor puede cambiar tanto en el bucle principal como en la interrupción. De esta forma

cada vez que se vaya a utilizar esa variable se leerá su valor primero para asegurarse que no haya cambiado.

De acuerdo con el esquema eléctrico del sensor de nivel expuesto en la Figura 9, los dos cables del sensor se van a conectar a GND y a un pin capaz de generar interrupciones. Esto obliga a definir el pin como *INPUT\_PULLUP* de tal forma que Arduino alimenta al pin al que está conectado el sensor con *Vcc* a través de una resistencia de *pull up* interna. De esta forma, si el sensor está abierto, en el pin se tiene un 1 y, si está cerrado, un 0.

El último tema a tratar se relaciona con los rebotes producidos en las conmutaciones del sensor. Como vemos en la Figura 51, cuando pasa de un estado a otro se puede generar ruido que incluso sea suficiente para generar de nuevo una interrupción de tal forma que nos cambiaría el estado de la variable y perderíamos la cuenta del nivel del sensor.

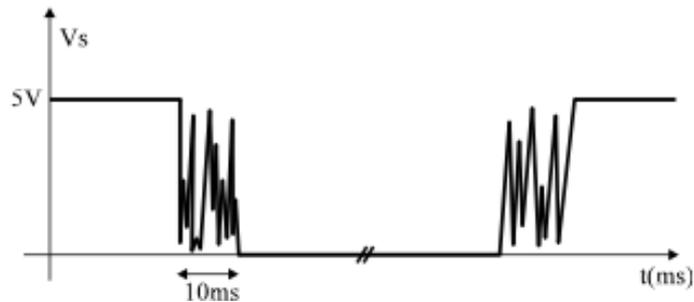


Figura 51. Rebotes en la conmutación del sensor de nivel

Para ello tenemos diferentes soluciones [34]: filtrar ese ruido por hardware mediante un condensador o circuitos más complejos o filtrarlo mediante software con temporizaciones de milisegundos en las cuales no se leerá el estado del sensor. La solución más fácil es utilizar un condensador de  $1\mu\text{F}$  en paralelo de la siguiente manera:

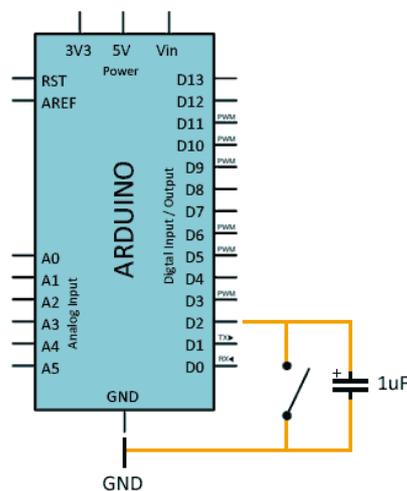


Figura 52. Filtrado de los rebotes mediante condensador

Una vez tratado todos los temas necesarios para hacer funcionar el sensor mediante interrupción de hardware, desarrollamos un programa básico (Anexos, Capítulo 4.1) que únicamente nos muestre por pantalla del ordenador cuando cambia de estado el sensor.

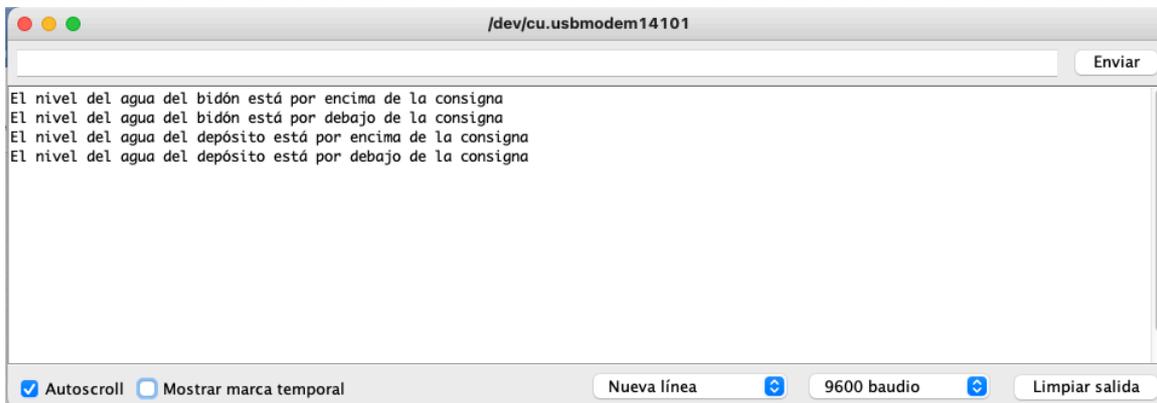


Figura 53. Mensajes mostrados al cambiar el estado de los sensores de nivel

## 7.2. Interrupción de reloj

Las interrupciones de reloj se utilizan prácticamente en todos los apartados del software, por ello esta sección demostrativa es la base de todas las temporizaciones utilizadas en el proyecto. Arduino permite hacer temporizaciones mediante la función *delay()* o a través de interrupción de reloj (*Timer*) [27] [29]. La diferencia es básicamente que la función *delay()* paraliza todo el programa el tiempo que dure la temporización de tal forma que el microcontrolador no puede trabajar de forma paralela. Sin embargo, de la otra manera podemos crear interrupciones cada cierto tiempo sin la obligación de parar el microcontrolador toda la temporización.

Arduino UNO ofrece la posibilidad de usar tres temporizadores diferentes: *Timer 0* es de 8 bits y se utiliza en las funciones *delay()* y *millis()*, *Timer 1* de 16 bits y lo utiliza la librería *servo* y el *Timer 2* que se usa en la función *tone()*. Como no vamos a utilizar la librería *servo* en nuestro trabajo, vamos a utilizar el *Timer 1* que ofrece un tiempo de desbordamiento mucho mayor al ser de 16 bits.

Las interrupciones de reloj se generan mediante hardware a través de un cristal de cuarzo (oscilador) ya incluido en la placa Arduino UNO y por software se realiza un contador para controlar el tiempo de temporización. La unidad de media se denomina *tick* y depende de la frecuencia de oscilación del microcontrolador, que en Arduino UNO es

de 16 Mega Hertzios. Por tanto, el microcontrolador es capaz de generar una interrupción cada 62.5 nano segundos, aunque como no necesitamos temporizaciones de menos de minutos utilizaremos el *prescaler* que nos permite aumentar el periodo de generar interrupciones hasta en un factor de 1024.

Con el *prescaler* de 1024 generamos una interrupción cada 1/15.625 Kilo Hertzios es decir un *tick* de 64 micro segundos. Con el *Timer* de 16 bits, el contador será capaz de contar hasta 4.19424 segundos sin producir desbordamiento (Figura 54), entonces podemos seleccionar un periodo de interrupción de 4 segundos de tal forma que cada vez que el contador llegue a ese valor se pondrá a 0 y volverá a empezar.

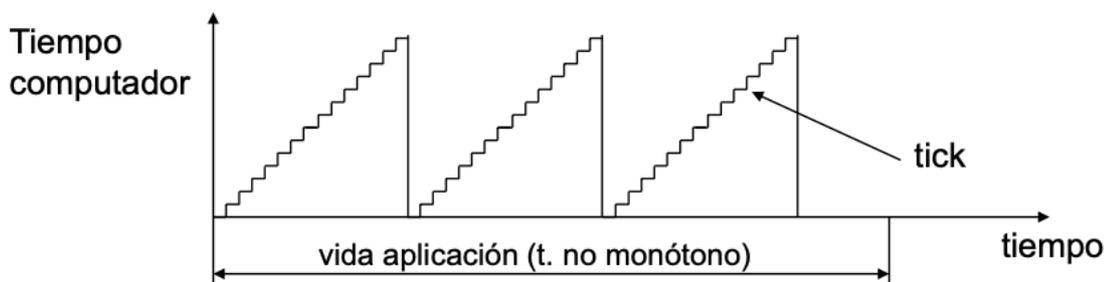


Figura 54. Valor del contador de ticks en cada temporización

Una vez fijados los valores con los que queremos fijar la interrupción de reloj, investigamos en la hoja de características del procesador de Arduino UNO (ATmega329P) qué registros debemos configurar. En la hoja de características vemos que el *Timer 1* tiene tres registros de 16 bits para la configuración de: temporizaciones de ejecuciones de programa (TCNT1, *Timer/Counter*), generación de ondas (OCR1A/B, *Output Compare Registers*) y medición de tiempo en señales (ICR1, *Input Compare Registers*).

Además, cuenta cuatro registros de 8 bits, dos para realizar el contador de ticks (TCCR1A/B, *Timer/Counter Control Registers*) y otros dos para el *flag* (banderín) de la interrupción (TIFR1, *Timer Interrupt Flag Register*) y para el enmascaramiento de la misma (TIMSK1, *Timer Interrupt Mask Register*).

Una vez analizado los registros del *Timer 1*, procedemos a configurar los necesarios para el funcionamiento que necesitamos. Primero ponemos el registro TCCR1A a 0, para ajustar el *Timer* en *Normal Mode*. A continuación, fijamos el *prescaler* en 1024 que lo hacemos dándole el valor de 1 a los bits CS12 y CS10 (Figura 55).

### 16.11.2 TCCR1B – Timer/Counter1 Control Register B

Bit (0x81)	7	6	5	4	3	2	1	0	TCCR1B
	ICNC1	ICES1	–	WGM13	WGM12	CS12	CS11	CS10	
Read/Write	R/W	R/W	R	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Table 16-5. Clock Select Bit Description

CS12	CS11	CS10	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	clk <sub>IO</sub> /1 (No prescaling)
0	1	0	clk <sub>IO</sub> /8 (From prescaler)
0	1	1	clk <sub>IO</sub> /64 (From prescaler)
1	0	0	clk <sub>IO</sub> /256 (From prescaler)
1	0	1	clk <sub>IO</sub> /1024 (From prescaler)
1	1	0	External clock source on T1 pin. Clock on falling edge.
1	1	1	External clock source on T1 pin. Clock on rising edge.

Figura 55. Registro TCCR1B

Después calculamos con la siguiente fórmula (Figura 56) el registro de comparación para realizar el contador de ticks con la frecuencia de 4 segundos que hemos comentado anteriormente, obteniendo un valor de 3035.

$$TCNT1 = 65535 - \frac{16,000,000\text{Hz} * \text{T tiempo (s)}}{\text{Prescaler}}$$

Figura 56. Fórmula para calcular el registro de comparación del Timer

Por último, habilitaremos las interrupciones con el registro TIMSK1 a través del bit TOIE1 (Figura 57).

### 16.11.8 TIMSK1 – Timer/Counter1 Interrupt Mask Register

Bit (0x6F)	7	6	5	4	3	2	1	0	TIMSK1
	–	–	ICIE1	–	–	OCIE1B	OCIE1A	TOIE1	
Read/Write	R	R	R/W	R	R	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

• **Bit 0 – TOIE1: Timer/Counter1, Overflow Interrupt Enable**

When this bit is written to one, and the I-flag in the Status Register is set (interrupts globally enabled), the Timer/Counter1 Overflow interrupt is enabled. The corresponding Interrupt Vector (See "Interrupts" on page 66) is executed when the TOV1 Flag, located in TIFR1, is set.

Figura 57. Registro TIMSK1

Para probar la interrupción hacemos un código de prueba (Anexos, Capítulo 4.2) que muestre un mensaje por pantalla cada vez que se genere una interrupción cada 4 segundos e incremente en una unidad el valor de una variable *counter*, de tal forma que cada unidad se corresponda con 4 segundos.

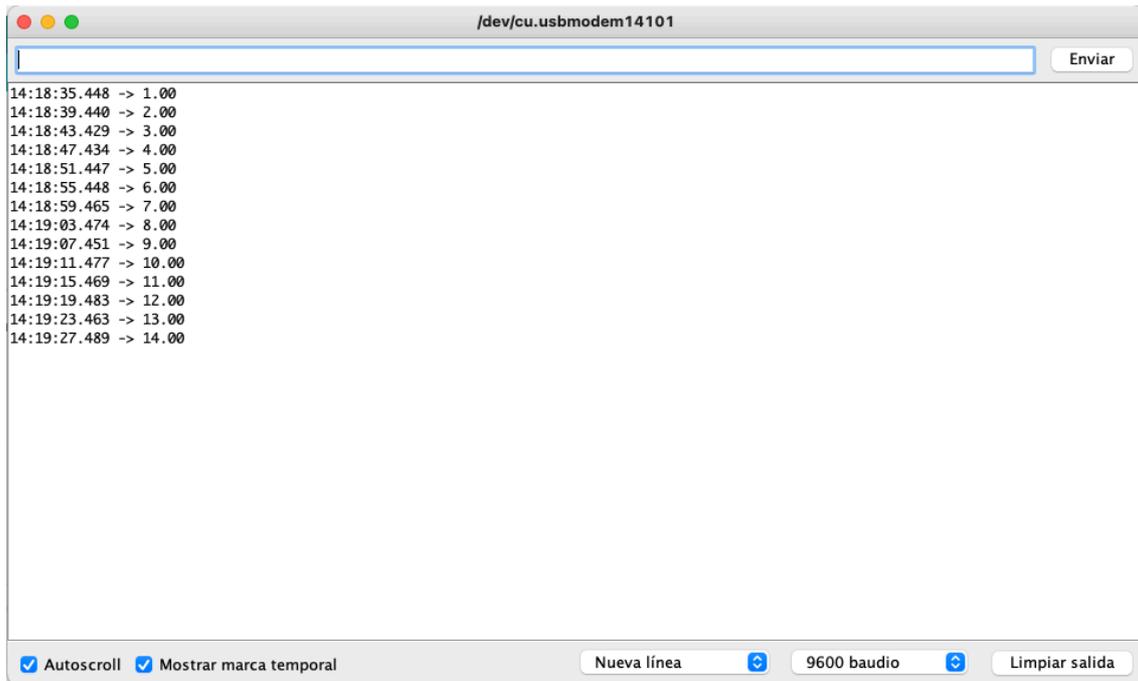


Figura 58. Interrupción cada 4 segundos

Hasta aquí hemos conseguido generar una interrupción cada 4 segundos, pero el verdadero propósito de las interrupciones del reloj en nuestro trabajo es poder iniciar temporizaciones esporádicas. Para realizar esta tarea vamos a utilizar tres variables y tres funciones:

Las variables son *Active\_Timer*, *T0* y *Final\_Time*, las dos primeras se utilizan de banderines para saber si está activado el temporizador y si ha finalizado y la tercera almacena el valor final al cual el contador tendrá que llegar para haber terminado la temporización.

Por otro lado, las funciones son *Get\_Time()*, *Set\_Timer*(n segundos) y *Time\_Out()*. La primera devuelve una copia de la variable contador para evitar modificarla en tiempo real, la segunda aplicará a *Final\_Time* la suma de Counter y del tiempo en segundos que queremos temporizar dividido entre 4. Y por último *Time\_Out()* devolverá el valor el banderín T0 que se podrá a 0 cuando termine la temporización.

Con el objetivo de probar las funciones comentadas, modificamos el programa anterior (Anexos, Capítulo 4.2) para crear desde el bucle principal un temporizador repetitivo de 12 segundos, de tal manera que cuando pase ese tiempo, nos avise y comience el siguiente tal como vemos en la Figura 58.

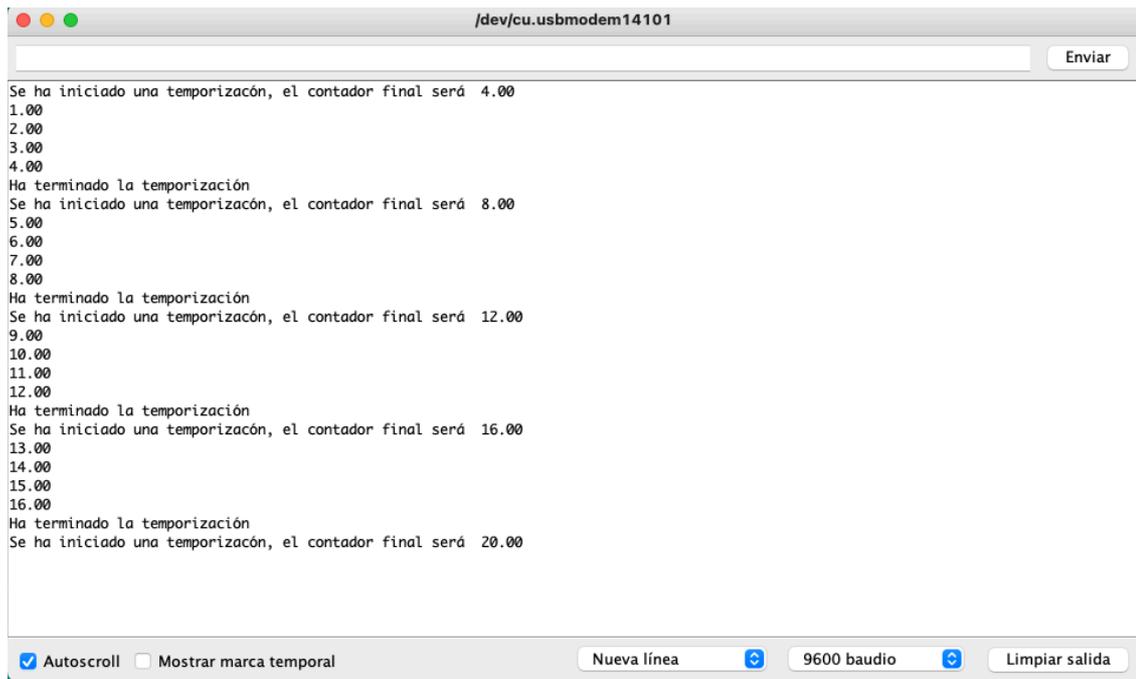


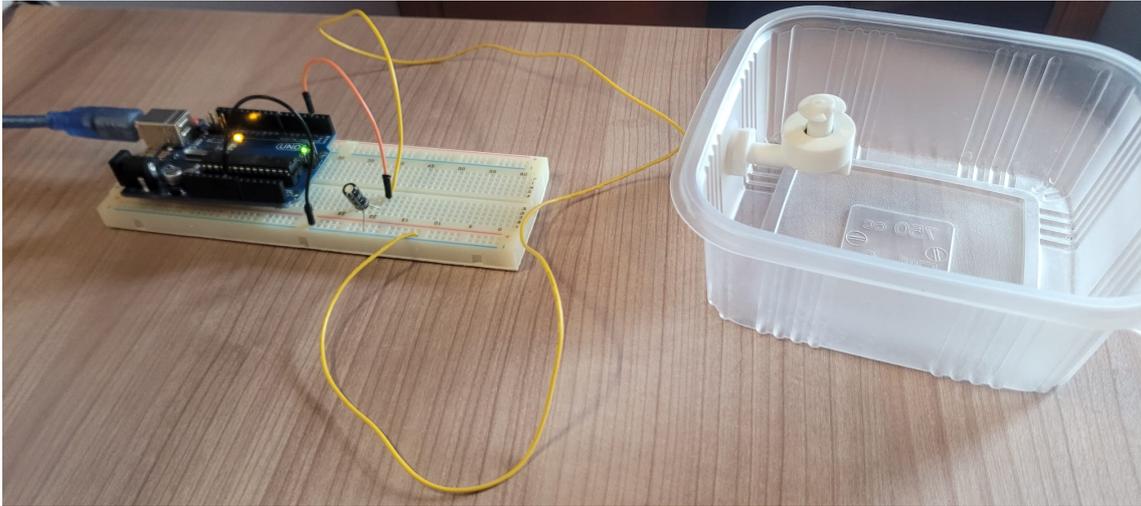
Figura 59. Temporizaciones repetitivas de 12 segundos

### 7.3. Programa completo

Una vez probadas cada parte de forma independiente realizamos el programa completo (Anexos, Capítulo 4.3) siguiendo el diagrama de flujo. Respecto a la bomba, por programa se activa o desactiva un pin como si estuviese conectada, por ello que cuando se decida comprarla y terminar de hacer el prototipo funcional, el código no debería necesitar modificaciones.

### 7.4. Pruebas del prototipo

Primero realizamos el conexionado del sensor de nivel y lo pusimos en marcha, al principio sin agua hasta hacer funcionar correctamente las interrupciones y posteriormente lo colocamos en un recipiente casero simulando el depósito y añadimos agua hasta variar el nivel y producir una interrupción. En las siguientes imágenes podemos ver el proceso:



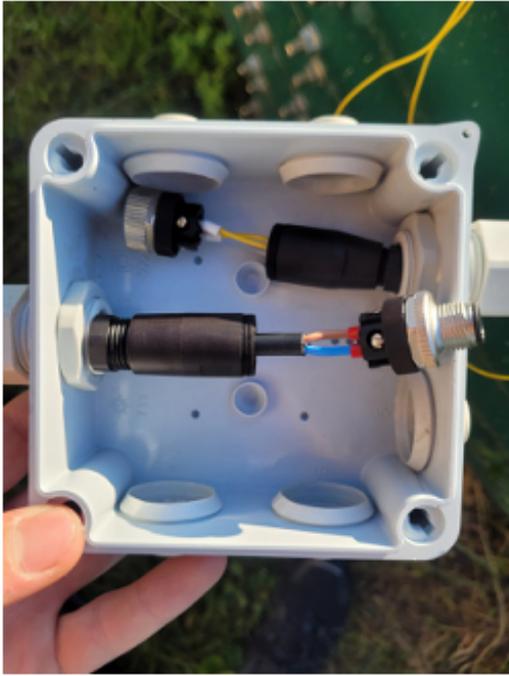
*Figura 60. Simulación del funcionamiento sin agua*



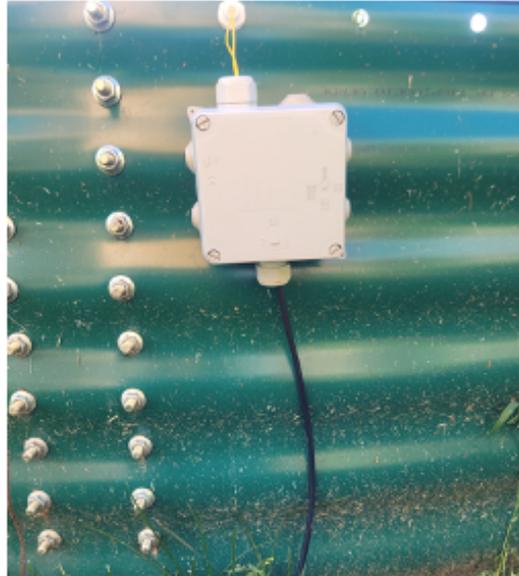
*Figura 61. Simulación del funcionamiento con el nivel alto*

## **7.5. Implantación del prototipo**

Una vez estaba el prototipo funcionando correctamente, era el momento de implantarlo en el huerto. Con el objetivo de que fuese duradero y no se viese dañado por la lluvia u otros efectos meteorológicos, nos preocupamos por utilizar diferentes elementos protectores como cajas estancas, prensas y conectores para los empalmes de los cables. El resultado final del montaje se muestra en las siguientes fotografías.



*Figura 62. Conexión del sensor del depósito*



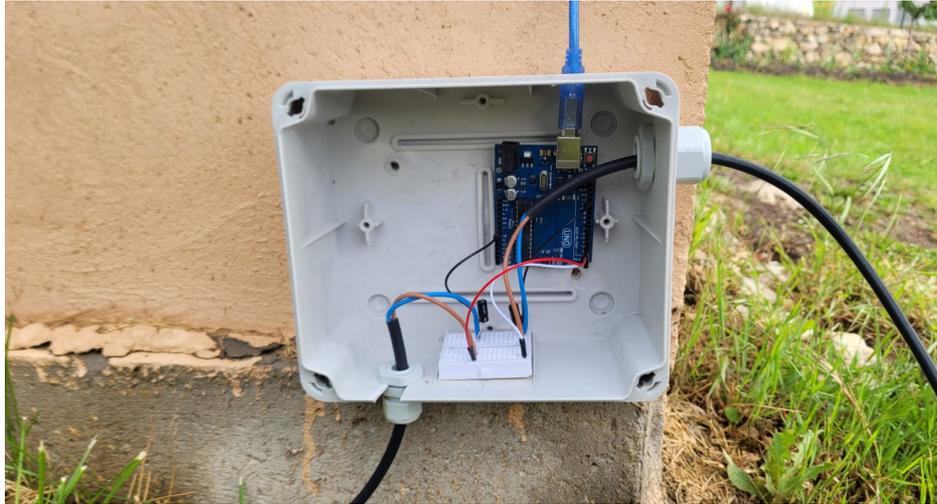
*Figura 63. Acabado final del sensor del depósito*



*Figura 64. Montaje del sensor en uno de los bidones*



*Figura 65. Acabo final del sensor del bidón*



*Figura 66. Conexión de ambos sensores con Arduino*



*Figura 67. Acabado final del prototipo*

## Capítulo 8. Conclusiones

La idea de este trabajo surge de unas necesidades reales existentes en el huerto familiar que llevábamos tiempo deseando solucionar. Una vez finalizada la carrera he considerado que este trabajo era una muy buena oportunidad para dedicarle tiempo a este proyecto.

El objetivo inicial del trabajo era realizar el estudio y construir un prototipo funcional, pero debido a la envergadura final del proyecto, la parte del prototipo ha quedado excluida de este. No obstante, este trabajo constituye la base de un proyecto mayor que mi familia y yo queremos realizar a corto y medio plazo en el huerto, con los bloques ya estudiados y añadiendo más elementos que aporten más funcionalidades para conseguir automatizar todavía más el control y el riego del huerto.

En este el trabajo hemos solucionado de forma teórica los dos temas principales que nos han llevado a plantearlo. Por un lado, con la selección de la bomba de agua del circuito inferior de acuerdo con los cálculos, incrementamos la presión para que todos los goteros funcionen por igual sin provocar que se sequen o pudran determinados alimentos. Por otro, con el estudio del módulo GSM se solventa el control a distancia de los tiempos de riego a distancia, sin tener que acudir físicamente al pueblo, además del resto de funcionalidades estudiadas a lo largo del trabajo.

Para terminar, en los Anexos (Capítulo 5) hemos realizado un cálculo orientativo del presupuesto total que conllevaría implantar todas las funcionalidades estudiadas en el huerto. El coste total aproximado sería de 618€. A primera vista parece un presupuesto elevado para un huerto familiar, pero hay que tener en cuenta diferentes aspectos.

- Como he comentado en la introducción, actualmente, utilizamos programadores de riego que cada temporada se hacen inservibles debido a las bajas presiones y a la cal. Por ello, cada año necesitamos comprar dos nuevos por un coste aproximado de 85€.
- Además, para realizar el traspaso de agua desde el depósito hasta los bidones utilizamos una bomba de gasolina, la cual a lo largo de la temporada consume cerca de 60€.
- Otro de los aspectos importantes, pero difícil de cuantificar, es la producción. La implantación de este proyecto aumentaría rendimientos al consumir menos agua y conseguir que cada alimento tenga su proporción necesaria de la misma, sin llegar a secarse ni pudrirse. Este hecho provoca que cada año invirtamos cerca de 60€ en replantes debido a los problemas del riego.

Por tanto, cada año invertimos cerca de 205€ para la temporada debido al sistema ineficiente actual. De esta forma, la inversión del proyecto estaría amortizada en un periodo de tres años, esperando que fuese duradera en el tiempo. Este es un argumento que evidencia claramente la viabilidad económica del estudio.

## Capítulo 8. Bibliografía

- [1] Realpars, “What is a level sensor”, 2019. [En línea]. Disponible en: <https://realpars.com/level-sensor/> . [Accedido: 15-octubre-2020]
- [2] SST Sensing Ltd Company, “7 Main Types of Level Sensing Methods - How do they differ?” [En línea]. Disponible en: <https://sstsensing.com/7-main-types-of-level-sensors/> . [Accedido: 15-octubre-2020]
- [3] Maher Smart Agrocontrollers, “Programador de riego con sensores de humedad de suelo”, [En línea]. Disponible en: <https://maherelectronica.com/programacion-de-riegos-sensores-humedad/>. [Accedido: 3-noviembre-2020]
- [4] Prismab, “Sensores de suelo para agricultura de precisión”, 2019. [En línea]. Disponible en: <https://prismab.com/sensores-de-suelo-para-agricultura-de-precision/> . [Accedido: 4-noviembre-2020]
- [5] El taller del huerto, “Sensor de humedad capacitivo vs resistivo”, 2020. [En línea]. Disponible en: <https://eltallerdelhuerto.com/sensor-capacitivo-vs-resistivo> . [Accedido: 4-noviembre-2020]
- [6] Seeedstudio, “DHT11 vs DHT22 – Which Temperature and Humidity Sensor Should You Use?”, 2020. [En línea]. Disponible en: <https://www.seeedstudio.com/blog/2020/04/20/dht11-vs-dht22-am2302-which-temperature-humidity-sensor-should-you-use/> . [Accedido: 13-noviembre-2020]
- [7] Climasmonterrey, “¿Cuáles son los Tipos de Válvulas de Solenoide?”, [En línea]. Disponible en: <https://www.climasmonterrey.com/cuales-son-los-tipos-de-valvulas-solenoides> . [Accedido: 21-noviembre-2020]
- [8] DirectIndustry, “Qué electroválvula elegir”, [En línea]. Disponible en: <http://guide.directindustry.com/es/que-electrovalvula-elegir/> . [Acceso: 21/11/2020]
- [9] Bermad, “Válvulas de solenoide de 3 vías tipo Latch”, [En línea]. Disponible en: <https://www.bermad.com/es/product/valvulas-de-solenoide-de-3-vias-tipo-latch/> . [Accedido: 23-noviembre-2020]

- [10] Viagua, “Electroválvulas para riego. El control del agua que necesitan los sistemas automáticos de riego”, [En línea]. Disponible en: <https://viagua.es/electrovalvulas-de-riego/> . [Accedido: 30-noviembre-2020]
- [11] Norberto Fueyo Díaz, Apuntes Mecánica de fluidos, Universidad de Zaragoza, 2019. [Accedido: 30-noviembre-2020]
- [12] Evo Industria, “Tipos de bombas de agua”, [En línea]. Disponible en: <https://evointustria.com/tipos-de-bombas-de-agua/> . [Acceso: 4-diciembre-2020]
- [13] Como Funciona, “Cómo funciona una bomba hidráulica”, [En línea]. Disponible en: <https://como-funciona.co/una-bomba-hidraulica>. [Acceso: 6-diciembre-2020]
- [14] Pompe Zanni, “Diferencia entre bombas centrífugas y volumétricas”, 2019. [En línea]. Disponible en <https://www.pomezanni.it/es/news-es/diferencia-entre-bombas-centrifugas-y-volumetricas/> . [Accedido: 8-diciembre-2020]
- [15] Ideas Mercado Libre, “Tipos de bombas de agua: cuál conviene ante cada necesidad”, 2019. [En línea]. Disponible en <https://ideas.mercadolibre.com/ar/home-deco/tipos-de-bombas-de-agua/> . [Accedido: 12-diciembre-2020]
- [16] Cindex, “4 razones por las que las bombas sumergibles son mejores que las bombas de chorro”, 2019. [En línea]. Disponible en <http://www.cindex.com.mx/bombas/4-razones-por-las-que-las-bombas-sumergibles-son-mejores-que-las-bombas-de-chorro/> .[Acceso: 14-dic-2020]
- [17] Luis Llamas, “Manejar cargas de más de 220v con Arduino y salida por relé”, 2016. [En línea]. Disponible en <https://www.luisllamas.es/Arduino-salida-rele/> .[Accedido: 2-enero-2021]
- [18] Luis Llamas, “Conmutar cargas con Arduino y relé de estado sólido”, 2016. [En línea]. Disponible en <https://www.luisllamas.es/Arduino-rele-estado-solido-ssr/> .[Accedido: 2-enero-2021]
- [19] Luis Llamas, “Controlar grandes cargas con Arduino y transistor mosfet”, 2016. [En línea]. Disponible en <https://www.luisllamas.es/Arduino-transistor-mosfet/> .[Accedido: 4-enero-2021]
- [20] Inventable “Como conectar un mosfet de potencia a un microcontrolador”, [En línea]. Disponible en <https://www.inventable.eu/como-conectar-un-mosfet-a-un-microcontrolador/> .[Accedido: 6-enero-2021]

- [21] Luis Llamas, “Medir el ángulo y sentido de giro con Arduino y encoder rotativo”, 2016. [En línea]. Disponible en <https://www.luisllamas.es/Arduino-encoder-rotativo/>. [Accedido: 21-enero-2021]
- [22] Rotary encoder module KY-040, Henry’s Bench. 2015. [En línea]. Disponible en <http://eeshop.unl.edu/pdf/KEYES%20Rotary%20encoder%20module%20KY-040.pdf>. [Accedido: 21-enero-2021]
- [23] Universidad Internacional de Valencia, “¿Qué es GSM y cómo funciona?”. 2018 [En línea]. Disponible en <https://www.universidadviu.com/es/actualidad/nuestros-expertos/que-es-gsm-y-como-funciona>. [Accedido 15-febrero -2021]
- [24] Sistemas, “Definición de GSM”. 2018 [En línea]. Disponible en <https://sistemas.com/gsm.php>. [Accedido: 15-febrero -2020]
- [25] Universidad Internacional de Valencia, “Sistemas GPRS, concepto general y usos principales”. 2017 [En línea]. Disponible en <https://www.universidadviu.com/es/actualidad/nuestros-expertos/sistemas-gprs-concepto-general-y-usos-principales>. [Accedido 20-febrero -2021]
- [26] Formación DKA, “Curso Energía Solar Fotovoltaica”, [En línea]. Disponible en <https://www.formacion-dka.es/cursos-gratis-online/course/view.php?id=561>. [Acceso: 20-marzo-2021]
- [27] Bonifacio Martín del Brío, Antonio Bono y José Luis Villaroel Salcedo, Apuntes Sistemas Electrónicos Programables, Universidad de Zaragoza, 2019-2020. [Accedido: 2-noviembre-2020]
- [28] Arduino, “attachInterrupt()”. [En línea]. Disponible en: <https://www.Arduino.cc/reference/en/language/functions/external-interrupts/attachinterrupt/>. [Accedido: 15-mayo-2021]
- [29] Control Automático Educación, “Timer Arduino” [En línea]. <https://controlautomaticoeducacion.com/Arduino/timer-Arduino/> [Acceso 20/5/2021]