



**Universidad de Valladolid**



**ESCUELA DE INGENIERÍAS  
INDUSTRIALES**

**UNIVERSIDAD DE VALLADOLID**

**ESCUELA DE INGENIERIAS INDUSTRIALES**

**Grado en Ingeniería Electrónica Industrial y Automática**

# **Sistema de riego automatizado para pequeñas instalaciones**

**Autora:**

**Zapatera Rodríguez, Paula**

**Tutor:**

**Arranz Gimón, Ángel Eugenio  
Departamento de Tecnología  
Electrónica**

**Valladolid, julio de 2022.**



## Resumen

Este trabajo recoge el desarrollo y la implementación de un sistema de riego automatizado destinado a pequeñas instalaciones basado en la tarjeta Arduino UNO R3.

Se ha diseñado un sistema que, mediante una serie de sensores y actuadores, es capaz de recopilar información de la humedad y la temperatura del entorno. A partir de esas lecturas, determina el estado del sistema completo y actúa en consecuencia para lograr un riego automatizado y eficiente.

Además, se ha construido una maqueta para simular el comportamiento de la instalación en una situación próxima a la realidad.

**Palabras clave:** Arduino, riego automatizado, Reloj de Tiempo Real (RTC), sensor de humedad, Entorno de Desarrollo Integrado (IDE)

## Abstract

This work includes the development and implementation of an automatic irrigation system for small installations based on the Arduino UNO R3 card.

A system has been designed that through a series of sensors and actuators is able to collect information on the humidity and temperature of the environment. From those readings, it determines the state of the entire system and acts accordingly to achieve automated and efficient irrigation.

In addition, a model has been built to simulate the behavior of the installation in a situation close to reality.

**Keywords:** Arduino, automated irrigation, Real Time Clock (RTC), humidity sensor, Integrated Development Environment (IDE)



## ÍNDICE

<b>INTRODUCCIÓN Y OBJETIVOS</b> .....	<b>1</b>
<b>1. JUSTIFICACIÓN DEL PROYECTO</b> .....	<b>1</b>
<b>2. OBJETIVOS PRINCIPALES</b> .....	<b>1</b>
<b>3. ESTRUCTURA DE LA MEMORIA</b> .....	<b>2</b>
<b>DESARROLLO DEL TFG</b> .....	<b>3</b>
<b>CAPÍTULO 1: MARCO TEÓRICO</b> .....	<b>3</b>
<b>1.1. DEFINICIÓN Y TIPOS DE RIEGO</b> .....	<b>3</b>
1.1.1 Riego por superficie .....	3
1.1.2 Riego presurizado .....	4
<b>1.2. CAMPO DE APLICACIÓN DEL PRESENTE PROYECTO</b> .....	<b>6</b>
<b>1.3. QUÉ ES ARDUINO Y CÓMO UTILIZARLO</b> .....	<b>6</b>
<b>1.4. PROTOCOLO DE COMUNICACIÓN I2C</b> .....	<b>9</b>
<b>CAPÍTULO 2: MÉTODO</b> .....	<b>11</b>
<b>2.1. PRIMEROS PASOS</b> .....	<b>11</b>
<b>2.2. COMPONENTES UTILIZADOS</b> .....	<b>12</b>
2.2.1. Tarjeta Arduino UNO R3 .....	13
2.2.2 Sensor de humedad y temperatura ambiente - DHT11 .....	17
2.2.3 Higrómetro FC28.....	19
2.2.4 Sensor capacitivo de humedad de suelo .....	20
2.2.5 Sensor de nivel de agua .....	22
2.2.6 Servomotor SG90.....	23
2.2.7 Reloj de tiempo real (Módulo RTC) – DS3231 .....	25
2.2.8 Relé 5V .....	26
2.2.9 Bomba de agua .....	29
2.2.10 Pantalla OLED.....	30
<b>2.3. SISTEMAS DE ALIMENTACIÓN</b> .....	<b>32</b>
2.3.1 Alimentación de la tarjeta Arduino .....	32
2.3.2 Alimentación externa .....	33
<b>2.4. SISTEMA FINAL</b> .....	<b>35</b>
<b>CAPÍTULO 3: PROGRAMACIÓN</b> .....	<b>38</b>
<b>3.1. LIBRERÍAS UTILIZADAS</b> .....	<b>39</b>
3.1.1 <Wire.h> .....	40
3.1.2 <OLED_I2C.h> .....	40
3.1.3 <DHT.h> .....	41
3.1.4 <Servo.h> .....	41
3.1.5 <RTClib.h> .....	41
3.1.6 <SimpleTimer.h>.....	42
<b>3.2. DESCRIPCIÓN DEL CÓDIGO</b> .....	<b>42</b>

3.2.1	Función bcdToDec()	43
3.2.2	Función setup()	43
3.2.3	Función <i>loop()</i>	44
3.2.4	Función readDS3231time()	48
3.2.5	Función <i>ok()</i>	48
3.2.6	Función <i>back()</i>	49
3.2.7	Función <i>down()</i>	49
3.2.8	Función <i>sleep()</i>	50
3.2.9	Función leerDHT()	51
3.2.10	Función sensorHum()	51
3.2.11	Función nivelAgua()	51
3.2.12	Función servoDep()	52
3.2.13	Función riegoManual()	52
3.2.14	Función lecturas()	53
3.2.15	Función RiegoAuto()	54
<b>3.3.</b>	<b>ESPACIO EN MEMORIA</b>	<b>55</b>
<b>CAPÍTULO 4:</b>	<b>DISEÑO</b>	<b>56</b>
4.1.	MAQUETA CONSTRUIDA	56
4.2.	MANUAL DE FUNCIONAMIENTO	60
4.3.	COSTE DEL EQUIPO	63
<b>CAPÍTULO 5:</b>	<b>OTROS CÁLCULOS</b>	<b>65</b>
5.1.	CONSUMOS DE LA INSTALACIÓN	65
5.2.	CÁLCULO DE RESISTENCIAS SERIE PARA LED	66
5.3.	SISTEMA DE ALIMENTACIÓN FOTOVOLTAICO	67
<b>CONCLUSIONES</b>		<b>70</b>
OBJETIVOS CUMPLIDOS		70
TRABAJO FUTURO		70
<b>BIBLIOGRAFÍA</b>		<b>72</b>
<b>ANEXOS</b>		<b>76</b>
<b>DATASHEETS</b>		<b>76</b>
Arduino UNO R3		77
Sensor DHT11		80
Higrómetro FC-28: Comparador LM393		84
Higrómetro capacitivo		91
Sensor de nivel de agua		92
Servomotor SG90		93
Módulo RTC		94
Relé 5V		100
Bomba de agua		102
Pantalla OLED		104

Módulo TP4056.....	107
Convertidor step-up.....	110
Diodos LED.....	116





## ÍNDICE DE FIGURAS

FIGURA 1. RIEGO POR SURCOS.....	4
FIGURA 2. SISTEMA DE RIEGO HIDROPÓNICO POR GOTEO. [3] .....	5
FIGURA 3. ENTORNO PRINCIPAL DE ARDUINO IDE. ....	7
FIGURA 4. ATAJOS ARDUINO IDE.....	8
FIGURA 5. ÁREA DE PROGRAMACIÓN ARDUINO IDE.....	8
FIGURA 6. BUS I2C. ....	9
FIGURA 7. TRANSFERENCIA DE DATOS I2C. [11] .....	10
FIGURA 8. DIRECCIONES DE TRANSFERENCIA BUS I2C. [11] .....	10
FIGURA 9. DIAGRAMA DE BLOQUES DEL PLANTEAMIENTO GENERAL DEL SISTEMA A DESARROLLAR.....	12
FIGURA 10. TARJETA ARDUINO UNO R3. ....	14
FIGURA 11. PINES DE ARDUINO UNO R3.[11] .....	17
FIGURA 12. PINES DEL SENSOR DHT11.....	18
FIGURA 13. CONEXIÓN DEL SENSOR DHT11 CON ARDUINO. ....	18
FIGURA 14. SENSOR DE HUMEDAD RESISTIVO. ....	19
FIGURA 15. CONEXIÓN DEL SENSOR RESISTIVO DE HUMEDAD CON ARDUINO. ....	20
FIGURA 16. SENSOR CAPACITIVO. ....	20
FIGURA 17. ESQUEMÁTICO DEL SENSOR CAPACITIVO DE HUMEDAD. [16] .....	21
FIGURA 18. CONEXIÓN DEL SENSOR CAPACITIVO DE HUMEDAD CON ARDUINO.....	22
FIGURA 19. SENSOR DE NIVEL DE AGUA. ....	22
FIGURA 20. CONEXIÓN DEL SENSOR DE NIVEL DE AGUA CON ARDUINO.....	23
FIGURA 21. MICROSERVO SG90. ....	24
FIGURA 22. CICLO DE FUNCIONAMIENTO DEL SERVOMOTOR. ....	24
FIGURA 23. SERVOMOTOR CONECTADO CON ARDUINO. ....	25
FIGURA 24. RELOJ EN TIEMPO REAL: MÓDULO RTC. ....	25
FIGURA 25. CONEXIÓN DEL MÓDULO RTC CON ARDUINO.....	26
FIGURA 26. A LA IZQUIERDA, RELÉ EN REPOSO. A LA DERECHA, BOBINA ENERGIZADA. [21].....	27
FIGURA 27. PARTES DEL MÓDULO RELÉ SIMPLE.....	27
FIGURA 28. ESQUEMA INTERNO DEL MÓDULO RELÉ. ....	28
FIGURA 29. CONEXIÓN DEL RELÉ CON ARDUINO.....	28
FIGURA 30. MINI-BOMBA DE AGUA SUMERGIBLE.....	29
FIGURA 31. CONSTRUCCIÓN DE LA BOMBA SUMERGIBLE. ....	29
FIGURA 32. PANTALLA OLED 128X64. ....	30
FIGURA 33. BOTÓN PULSADOR. ....	31
FIGURA 34. ESQUEMA DE CONEXIÓN DE LA PANTALLA OLED Y DE LOS PULSADORES. ....	31
FIGURA 35. CONFIGURACIÓN DE RESISTENCIAS PARA PULSADORES.....	32
FIGURA 36. ADAPTADOR DE CORRIENTE PARA ARDUINO. ....	33
FIGURA 37. MÓDULO TP4056. ....	34
FIGURA 38. STEP-UP DC-DC BOOST CONVERTER.....	34
FIGURA 39. CIRCUITO DE ALIMENTACIÓN EXTERNO PARA LA BOMBA DE AGUA. [26].....	35
FIGURA 40. DIAGRAMA DE FLUJO DEL FUNCIONAMIENTO DEL SISTEMA.....	37
FIGURA 41. DIAGRAMA DE BLOQUES DE LA INSTALACIÓN COMPLETA. ....	39
FIGURA 42. DECLARACIÓN DE LIBRERÍAS EN LA CABECERA DEL CÓDIGO. ....	39
FIGURA 43. DECLARACIÓN DE VARIABLES GLOBALES.....	43

FIGURA 44. FUNCIÓN BCDTODEC.....	43
FIGURA 45. FUNCIÓN SETUP().....	44
FIGURA 46. FUNCIÓN LOOP(): LLAMADA A FUNCIONES.....	44
FIGURA 47. FUNCIÓN LOOP(): CONFIGURACIÓN DE PÁGINA 0. ....	45
FIGURA 48. FUNCIÓN PÁGINA0(). ....	45
FIGURA 49. FUNCIÓN LOOP(): CONFIGURACIÓN DE PÁGINA 1. ....	46
FIGURA 50. FUNCIÓN LOOP(): CONFIGURACIÓN DE PÁGINA 2. ....	46
FIGURA 51. FUNCIÓN LOOP(): CONFIGURACIÓN DE PÁGINA 3. ....	47
FIGURA 52. FUNCIÓN LOOP(): CONFIGURACIÓN DE PÁGINA 4. ....	47
FIGURA 53. FUNCIÓN PAGINA4(). ....	48
FIGURA 54. FUNCIÓN READD3231TIME(). [34] .....	48
FIGURA 55. FUNCIÓN OK(). ....	49
FIGURA 56. FUNCIÓN BACK(). ....	49
FIGURA 57. FUNCIÓN DOWN(). ....	50
FIGURA 58. FUNCIÓN SLEEP(). ....	50
FIGURA 59. FUNCIÓN SENSORDHT(). ....	51
FIGURA 60. FUNCIÓN SENSORHUM(). ....	51
FIGURA 61. FUNCIÓN NIVELAGUA(). ....	52
FIGURA 62. FUNCIÓN SERVODEP(). ....	52
FIGURA 63. FUNCIÓN RIEGOMANUAL(). ....	53
FIGURA 64. FUNCIÓN LECTURAS(). ....	53
FIGURA 65. FUNCIÓN RIEGOAUTO(). ....	55
FIGURA 66. ESPACIO EN MEMORIA. [13] .....	55
FIGURA 67. MAQUETA CONSTRUIDA. ....	56
FIGURA 68. PARTE FRONTAL (IZQUIERDA) Y SENSOR DHT11 (DERECHA).....	57
FIGURA 69. PULSADORES DE LA INTERFAZ. ....	57
FIGURA 70. SERVOMOTOR. ....	57
FIGURA 71. INSTALACIÓN DE CONTROL.....	58
FIGURA 72. PANELES SOLARES. ....	58
FIGURA 73. DEPÓSITO DE AGUA CON BOMBA Y SENSOR DE NIVEL. ....	59
FIGURA 74. CONEXIONES DE TUBERÍA Y GOTEROS.....	59
FIGURA 75. SISTEMA DE RIEGO POR GOTEO. TUBERÍA, ENLACE DE 4 VÍAS Y ASPERSOR DE RIEGO POR GOTEO.....	60
FIGURA 76. CIRCUITO ELÉCTRICO DIODO LED.....	66
FIGURA 77. CONFIGURACIÓN SERIE/PARALELO DE LAS PLACAS SOLARES. [36] .....	68

## ÍNDICE DE TABLAS

TABLA 1. COMPARACIÓN DE SISTEMAS DE RIEGO.[4] .....	5
TABLA 2. LISTA DE COMPONENTES Y CONEXIÓN DE PINES. ....	13
TABLA 3. PINES ARDUINO UNO R3. ....	15
TABLA 4. CARACTERÍSTICAS SENSOR DHT11. ....	18
TABLA 5. PINES DE LA PANTALLA OLED. ....	31
TABLA 6. CONDICIONES DE LA INSTALACIÓN. ....	36
TABLA 7. DECLARACIÓN DE PINES EN EL CÓDIGO DE ARDUINO. ....	42
TABLA 8. ESTADO DE ACTIVACIÓN DE RELÉ Y LED. ....	52
TABLA 9. MEMORIA ARDUINO UNO R3. ....	55
TABLA 10. INTERFAZ DE LA PANTALLA OLED. [35] .....	62
TABLA 11. COSTE DEL EQUIPO. ....	64
TABLA 12. CONSUMO DE CADA DISPOSITIVO EN MA. ....	65
TABLA 13. CARACTERÍSTICAS DIODOS LED. ....	66
TABLA 14. CARACTERÍSTICAS DE UN PANEL SOLAR. ....	67
TABLA 15. TIEMPOS DE CARGA PARA BATERÍAS CON DISTINTAS CAPACIDADES. ....	69



## GLOSARIO DE TÉRMINOS Y ABREVIATURAS

**Microcontrolador:** circuito integrado programable que es capaz de ejecutar las órdenes grabadas en su memoria. Sus tres partes funcionales son la unidad de procesamiento, la memoria y los periféricos de entrada/salida.

**SDA:** *System Data* (línea de datos).

**SCL/SCK:** *System Clock* (señal de reloj).

**RS-232:** *Recommended Standard 232*. Interfaz que designa una norma para el intercambio de datos entre un DTE (Equipo Terminal de Datos) y un DCE (Equipo de Comunicación de Datos).

**PWM:** *Pulse Width Modulation* (modulación por ancho de pulso).

**SPI:** protocolo de comunicaciones que utiliza cuatro señales para la comunicación entre un maestro y un esclavo: SS, SCK, MOSI y MISO.

**SS:** *Slave Select* (selección de esclavo).

**MOSI:** *Master Out Slave In* (salida maestro, entrada esclavo).

**MISO:** *Master In Slave Out* (entrada maestro, salida esclavo).

**Sketch:** programa/lista de instrucciones de Arduino.

**OLED:** *Organic Light-Emitting Diode* (diodo orgánico de emisión de luz).

**RTC:** *Real Time Clock* (reloj de tiempo real).

**TX:** *Data Out to Device*. Transmite datos mediante protocolo serial.

**RX:** *Data In from Device*. Recibe datos mediante protocolo serial.

**V:** Voltaje en voltios (V).

**I:** Corriente en amperios (A).

**P:** Potencia en vatios (W).



## INTRODUCCIÓN Y OBJETIVOS

### 1. JUSTIFICACIÓN DEL PROYECTO

La realización del presente Trabajo de Fin de Grado se ha realizado con el objetivo de diseñar y construir un sistema de riego automatizado que trabaje de forma simple y eficiente.

Para ello, se han tenido en cuenta los diferentes factores que pueden afectar a pequeños jardines o cultivos y se ha implementado un sistema que facilite el cuidado y la conservación de estos.

Además, se ha querido construir un sistema que fuese económico y fácil de utilizar, con materiales asequibles y con una programación sencilla, que pueda ser implementado por cualquier persona que posea conocimientos básicos sobre programación y electrónica y con un ajustado presupuesto.

El sistema diseñado pretende mejorar los equipos domésticos comerciales actuales, consiguiendo detectar automáticamente temperatura y humedad. Además, se pretende reducir el consumo dotando de alimentación solar el sistema de llenado del depósito de agua que se utiliza para el riego.

### 2. OBJETIVOS PRINCIPALES

El principal objetivo de este proyecto es crear un sistema que vuelva independiente cualquier pequeña instalación botánica que se pueda tener en una vivienda. Su uso está destinado a usuarios aficionados a la jardinería que quieran utilizarlo en pequeños proyectos de ámbito doméstico.

Requiere de poco espacio para ser colocado y tiene una sencilla y flexible instalación, ya que puede modificarse a gusto y necesidad del usuario, cumpliendo perfectamente su función en instalaciones de tamaño reducido.

Además, debido a los sensores y actuadores empleados en su construcción, evita que se desperdicie una gran cantidad de agua y proporciona al usuario distintos datos sobre la instalación y su entorno que logran crear un sistema monitorizado, intuitivo y eficiente.

### 3. ESTRUCTURA DE LA MEMORIA

La presente memoria se ha organizado en diferentes capítulos en los que se irán tratando los aspectos más importantes y destacables del sistema implementado.

En primer lugar, en el *Capítulo 1: Marco teórico*, se realizará un breve análisis de los sistemas de riego y se describirán algunos de los tipos que más se utilizan en pequeñas y grandes instalaciones.

A continuación, en el *Capítulo 2: Método*, se explicará con detalle el sistema utilizado para la programación e implementación del sistema y se analizará cada uno de los elementos que lo componen, tales como sensores, actuadores, dispositivos de alimentación, etc. Además, se hará un recorrido sobre la forma de proceder y la metodología seguida para llevar a cabo este proyecto, analizando las soluciones propuestas para enfrentar dicho trabajo.

En el *Capítulo 3: Programación*, se analizará con detalle el código utilizado en la programación del sistema, haciendo hincapié en las librerías, variables y funciones utilizadas para la monitorización de los datos recogidos por los sensores.

En el *Capítulo 4: Diseño*, se va a describir detalladamente la maqueta construida para la visualización del funcionamiento del programa, añadiendo un pequeño manual de usuario y una tabla que recogerá el coste del equipo empleado para la construcción de esta.

Por otro lado, en el *Capítulo 5: Otros cálculos*, quedarán reflejados los consumos y requerimientos específicos de determinadas partes del sistema.

Por último, se analizarán los objetivos cumplidos y se realizará una propuesta de trabajo futuro para mejorar y optimizar la instalación actual y, para complementar las explicaciones, se han añadido tablas, imágenes y ecuaciones, además de incluir unos *Anexos* con más documentación sobre cada dispositivo.



## DESARROLLO DEL TFG

### CAPÍTULO 1: MARCO TEÓRICO

#### 1.1. DEFINICIÓN Y TIPOS DE RIEGO

**Riego:** “Acción y efecto de regar o esparcir agua al suelo o a plantas cultivadas. Mediante diversas técnicas, se aplica agua a canteros o directamente sobre las plantas, para asegurar la producción de cultivos cuyas exigencias hídricas no son satisfechas por la aportación de procedencia natural.”[1]

Existen distintos tipos de riego y lo más importante a la hora de elegir cuál utilizar es tener en cuenta que el sistema de riego debe ser lo más eficiente posible.

Se pueden clasificar los métodos de riego principalmente en dos categorías: riego por superficie y riego presurizado. [2]

##### 1.1.1 Riego por superficie

La principal característica de este tipo de riegos es que sus costes de energía y de inversión son bastante bajos. No es el sistema más eficiente, ya que se producen muchas pérdidas de agua por infiltración. Se utiliza en cultivos arbóreos y de hortalizas y se basa en un sistema de riego por inundación, cubriendo toda la superficie con agua, lo que hace que el consumo sea muy elevado.

- Riego tendido: es uno de los sistemas de riego más antiguos que existen, también el más ineficiente. La distribución del agua no se realiza igual por todas partes, lo que hace que algunas plantas o cultivos tengan un exceso de agua y otras cuya cantidad de esta sea nula.
- Riego por surcos o melgas: el agua avanza por una especie de canales escarados en la tierra desde un sitio alto a otro más bajo.

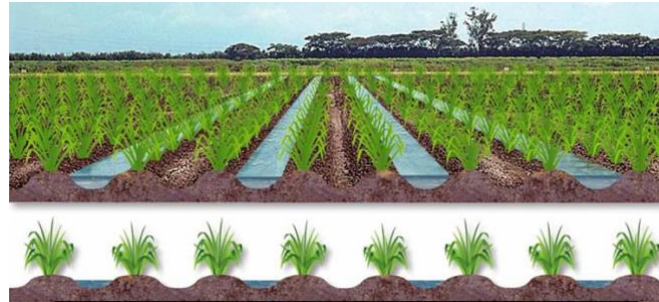


Figura 1. Riego por surcos.

### 1.1.2 Riego presurizado

En este tipo de riego el agua es conducida por tuberías a una cierta presión, llegando directamente a la zona de regadío. Con este método se evitan las pérdidas por infiltración logrando aprovechar mayor cantidad de agua.

Un sistema de riego presurizado está formado por un emisor o emisores a los que le llega el agua gracias a un sistema formado básicamente por una fuente de presurización (una bomba o la propia pendiente del terreno) y unas tuberías de conducción.

- Riego por aspersión: permite aplicar el agua en forma de lluvia sobre las plantas. Debe utilizarse en zonas en las que la fuerza del viento no sea significativa, ya que se producirían muchas pérdidas. Además, utiliza mucha energía eléctrica al necesitar una elevada presión de funcionamiento. Sin embargo, tiene una gran cantidad de ventajas ya que es uno de los sistemas más eficientes, cubre grandes áreas de terreno y se puede automatizar.
- Riego por microaspersión: funciona exactamente igual que el sistema anterior, pero necesita una presión mucho más baja ya que las gotas de agua que pulveriza son mucho más finas.
- Riego por goteo o localizado: consiste en la aplicación continua de agua en forma de gotas en un lugar próximo a la planta, con lo que se consigue formar un bulbo húmedo bajo cada gotero que permite el mejor desarrollo de las raíces de esta. Este sistema necesita aún menos presión que los anteriores, ya que el agua lleva presión dentro de la tubería pero al salir por los goteros la pierde. Sus principales inconvenientes están en el mantenimiento de los emisores para que no se tapone la salida del agua, además de necesitar una fuente constante de agua. Sin embargo, es un sistema muy eficiente que se adapta a numerosos tipos de terreno.

- Riego hidropónico: es el sistema más eficiente, pero también el más complejo, ya que requiere introducir sustancias químicas y nutritivas en el agua destinada a los cultivos. [3]

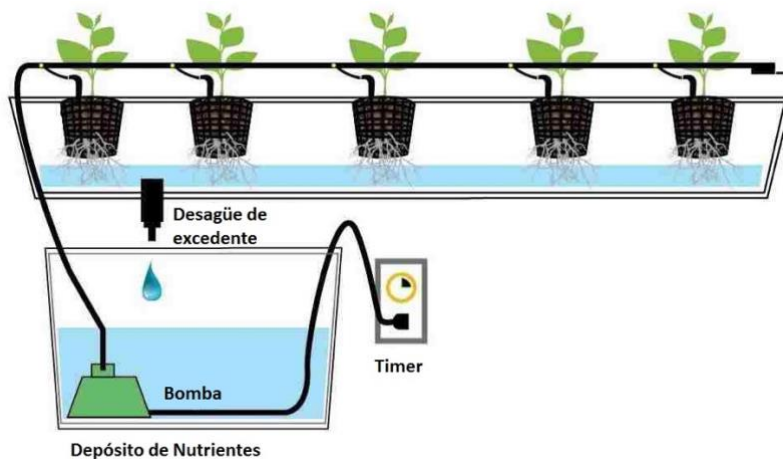


Figura 2. Sistema de riego hidropónico por goteo. [3]

- Riego por nebulización: consiste en pulverizar agua para que, en contacto con el aire se evapore y refresque el ambiente. Se utiliza sobre todo en viveros e invernaderos para minimizar el exceso de calor.

A continuación, en la Tabla 1, se realiza una comparación entre los sistemas de riego más utilizados.

TIPO	EFICIENCIA	VENTAJAS	DESVENTAJAS
Riego por gravedad	55%	Económico Requerimientos energéticos nulos. El viento no es un factor limitante.	No es conveniente usarlo en terrenos con desnivel. Propenso a la aparición de maleza y de enfermedades.
Riego por aspersión	85%	Alta eficiencia. Cubre grandes superficies de terreno. Permite automatizar el riego. Larga vida útil.	Alta inversión y mano de obra. Instalación más compleja. Podría incrementar la aparición de maleza.
Riego por goteo	95%	Crecimiento adecuado de las raíces. Mantiene la humedad constante. Permite la aplicación de fertilizantes en el agua. Riego específico. Disminuye el gasto de agua. Pueden realizarse otras labores de cultivo a la vez. Se adapta a terrenos rocosos o con pendientes.	Puede taparse si no se filtra el agua. Alta inversión inicial. Costos de instalación y diseño elevados.

Tabla 1. Comparación de sistemas de riego.[4]

## 1.2. CAMPO DE APLICACIÓN DEL PRESENTE PROYECTO

El sistema implementado está pensado para utilizarse en pequeñas instalaciones, tales como macetas o pequeños jardines. El principal objetivo es reducir y controlar el gasto de agua que va a tener la instalación, por lo que se ha optado por utilizar el riego por goteo.

Este sistema, como se indicaba anteriormente, es de los más sostenibles y eficientes. Además, resulta tener una fácil implementación ya que se puede encontrar una gran cantidad de materiales para realizar instalaciones económicas y eficientes al mismo tiempo.

Al hidratar directamente las raíces de las plantas, que es donde más agua se necesita, se reducen los costes y el estancamiento del agua en el sistema. Además, al no mojarse ni las hojas ni las flores, la planta no se marchita tan rápido y tiene un mayor tiempo de vida.

El nivel de humedad que debe tener un cultivo dependerá de la naturaleza de este y de la fase de crecimiento en la que se encuentre. Por ejemplo, para una plantación interior se recomienda un nivel de humedad entre el 60% y el 80%, reduciéndose a medida que crece la planta. [5]

## 1.3. QUÉ ES ARDUINO Y CÓMO UTILIZARLO

Arduino es una plataforma de desarrollo que se basa en un microcontrolador reprogramable y que funciona mediante hardware y software libre fáciles de usar. Además dispone de un software multiplataforma que funciona tanto en Windows, como en Linux y MacOS.

Puede establecer conexiones entre el microcontrolador y diferentes sensores y actuadores gracias a los pines hembra incorporados en cada tarjeta, que corresponden a los diferentes pines de conexión del microcontrolador.

Su programación se realiza usando el “*Arduino Programming Language*” y el “*Arduino Development Environment*”.

Al ser una plataforma de código abierto, son incontables la cantidad de recursos que se pueden encontrar sobre esta herramienta. Además, se ha forjado una gran comunidad de programadores y creadores que utilizan este software para realizar todo tipo de proyectos, desde las ideas más sencillas (como encender un LED) hasta las más complejas (por ejemplo, domotizar la instalación de una vivienda).

Arduino nació en el Instituto de diseño Interactivo de Ivrea, en Italia, en el año 2005. Surgió de la necesidad de disponer en las aulas de un dispositivo de bajo coste que permitiera la realización y el desarrollo de diferentes proyectos electrónicos.

Como se ha mencionado anteriormente, Arduino es un sistema libre y extensible y por eso se utiliza tanto. Cualquiera puede ampliar y desarrollar su entorno de trabajo y esto ha desencadenado en el surgimiento de numerosas placas electrónicas no oficiales y el desarrollo de librerías que se adaptan de mejor forma a las necesidades y posibilidades de cada usuario.

Su lenguaje de programación está basado en C++, por lo que es de fácil comprensión y permite el desarrollo de proyectos tanto a nuevos programadores como a otros más experimentados.

Además, es un recurso de bajo costo con una gran versatilidad y con infinitas posibilidades de reutilización. [6][7][8]

En el [Capítulo 2.2: Componentes utilizados](#) se explica detalladamente la tarjeta Arduino UNO R3, que es la elegida para el desarrollo de este proyecto.

En cuanto al entorno de trabajo en el que se va a realizar la programación del Arduino elegido, se tiene la plataforma ARDUINO Genuino (Arduino IDE). Al iniciar dicho programa, aparecerá la interfaz principal donde poder comenzar a desarrollar el código del proyecto (Figura 3).



Figura 3. Entorno principal de Arduino IDE.

Como se puede ver en la Figura 4, se dispone de varios atajos que simplifican la comprobación del código que está siendo desarrollado. La función de cada uno de ellos, de izquierda a derecha, es la siguiente:

- *Verificar*: comprueba que no hay errores de sintaxis en el código y que no existe ninguna incoherencia.
- *Subir*: mediante este botón, se carga el código a la placa Arduino.
- *Nuevo*: genera un nuevo sketch de Arduino, totalmente en blanco.
- *Abrir*: permite cargar en el programa un código ya existente guardado con anterioridad.
- *Salvar*: permite guardar sin necesidad de comprobar antes el código, tal y como hace la opción *Verificar*.
- *Monitor Serie*: permite visualizar datos, valores de parámetros, etc. al usuario.



Figura 4. Atajos Arduino IDE.

Por otro lado, se tiene el área de programación y desarrollo del código (Figura 5). Este se divide principalmente en dos funciones: *setup* y *loop*. La primera de ellas se ejecuta una sola vez durante todo el proceso mientras que la segunda se repite en bucle. Antes de estas funciones, en la cabecera principal del programa, se deben declarar las variables globales, las constantes, los objetos y las librerías necesarias para el proyecto en desarrollo.

```
void setup() {  
  // put your setup code here, to run once:  
}  
  
void loop() {  
  // put your main code here, to run repeatedly:  
}
```

Figura 5. Área de programación Arduino IDE.

Por último, se tiene el área de mensajes y la consola (zona inferior en verde y negro). En esta zona aparecerá si el código está siendo verificado, si se ha compilado o si se ha subido a la tarjeta Arduino, además de mostrar todos los posibles errores que podría tener el código desarrollado. También muestra la línea en la que se encuentra situado el cursor, el tipo de placa que se está usando y el puerto de comunicación al que está conectada.

## 1.4. PROTOCOLO DE COMUNICACIÓN I2C

En este proyecto, se va a utilizar una pantalla OLED y un módulo de reloj en tiempo real (RTC). Estos dos dispositivos utilizan la comunicación I2C (*Inter-Integrated Circuit*) o TWI (*Two Wire Interface*) para establecer conexión con la tarjeta Arduino y poder intercambiar datos.

I2C es un bus de comunicaciones en serie que fue diseñado por Phillips en 1982 con el fin de facilitar la comunicación interna entre circuitos integrados.

Se trata de un bus maestro – esclavo que requiere únicamente dos conexiones: la señal de reloj (CLK/SCL) y la señal de datos (SDA). [9]

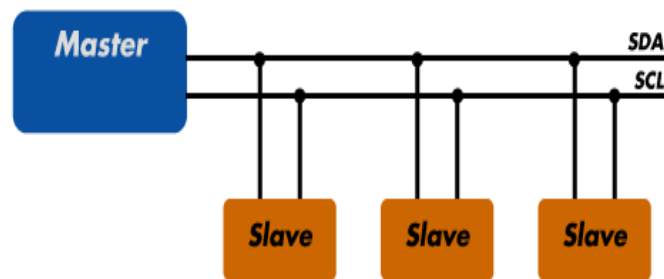


Figura 6. Bus I2C.

Cada dispositivo debe tener una dirección única dentro de este bus, lo que permitirá acceder a cada uno de ellos de forma individual. Esta dirección puede ser fijada tanto por hardware como por software.

El dispositivo maestro establece la comunicación con los esclavos, a quienes puede enviar o de los que puede recibir datos. Estos esclavos no pueden ni iniciar la comunicación ni comunicarse entre ellos.

El bus I2C es un bus síncrono que mantiene sincronizados a todos los dispositivos conectados en él. Así, todos disponen del mismo reloj y de la misma velocidad de transmisión.

El proceso de comunicación en el bus sigue una serie de pasos [10]:

1. El maestro empieza la comunicación enviando una *start condition*, con lo que pone en espera a los esclavos.
2. El maestro se dirige al esclavo con el que quiere establecer la comunicación enviando los 7 bits que componen la dirección de este último. El octavo bit, el de menor peso, servirá para indicar si quiere enviar (escritura = 0) o recibir (lectura = 1) información.

3. Cada esclavo, compara la dirección enviada por el maestro con la suya propia para ver si estas coinciden.
4. Cada byte leído/escrito por el maestro debe ser validado por un bit de ACK (*Acknowledged* - bit de confirmación de recepción de datos).
5. Cuando la comunicación ha finalizado, el maestro envía una *stop condition* para dejar libre el bus.

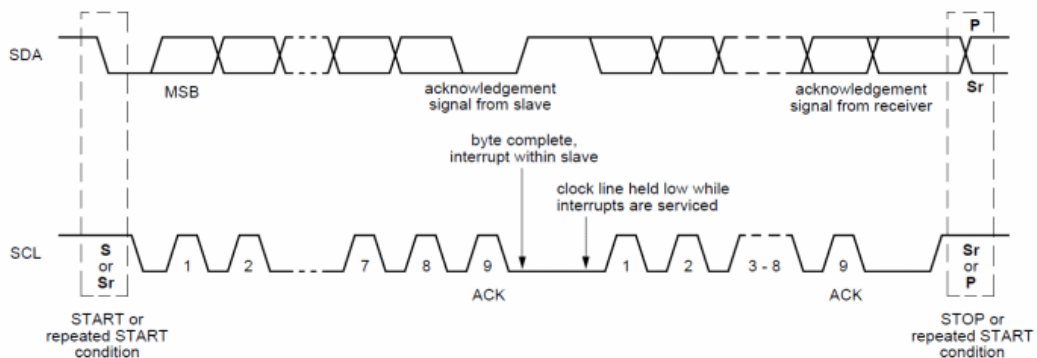


Figura 7. Transferencia de datos I2C. [11]

Cuando los datos son enviados por SDA, SCL envía los pulsos de reloj que mantienen sincronizados al maestro y al esclavo. Los datos se envían como un bit en cada pulso de reloj, lo que supone una frecuencia de transmisión de 100kHz.

Con esos 7 bits de dirección es posible acceder a 112 dispositivos diferentes en un mismo bus, ya que 16 de las 128 direcciones disponibles se reservan para usos especiales.

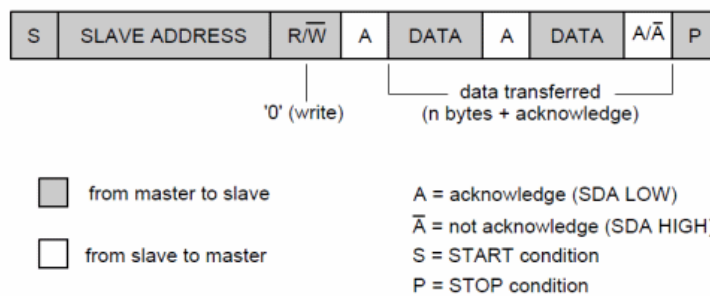


Figura 8. Direcciones de transferencia bus I2C. [11]

En el caso de Arduino, el soporte I2C viene configurado en determinados pines, que variarán en función del modelo de tarjeta Arduino empleado. En este caso se va a utilizar la tarjeta Arduino UNO R3 cuyo pin A4 está asociado a SDA y el pin A5 a SCK.

Para poder utilizar el bus I2C en Arduino, será imprescindible incluir la librería *Wire.h*, ya que contiene las funciones necesarias para controlar el hardware integrado.



## CAPÍTULO 2: MÉTODO

### 2.1. PRIMEROS PASOS

La idea principal de este proyecto, era realizarlo de tal forma que quedara un código sencillo y compacto y un montaje que se pudiera aplicar a cualquier pequeño proyecto de jardinería, desde una maceta hasta un jardín pequeño en una vivienda unifamiliar.

A raíz de este pensamiento, empezaron a surgir las preguntas sobre qué sensores y actuadores sería más conveniente utilizar, cuál sería la mejor forma de alimentar las distintas partes del circuito final o cómo implementar un código de Arduino que fuera claro y conciso.

Con esto presente, se empezó a desarrollar el proyecto poco a poco, buscando información y haciendo pruebas en cada nuevo paso que se daba.

En primer lugar se adquirió la tarjeta Arduino UNO R3, ya que iba a ser el componente fundamental e indispensable en este trabajo. A continuación, se empezó a desarrollar el código destinado a la programación del riego y, en función de las necesidades que iban surgiendo en el proyecto, se fueron consiguiendo los distintos sensores y actuadores.

Se observó que el funcionamiento de la instalación debía basarse en la lectura de algún parámetro que determinara si era necesario o no activar el riego, además de programar el mismo a unas horas determinadas de forma que se tuviera control sobre él.

De este modo, surgió la necesidad de determinar la humedad de la tierra utilizada y se buscaron una serie de sensores que podían obtener dicho parámetro. Además, para poder programar las horas de riego se buscó un módulo reloj que permitiera desarrollar un código para ese fin.

Con la base de funcionamiento del sistema clara, se fueron buscando otros componentes que completaran la instalación. De esta forma, se buscó la forma de monitorizar los valores para que el usuario tuviese conocimiento de estos mediante un display con pulsadores, añadiendo además información sobre la cantidad de agua disponible para regar, señalización luminosa que avisa sobre el estado del sistema o un pequeño servomotor que representa la apertura y el cierre de la electroválvula que llenaría el depósito en un caso real.

En la Figura 9, se puede ver un diagrama de bloques que representa el planteamiento general del sistema que se irá desarrollando a lo largo de esta memoria.

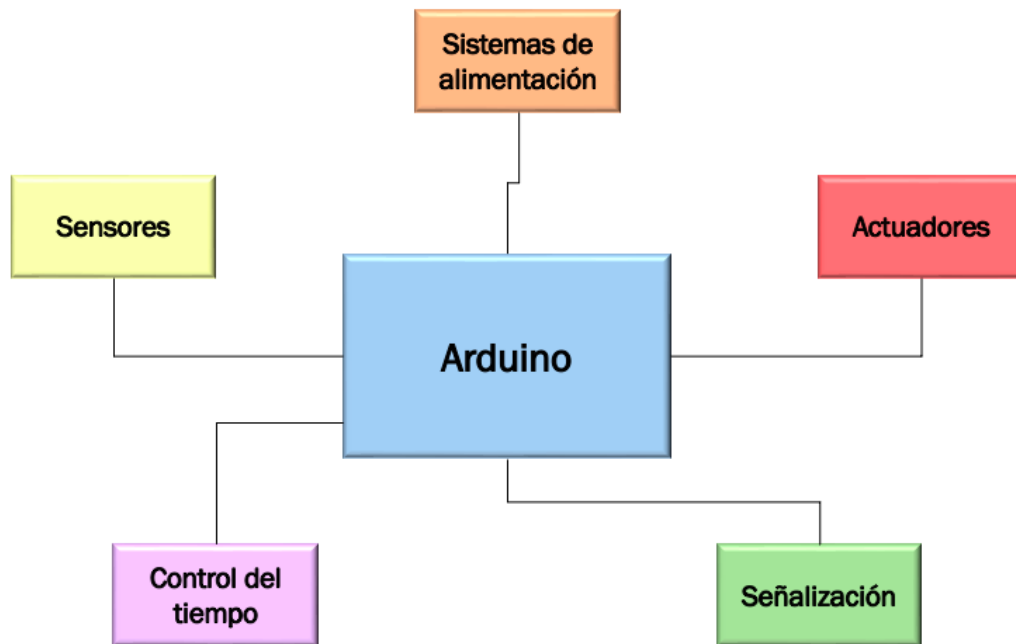


Figura 9. Diagrama de bloques del planteamiento general del sistema a desarrollar.

Antes de incorporar cualquiera de los componentes, se fueron probando individualmente mediante sencillos programas implementados en la tarjeta Arduino y, una vez probados todos y cada uno de ellos, se fueron programando en conjunto hasta conseguir que todos trabajaran en el mismo entorno, consiguiendo así la programación deseada.

Por último, se diseñó y construyó una pequeña maqueta para simular el funcionamiento y comprobar que realmente la idea estaba bien desarrollada.

En los siguientes puntos del proyecto, se describe con detalle todos los componentes y en el [Capítulo 3: Programación](#) se analiza la función de cada uno de ellos.

## 2.2. COMPONENTES UTILIZADOS

Los componentes utilizados aparecen en la Tabla 2. Además, se indica la conexión de cada uno de sus pines y en los [Anexos](#) se adjuntan también sus hojas de características.

En los siguientes puntos se describe en detalle cada componente de forma individual.

	PINES	CONEXIONES
Pantalla OLED	GND	GND
	CDD	3.3V
	SCK	SCL
	SDA	SDA
Sensor DHT11	VDD	5V
	DATA	Pin digital 7
	NC	No conectado
Pulsadores	GND	GND
	DATA	Pines digitales 2, 3, 4 y 5
Reloj de tiempo real (módulo RTC)	VCC	5V
	GND	GND
	SDA	Pin analógico A4
	SCL	Pin analógico A5
Sensor de humedad resistivo	A0	Pin analógico A0
	DO	No conectado
	GND	GND
	VCC	5V
Sensor de humedad capacitivo	GND	GND
	VCC	5V
	AUOT	Pin analógico A2
Sensor de nivel de agua	S	Pin analógico A1
	+	5V
	-	GND
Servomotor	GND	GND
	VCC	5V
	SEÑAL	Pin digital 9
Diodo LED	+	GND
	-	Pin digital 6 (rojo), 10 (verde) y 11 (amarillo)
Relé 5V	VCC	5V
	GND	GND
	IN	Pin digital 8
	NO	Bomba de agua
	COM	Negativo de la alimentación externa de la bomba
	NC	No conectado

Tabla 2. Lista de componentes y conexión de pines.

### 2.2.1. Tarjeta Arduino UNO R3

Arduino UNO R3 es una placa de desarrollo de software basada en el microcontrolador ATmega328P [12]. Es una de las placas más populares de la firma y su fama se debe a:

- Su precio, ya que es una de las más económicas del mercado.
- Su facilidad de uso, haciendo muy sencilla la implementación de numerosos proyectos.
- Su compatibilidad con cualquier sistema operativo.
- La disponibilidad de complementos para esta placa es muy alta y se pueden añadir numerosos módulos que le añaden aún más funciones de las que ya posee.
- Su durabilidad y seguridad, lo que hace que se pueda utilizar sin miedo de estropearla porque, en el peor de los casos, se podría reemplazar el microcontrolador y seguir utilizándola.

En la Figura 10, se puede ver una imagen general de la placa y sus partes principales.

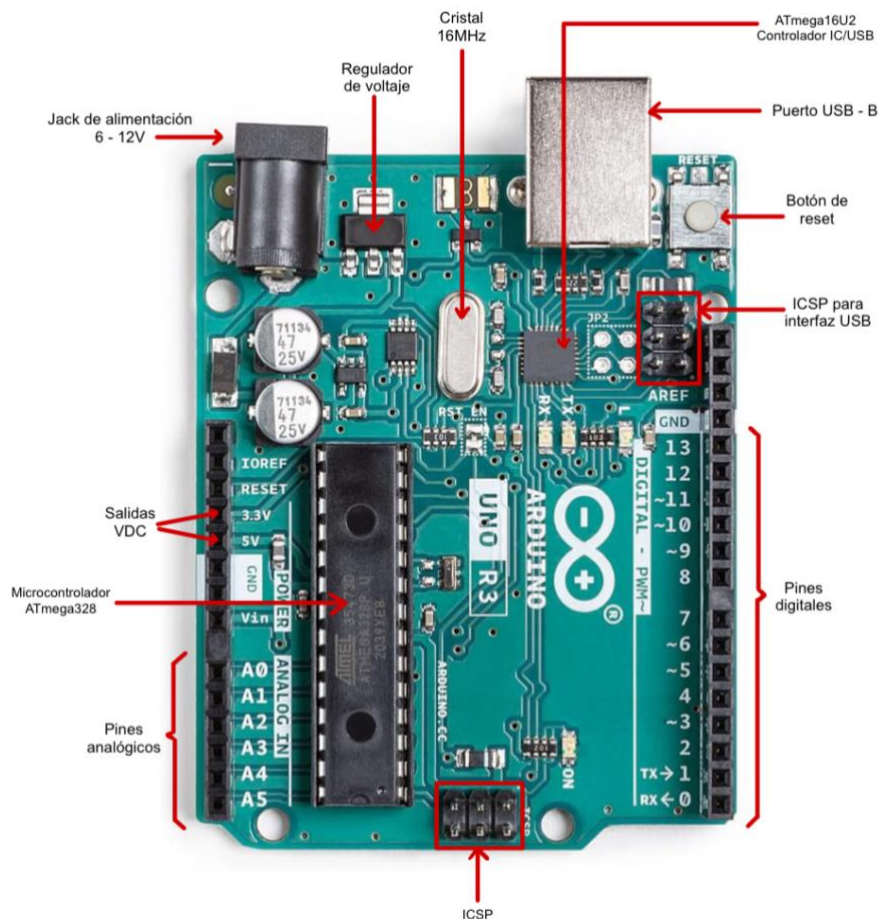


Figura 10. Tarjeta Arduino UNO R3.

Cuenta con 14 pines digitales de entrada/salida, de los cuales 6 disponen de señal PWM, y 6 entradas analógicas de 10 bits. Además, tiene 3 pines de GND y 2 de alimentación (3.3V y 5V). En la Tabla 3 se resume la función de cada uno de ellos.

PIN/LED	TIPO	DESCRIPCIÓN
NC	NC	No conectado.
IOREF	IOREF	Proporciona 5V como tensión de ref.
PC6	Reset	Reset.
+3V3	Power	Alimentación de 3.3V, 50mA.
+5V	Power	Alimentación de 5V, 40mA.
GND	Power	Tensión de tierra a 0V.
VIN	Power	Entrada de alimentación.
A0	Analog/GPIO	Entradas analógicas, encargadas de leer valores de tensión entre 0V y 5V, con una resolución de $1024 = 10$ bits.
A1	Analog/GPIO	
A2	Analog/GPIO	
A3	Analog/GPIO	
A4/SDA	Analog input/I2C	Entrada analógica/Señal de datos I2C.
A5/SCL	Analog input/I2C	Entrada analógica/Señal de reloj I2C.
D0/RX	Digital/GPIO	Pin digital para comunicación por puerto serie.
D1/TX	Digital/GPIO	
D2	Digital/GPIO	IN/OUT digital. Entrada de interrupciones.
~D3	Digital/GPIO	IN/OUT digital. PWM. Interrupciones.
D4	Digital/GPIO	IN/OUT digital.
~D5	Digital/GPIO	IN/OUT digital. Salida PWM.
~D6	Digital/GPIO	IN/OUT digital. Salida PWM.
D7	Digital/GPIO	IN/OUT digital.
D8	Digital/GPIO	IN/OUT digital.
~D9	Digital/GPIO	IN/OUT digital. Salida PWM.
~D10/SS	Digital	IN/OUT digital. Salida PWM. SS: <i>Slave Select</i> .
~D11/MOSI	Digital	IN/OUT digital. Salida PWM. MOSI: <i>Master Out Slave In</i> .
D12/MISO	Digital	IN/OUT digital. MISO: <i>Master In Slave Out</i> .
D13/SCK	Digital	IN/OUT digital. SCK: <i>Signal Clock</i> .
AREF	Digital	Referencia analógica.
SDA	Digital	Señal de datos I2C.
SCL	Digital	Señal de reloj I2C.
TX LED	LED	Parpadean cuando los datos se transmiten: USB – serie y USB – PC.
RX LED	LED	
POWER	LED	Placa alimentada.

Tabla 3. Pines Arduino UNO R3.

Su tensión de funcionamiento es de 5V. Este voltaje es el que le llega a la tarjeta si es alimentada mediante el puerto USB desde un ordenador, pero a través del Jack de alimentación puede aplicarse una tensión entre 6 y 20V,

siendo recomendable el rango existente entre 7 y 12V. Este voltaje se adapta a los 5V que necesita la placa mediante el regulador de tensión integrado en ella. En el presente proyecto, como se verá más adelante, se usará el puerto USB de un ordenador.

La corriente que circula por los pines tiene un límite de 40mA, siendo el valor máximo recomendado de 20mA. Si se utiliza el pin de alimentación de 3,3V, la corriente que circulará por él será de 50mA.

Esta placa de Arduino es la más robusta de la marca gracias a las mejoras que se fueron implementando en versiones anteriores de la misma. La evolución de las tarjetas de Arduino se divide en función de la forma de comunicación de estas con el ordenador. Anteriormente utilizaban una interfaz RS232 que ha sido sustituida por un puerto USB que simplifica el proceso de grabado y comunicación con el ordenador. Este puerto USB está gestionado por un microcontrolador ATmega16U2, que funciona como interfaz USB-Serie.

Para llevar a cabo esta comunicación serie, el puerto USB utiliza los pines digitales 1 (TX) y 0 (RX) para recibir y transmitir los datos necesarios, ya sea para comunicarse con el ordenador o con cualquier otro dispositivo que se haya conectado a estos dos pines.

Como se ha mencionado anteriormente, la placa Arduino UNO R3 está controlada por un microcontrolador ATmega328P, que es quien graba, almacena y ejecuta el código programado.

Este microcontrolador dota a la tarjeta Arduino de una memoria de almacenamiento que se divide de la siguiente forma [13]:

- 32 kB de memoria flash destinados a almacenar el sketch.
- 2 kB de memoria SRAM donde se almacenan las variables declaradas en el sketch.
- 1kB de memoria EEPROM que almacena datos que se quieren conservar ante posibles reinicios o fallos de alimentación.

Además, dispone de un cristal oscilador y de un resonador cerámico de 16MHz, utilizados por el ATmega16Us y el ATmega328P, respectivamente, para generar las señales de reloj. Su funcionamiento se basa en analizar las variaciones de voltaje, recogidas por el cristal y el resonador, para generar una onda cuadrada de su misma frecuencia. Esta onda determina la velocidad a la que el microcontrolador ejecuta las acciones programadas en él.

Por último, cabe destacar la existencia del botón de RESET. Este permite el reinicio del microcontrolador ATmega328P, haciendo que el código cargado se ejecute de nuevo desde el principio.

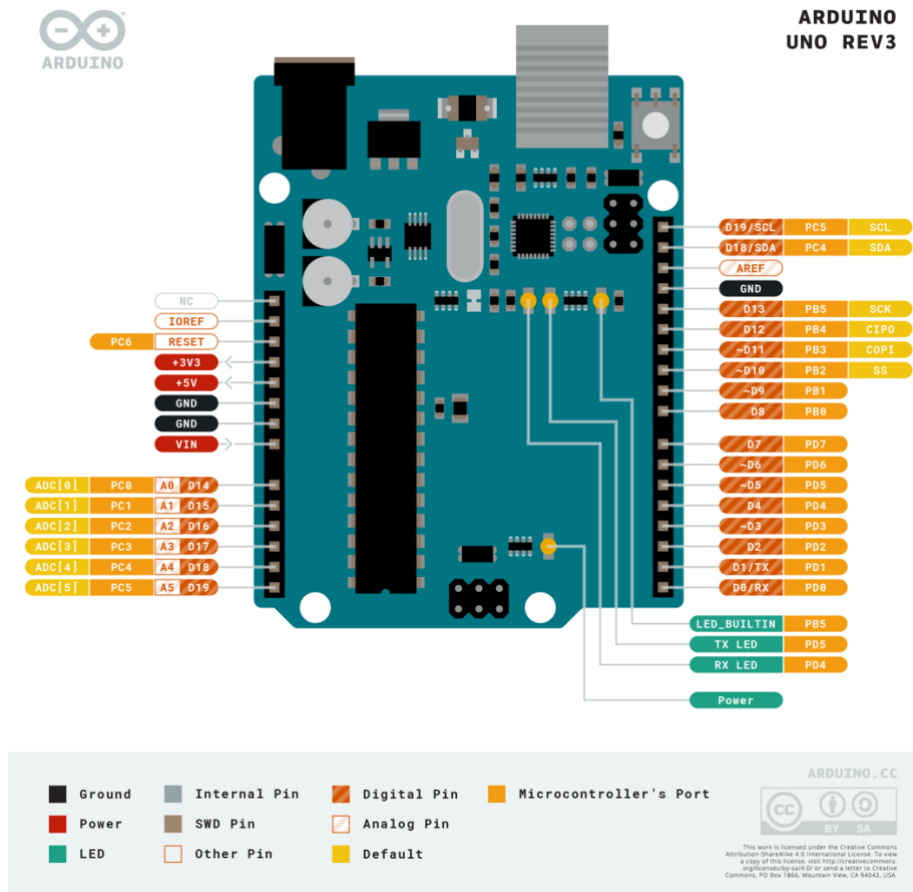


Figura 11. Pines de Arduino UNO R3.[11]

## 2.2.2 Sensor de humedad y temperatura ambiente - DHT11

El sensor DHT11 [14] va a proporcionar los datos de humedad y temperatura del ambiente mediante configuración digital. Este dispositivo tiene cuatro pines, de los que solo van a utilizarse tres: el de GND, el de alimentación y el de la señal digital que conectará con la tarjeta Arduino. El cuarto pin no está conectado.



Figura 12. Pines del sensor DHT11.

Este sensor puede encontrarse en el mercado en dos versiones distintas: una con PCB incorporada y otra sin ella. En este caso, se ha utilizado la versión sin PCB, por lo que será necesario introducir una resistencia pull-up (Figura 35) conectada a la salida digital. El valor de esta resistencia es de 5,1kΩs, valor recomendado por el fabricante.

El DHT11 destaca porque, a pesar de ser un sensor muy económico, tiene una alta fiabilidad y estabilidad. Sus principales características vienen reflejadas en la Tabla 4.

DHT11	
Rango de temperatura	de 0°C a 50°C ± 2°C
Rango de humedad	de 20% a 80% ± 5%
Tensión de funcionamiento	de 3.3V a 5V
Corriente máxima	2.5mA

Tabla 4. Características sensor DHT11.

Para configurar y obtener los datos de lectura de este sensor mediante su programación en el entorno de Arduino, se utiliza la librería *DHT.h*. Su conexión a Arduino se muestra en la Figura 13.

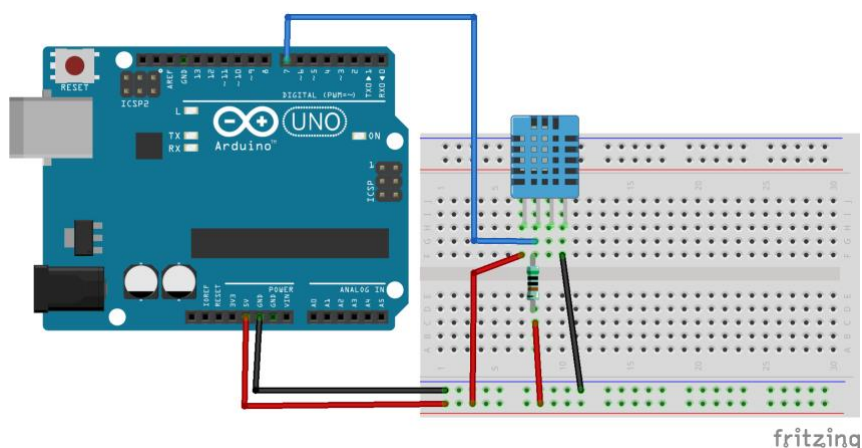


Figura 13. Conexión del sensor DHT11 con Arduino.



### 2.2.3 Higrómetro FC28

Un higrómetro FC28 [15] es un sensor que mide la humedad del suelo en función de la variación de su conductividad en función de su resistencia eléctrica. Las medidas que realiza no son demasiado precisas, pero para determinar si la tierra está húmeda o no es suficiente en este caso.

Su funcionamiento se basa en medir la resistencia entre 2 electrodos insertados dentro del suelo. El valor de esta resistencia dependerá de la humedad del suelo, teniendo para un suelo muy húmedo una resistencia muy baja (corto circuito) y para un suelo muy seco una resistencia será muy alta (circuito abierto).

Este dispositivo viene acompañado de una placa medidora que suministra una salida analógica proporcional a la resistencia medida por los electrodos resistivos y una salida digital dada por un comparador. Esta última se compara con el valor de referencia marcado mediante el potenciómetro incluido en la placa, obteniendo una señal HIGH de dicha salida digital cada vez que el valor medido por el sensor es mayor que la referencia establecida.

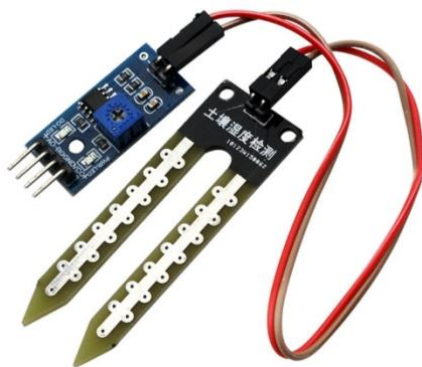


Figura 14. Sensor de humedad resistivo.

En este proyecto se ha utilizado la salida analógica, ya que obtiene datos más precisos numéricamente hablando. Los datos proporcionados por este sensor van desde 0 cuando está sumergido en agua a 1023 con el sensor al aire. Cuando detecta un ligero nivel de humedad, los valores que proporciona suelen estar entre 600 y 700, y para un suelo seco se posicionará entre 800 y 1023.

Debido al formato en el que salen estos datos, si se quiere obtener un porcentaje de humedad en 0 y 100%, hay que calibrar el sensor. De esta forma, mediante la función *map()* de Arduino se asignarán los valores de 0 y 1023 medidos por el sensor al 100% y al 0% de humedad, respectivamente.

A la hora de la verdad, se ha comprobado de forma experimental que, cuando el sensor detecta un valor de humedad del 100%, está leyendo un valor de 340 (y no de 0 como se supone que tendría que hacerlo). Sin embargo, cuando la humedad es del 0%, sí que identifica un valor de 1023.

La placa de medición que acompaña al higrómetro tiene 6 pines: dos de ellos se utilizan para dotar de polaridad al sensor y los otros cuatro son los que

conectan con Arduino. De estos últimos, se usarán el correspondiente a GND, a VCC (entre 3.3V y 5V) y el de la salida analógica (A0). En el esquema de la Figura 15 pueden verse estas conexiones.

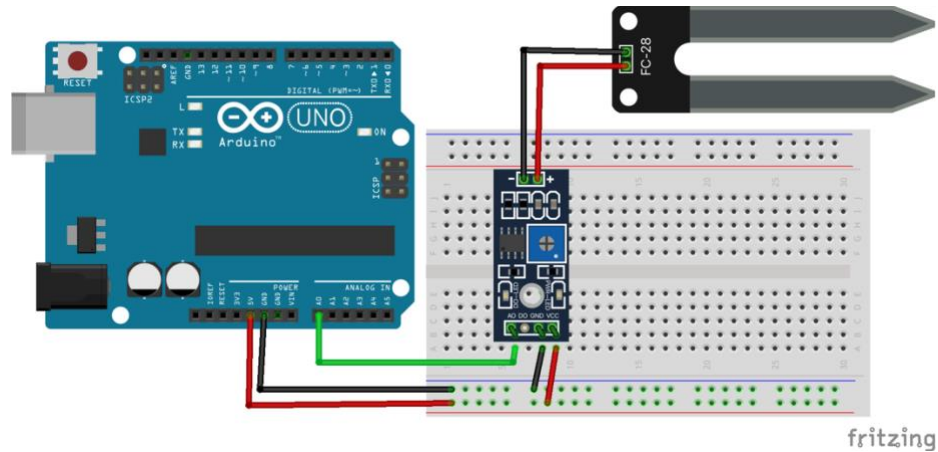


Figura 15. Conexión del sensor resistivo de humedad con Arduino.

Por último, cabe destacar la principal desventaja de este sensor. Al permanecer enterrado en el suelo y estar húmedo la mayor parte del tiempo, la corriente que circula por él produce un efecto de electrolisis que hace que se oxiden los contactos de cobre. La solución más rápida para esto es programar las mediciones de este sensor para que se realicen esporádicamente y no de forma continuada, con lo que no recibirá ningún tipo de alimentación mientras no esté midiendo.

#### 2.2.4 Sensor capacitivo de humedad de suelo

Además de usar un sensor de humedad resistivo como el mencionado en el apartado anterior, se ha utilizado un sensor de humedad capacitivo (Figura 16). La finalidad de usar ambos sensores es ver las diferencias en su funcionamiento y en los principios en los que se basan.



Figura 16. Sensor capacitivo.

En este caso, el sensor capacitivo utiliza una capacitancia entre electrodos que consigue aumentar de manera considerable la vida útil del dispositivo en comparación con el sensor resistivo anterior. [16]

Esos electrodos están cubiertos por una capa de protección contra la corrosión y su funcionamiento se basa en generar una onda cuadrada mediante un timer 555 (Figura 17) que, al aplicarla al sensor, en contacto con la tierra, se modifica por efecto de la capacitancia. Es decir, esa diferencia entre ondas es comparada por el sensor, lo que da lugar a una pequeña tensión que medirá la tarjeta Arduino. Así, cuanto mayor es el voltaje proporcionado por el sensor, menor es la humedad del suelo (alta capacitancia).

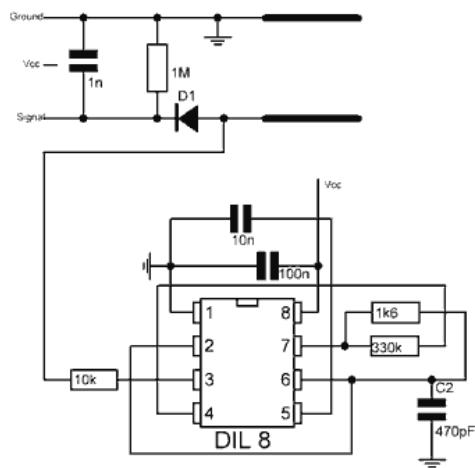


Figura 17. Esquemático del sensor capacitivo de humedad. [16]

La forma de obtener los datos de este sensor es la misma que en el caso del sensor resistivo: lee un rango de valores entre 0 y 1023 que se ajustará mediante la función *map()* en la programación del Arduino.

En este caso, para una humedad del 100% (metido totalmente en agua) el sensor lee un valor de 200, mientras que para un 0% de humedad (sensor al aire) detecta un valor de 523. Como se puede observar, los valores se alejan bastante de la referencia teórica, pero al calibrar el sensor con los valores reales se puede utilizar el dispositivo sin mayor inconveniente ya que se mantienen siempre estables.

Este sensor posee tres pines: GND, VCC y AUOT, y se conectarán al Arduino según se indica en la Figura 18.

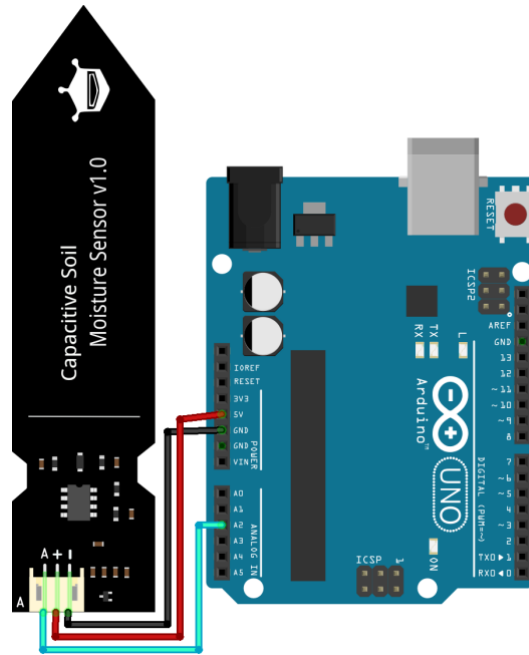


Figura 18. Conexión del sensor capacitivo de humedad con Arduino.

### 2.2.5 Sensor de nivel de agua

Este sensor podría denominarse también detector de presencia de agua, ya que de forma analógica valora la cantidad de agua que hay sobre su superficie.

En la Figura 19 se puede ver que tiene tres pines, dos de alimentación (+ y -) y uno de señal (S), que se conectará a una entrada analógica. La superficie de detección de agua está compuesta por un conjunto de pistas colocadas en paralelo de forma que el agua permita la conducción eléctrica entre ellas. El montaje proporciona una salida en tensión equivalente a la resistencia que ofrece el agua que hay entre dichos conductores. [17]



Figura 19. Sensor de nivel de agua.

La toma de datos funciona de forma similar que en los sensores de humedad vistos anteriormente. Este dispositivo recoge valores entre 0 y 1023 pero, al contrario que en los casos anteriores, cuando ahora se mide un valor

de 0 quiere decir que el nivel de agua es del 0%, siendo del 100% cuando lee un valor de 1023.

Este sensor no es demasiado preciso, y al comprobar de forma experimental el rango de valores en el que se mueve cuando está totalmente mojado y cuando está seco, se puede comprobar que varía de forma inestable.

Se ha intentado calibrar de la mejor forma posible y se ha establecido un valor de 580 para el 100% de humedad y de 250 cuando está prácticamente seco. Este último valor se debe a que, después de haber estado en contacto con el agua, tarda mucho en secarse del todo y obtener un valor de 0 (que es lo que marca cuando está directamente en contacto con el aire, totalmente seco).

Existen dispositivos mucho más precisos, pero también mucho más caros, por lo que para la simulación de este proyecto es suficiente con este y los anteriores sensores.

En el esquema de la Figura 20 se puede ver como conectar este sensor de nivel de agua a la placa Arduino.

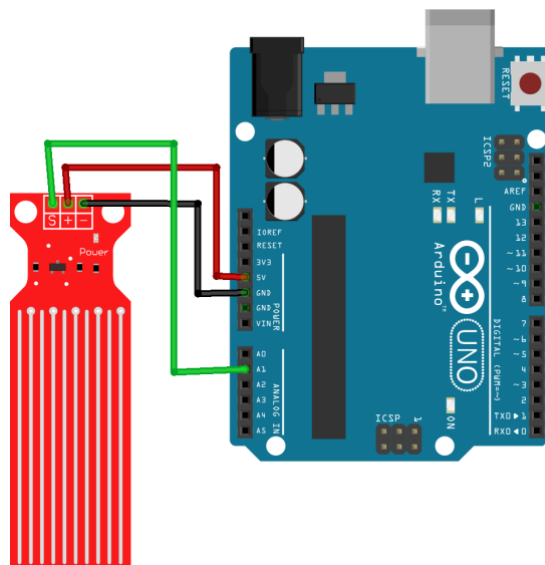


Figura 20. Conexión del sensor de nivel de agua con Arduino.

### 2.2.6 Servomotor SG90

Este micro servomotor es un pequeño actuador rotativo que permite controlar con precisión una determinada posición angular entre 0° y 180°. Está formado por un motor de corriente continua conectado a una serie de engranajes que reducen la velocidad y aumentan la fuerza del conjunto. [18]

Los elementos que lo componen se sitúan dentro de una carcasa de plástico que deja por fuera los cables de conexión y el eje del motor. A este último, irá conectada una pequeña hélice que simulará la apertura y el cierre del depósito de agua de la instalación. En un caso real de implementación del sistema, se utilizaría una electroválvula pero, en este caso, solo se utiliza el servomotor para simularlo.



Figura 21. MicroServo SG90.

Dispone de tres pines de conexión: dos de alimentación (GND y VCC) y un tercero de señal, que irá conectado a un pin digital a través del que recibirá una serie de pulsos que determinarán su funcionamiento.

La señal que hay que enviarle al servomotor para que se posicione en  $0^\circ$  tendrá un periodo de 20ms y un pulso de 1ms de duración. Por otro lado, para poder posicionarle en  $180^\circ$ , habrá que enviar un pulso de 2ms de duración sobre el mismo periodo. Por tanto, modulando el ancho del pulso entre 1ms y 2ms (Figura 21), se tiene control completo sobre su rango de giro.

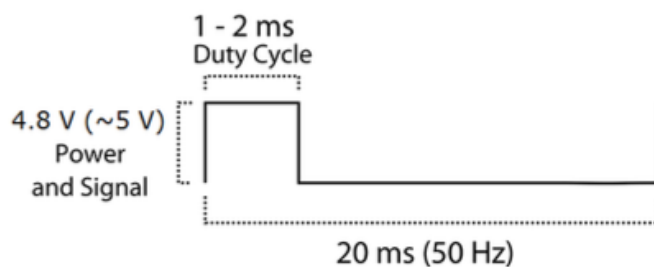


Figura 22. Ciclo de funcionamiento del servomotor.

Soporta una tensión de alimentación de entre 3V y 7.2V, pero su tensión óptima de funcionamiento se encuentra en torno a los 5V.

Para poder programar su funcionamiento en Arduino, hay que añadir la librería `Servo.h` y declarar este como objeto de dicha librería. Además, hay que declarar como variables el pulso mínimo y máximo del servomotor. Teóricamente, estos valores deberían de ser  $1000\mu\text{s}$  y  $2000\mu\text{s}$ , ya que es el

ancho de pulso correspondiente a  $0^\circ$  y  $180^\circ$ , respectivamente. Sin embargo, en la práctica se ha comprobado que esos valores de pulso se quedan algo cortos y que, en realidad, el pulso mínimo viene definido en torno a los  $550\mu\text{s}$  y el máximo alrededor de  $2300\mu\text{s}$ .

Por último, en la Figura 23 puede verse el diagrama de conexión del servomotor en la tarjeta Arduino.

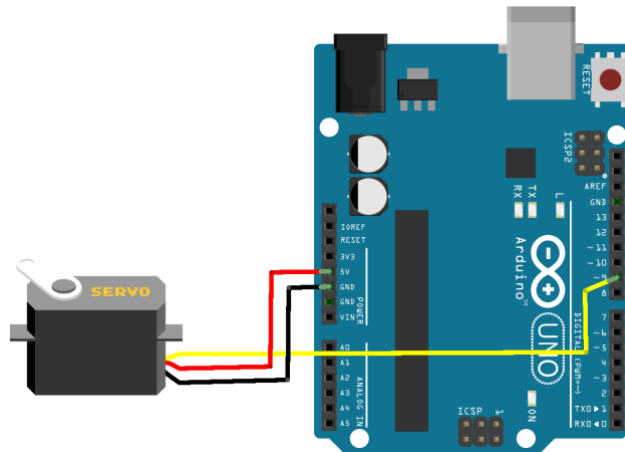


Figura 23. Servomotor conectado con Arduino.

### 2.2.7 Reloj de tiempo real (Módulo RTC) – DS3231

Este dispositivo va a permitir obtener la fecha y la hora en cualquier proyecto que lo requiera. Incluye una batería de 3V que mantiene alimentado el circuito integrado y guarda la cuenta del paso del tiempo aun cuando el módulo no está siendo alimentado de forma externa.

En este proyecto, se ha utilizado para poder informar al usuario en todo momento de la fecha y la hora actuales, además de para programar el riego automático en determinadas horas del día.

Tiene una gran precisión, ya que cuenta con un oscilador interno de cristal compensado por temperatura, y una alta velocidad, además de contar con un bajo consumo. Su tensión de alimentación se encuentra entre 3.3V y 5V y funciona mediante la interfaz de comunicación I2C.

El módulo RTC cuenta con 6 pines: dos de alimentación (GND y VCC), un pin de

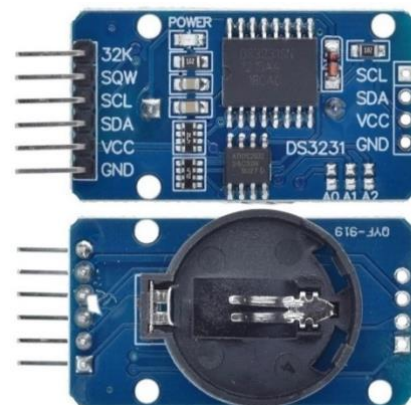


Figura 24. Reloj en tiempo real: módulo RTC.

señal de datos (SDA) y otro de señal de reloj (SCL), una salida directa del oscilador interno y una salida programable. En este caso, solo se utilizarán los cuatro primeros pines y se conectarán tal y como aparecen en la Figura 25.

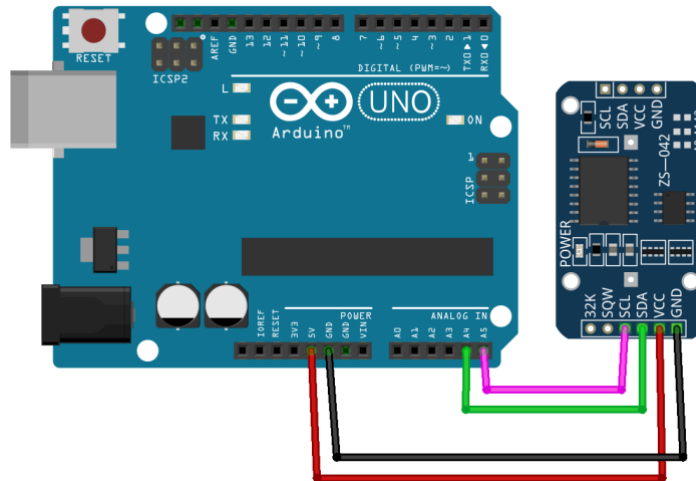


Figura 25. Conexión del módulo RTC con Arduino.

Para que fuera más sencillo el uso de este módulo, se ha utilizado la librería *RTClib.h* y, para conseguir un óptimo funcionamiento de la comunicación I2C, también se ha incluido la librería *Wire.h*. [19][20]

### 2.2.8 Relé 5V

Se ha utilizado un módulo relé simple de 5V para controlar el encendido y apagado de la bomba que se encarga de suministrar agua a la instalación de riego.

Un relé es un componente electromecánico que se utiliza para activar circuitos externos con un alto consumo de corriente o con distintos valores de tensión. En este caso, la bomba se va a alimentar de forma independiente al resto del sistema debido a la alta cantidad de corriente que consume cuando está funcionando. En el siguiente apartado de este capítulo, se describe en detalle la forma de alimentar cada parte del circuito.

Los relés están compuestos de una bobina de cobre y de una serie de contactos vinculados mediante un brazo en forma de L. Al circular corriente por la bobina se crea un campo magnético que se concentra en el núcleo y hace que se comporte como un electroimán. [21]



El brazo que une los contactos pivota por encima de la bobina y se encuentra unido al contacto común del relé. Además del común, un relé posee un contacto NO (normalmente abierto) y otro NC (normalmente cerrado).

Cuando el relé está en reposo, es decir, no se está aplicando ninguna tensión a la bobina, el terminal común toca el terminal normalmente cerrado. Si, en cambio, se energiza la bobina, se creará un campo magnético que hará que el brazo se mueva y junte el contacto común con el normalmente abierto. Con esto, se logra cerrar el circuito entre el terminal común y el NO creando un circuito aislado totalmente de la bobina.

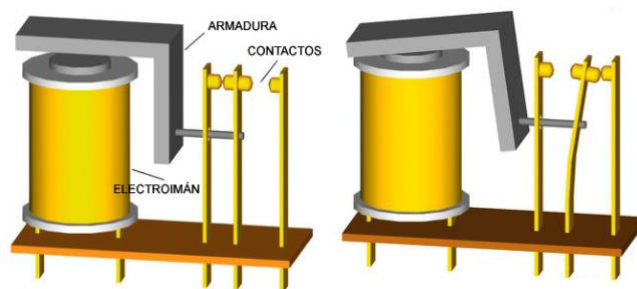


Figura 26. A la izquierda, relé en reposo. A la derecha, bobina energizada. [21]

La bobina se alimenta con 5V desde la tarjeta de Arduino y en los terminales de este, se pueden controlar tensiones de hasta 220V y 10A.

El terminal más usado es el denominado normalmente abierto, ya que su funcionamiento se asemeja al de un pulsador. Así, en reposo se tiene un circuito abierto y, alimentando la bobina, se consigue un circuito cerrado.



Figura 27. Partes del módulo relé simple.

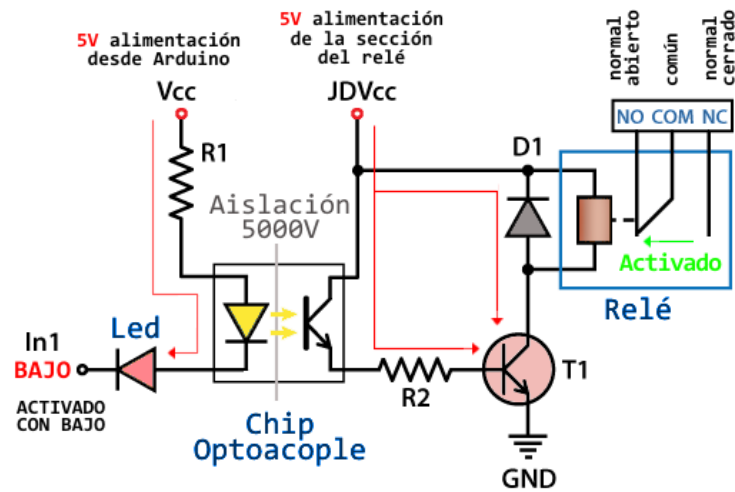


Figura 28. Esquema interno del módulo relé.

Dichos contactos del relé (NO, COM y NC) están conectados a una bornera. Por otro lado, dispone de tres pines: dos de alimentación (GND y VCC) y uno de señal (IN) que irá conectado a un pin digital de la tarjeta Arduino. Todos estos puertos de conexión pueden verse en el esquemático de la Figura 28, en el que también aparecen un transistor NPN y un optoacoplador cuya función es aportar seguridad al componente a la hora de energizar la bobina. Además, este módulo relé simple cuenta con un diodo LED rojo y otro verde que indicarán si la bobina está en reposo o alimentada, respectivamente.

Por último, en la Figura 29 se muestra la conexión de cada pin del relé.

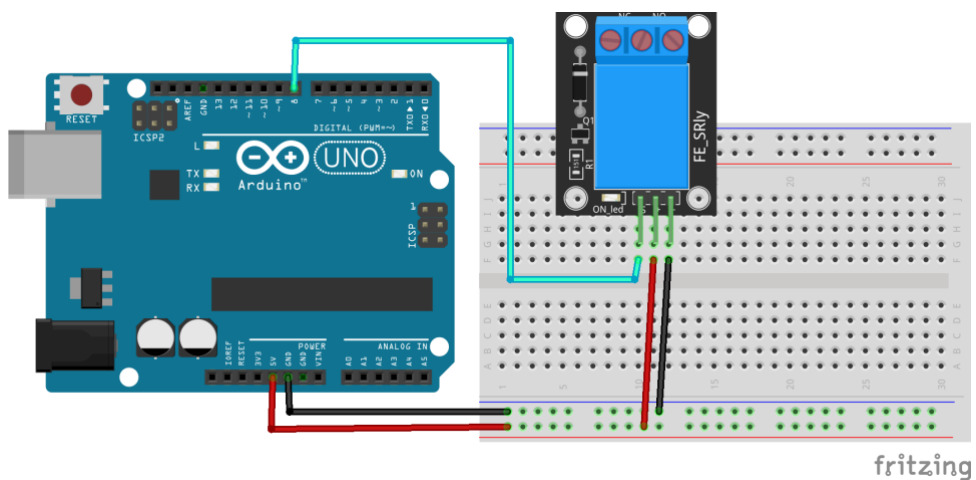


Figura 29. Conexión del relé con Arduino.

### 2.2.9 Bomba de agua

Para la distribución del agua desde el depósito hasta el sistema de riego por goteo, se ha utilizado una bomba como la de la Figura 30.



*Figura 30. Mini-bomba de agua sumergible.*

Esta bomba está controlada por un relé como el descrito anteriormente y se alimenta mediante un sistema fotovoltaico con batería que se describirá con detalle en el siguiente apartado de este capítulo.

Su tensión de funcionamiento se encuentra entre 2.5V y 6V y está formada por un motor de corriente continua de 100-200mA.

Este motor está conectado a un elemento rotativo llamado rodete, que está formado por una hélice que, al girar, impulsa el agua que pasa por ella. Cuando el agua entra por el centro del rodete, la hélice incrementa su energía hasta que sale de la bomba en sentido tangencial. [22]



*Figura 31. Construcción de la bomba sumergible.*

La bomba utilizada es una bomba sumergible, es decir, el motor se encuentra dentro de un encapsulado sellado que permite introducir todo el cuerpo de la bomba en el agua evitando así tener que emplear una tubería de aspiración.

Sin embargo, para llevar el agua desde el depósito al sistema de riego, se emplea una tubería de 7mm de diámetro que se conecta a la salida de la bomba. Esta tubería, a su vez, se conecta a un enlace de 4 vías que reparte el agua a través del circuito de riego.

Como se mencionó anteriormente, la bomba activa y desactiva su funcionamiento gracias al módulo relé. Para ello, se conecta al contacto normalmente abierto de este para que, cuando se energice la bobina, la alimentación llegue hasta la bomba.

### 2.2.10 Pantalla OLED

Para hacer más intuitivo el sistema, se ha diseñado una interfaz que proporciona al usuario información sobre el estado de la instalación, como la humedad y temperatura ambiente, la humedad de la tierra o el nivel de agua que hay en el depósito. Estos datos se obtienen a partir de los sensores descritos anteriormente y se reflejan en una pequeña pantalla OLED como la de la Figura 32.

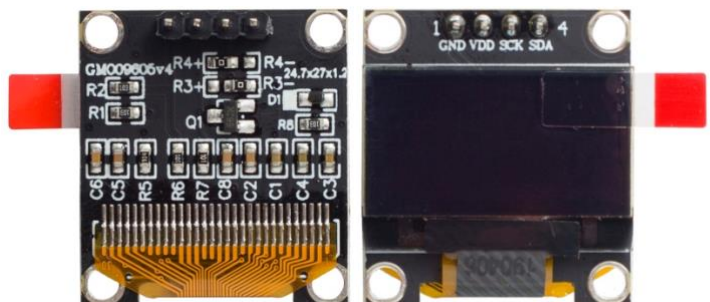


Figura 32. Pantalla OLED 128x64.

A través de unos pulsadores, configurados especialmente para este proyecto, se puede navegar por la pantalla y acceder a dicha información, además de activar el riego automático o el riego manual programados en el sketch.

La OLED empleada es una pantalla monocromática de 0,96" con una resolución de 128x64 píxeles, con interfaz I2C intercambiable. Su voltaje de funcionamiento está entre 3.3V y 5V, y cuando está encendida consume 10mAh. [23]

La equivalencia de sus pines viene reflejada en la Tabla 5.

Para configurarla de forma adecuada, ha sido necesario añadir distintas bibliotecas de Arduino. Para gestionar el uso del protocolo I2C se ha utilizado la librería *Wire.h* y para mostrar en la pantalla los diferentes datos se ha utilizado la librería *OLED\_I2C.h*. En el [Capítulo 3.1.: Librerías utilizadas](#) se describen estas librerías con mayor detalle.

PIN	FUNCIÓN
GND	Terminal de tierra a 0V
VDD	Pin de alimentación a 3.3/5V
SCK	Señal de reloj I2C
SDA	Señal de datos I2C

Tabla 5. Pines de la pantalla OLED.

Para poder navegar por los distintos escenarios configurados en la pantalla se han empleado cuatro pulsadores como los que se muestran en la Figura 33. Como puede verse, estos botones tienen cuatro patillas que están conectadas entre sí de dos en dos (la 1 con la 2 y la 3 con la 4).



Figura 33. Botón pulsador.

Estos botones pueden configurarse de dos formas distintas: con resistencia *pull-up* o con resistencia *pull-down* (Figura 35). En este caso se ha utilizado la resistencia *pull-down*, la cual se conecta entre GND y el punto de unión de uno de los extremos del pulsador y la señal de datos empleada. El segundo extremo de cada pulsador se conecta directamente a la señal de alimentación, que es de 5V. [24]

El valor de las resistencias utilizadas es de 10kΩs, con lo que se evita que circule mucha corriente cuando se oprime el botón y, a la vez, el ruido que puede llegarle al pin de datos no es significativo.

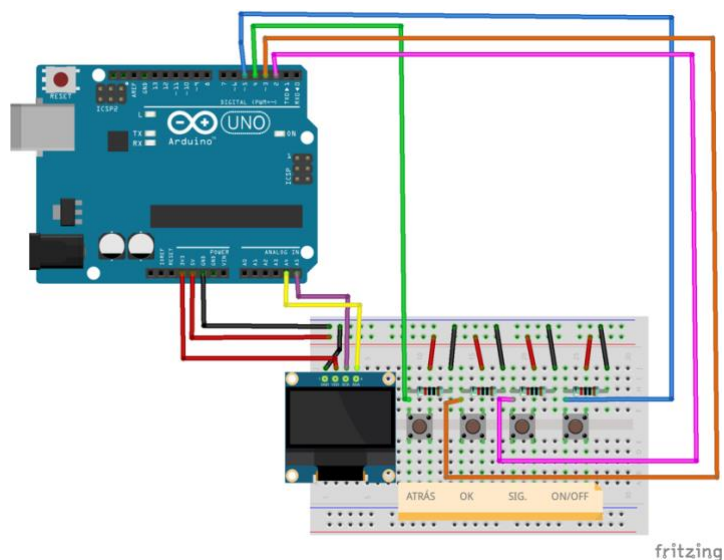


Figura 34. Esquema de conexión de la pantalla OLED y de los pulsadores.

La función principal de la configuración de estas resistencias es determinar el estado lógico en el que se encuentra cada pulsador cuando no está siendo accionado. En el caso de la resistencia *pull-down*, se mantiene un estado bajo (0V - LOW) cuando no se presiona el botón y un estado alto cuando se acciona (5V - HIGH).

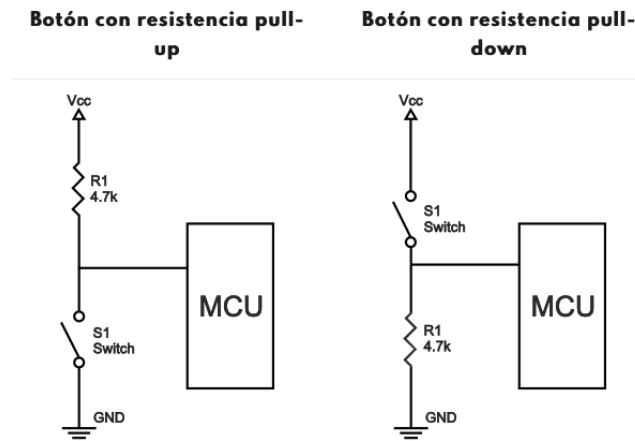


Figura 35. Configuración de resistencias para pulsadores.

El diseño de las distintas vistas de la pantalla se muestra en el [Capítulo 4.2.: Manual de funcionamiento](#), en la Tabla 10.

## 2.3. SISTEMAS DE ALIMENTACIÓN

### 2.3.1 Alimentación de la tarjeta Arduino

El sistema Arduino UNO R3 necesita una tensión de alimentación de 5V. Esta tensión puede conseguirse a través de una conexión USB al puerto de un ordenador, mediante una fuente de alimentación conectada al puerto Jack de la tarjeta o directamente desde el puerto VIN. [25]

En este caso se va a utilizar un adaptador de corriente AC/DC de 9V y 0.5<sup>a</sup> como el de la Figura 36. Al ser una entrada directa de corriente, la conexión del adaptador debe respetar una polaridad: el polo positivo debe situarse en el centro del conector.



Figura 36. Adaptador de corriente para Arduino.

El voltaje adecuado aplicable en esta entrada debe estar entre 7V y 12V, ya que al aplicar una tensión por debajo a por encima de estos valores puede dañar el regulador interno del Arduino y producir sobrecalentamiento en la tarjeta.

Como medida de seguridad adicional, la entrada tiene un diodo de protección contra la inversión de polaridad, por lo que si en algún momento no se respetara la tarjeta no sufriría daños y simplemente no funcionaría.

### 2.3.2 Alimentación externa

Como se vio al principio de la memoria del proyecto, se ha buscado realizar una instalación eficiente y sostenible, que sea lo más respetuosa posible con el medio ambiente y que no desperdicie energía ni agua durante su funcionamiento.

Por ello, para alimentar la bomba de agua, que es la parte que más consume, se ha utilizado un sistema de alimentación basado en paneles solares con batería. [26]

Cabe destacar que los módulos elegidos para llevar a cabo dicha instalación de alimentación son módulos comerciales con unas características que satisfacen las necesidades del proyecto y que, dada la envergadura del TFG y que su principal propósito es el diseño de un sistema de riego, no han sido diseñados y calculados extensamente. Simplemente se han tenido en cuenta sus características principales, tales como la capacidad de suministro de tensión o corriente, y su forma de funcionamiento básica.

Se han usado seis placas solares de 3V y 0.3W cada una. Para conseguir la tensión y corrientes necesarias para cargar la batería que alimentará la bomba, se han conectado los paneles de la siguiente forma: tres grupos de dos paneles en serie, para lograr una tensión máxima de 6V y todos ellos en

paralelo para conseguir los 300mA. Estos cálculos se detallan en el [Capítulo 5: Otros cálculos](#).

La batería utilizada es de 3.8V y se cargará mediante los paneles solares a través de un módulo TP4056 (Figura 37), que es un cargador de baterías de litio basado en un chip encapsulado en formato SOP-8. Además, dispone de un puerto mini-USB que sirve para conectar la alimentación.

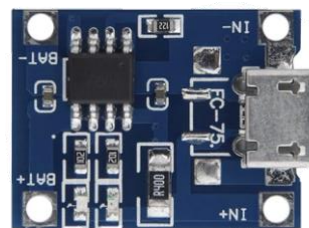


Figura 37. Módulo TP4056.

Por otro lado, el TP4056 [27] tiene dos LED, uno de carga y otro de completado, que se encenderán mientras se esté cargando la batería y cuando este proceso haya finalizado, respectivamente. Además, en el caso de no poder cargar la batería porque la incidencia del sol no es suficiente, ambos LED se encenderán avisando del “vacío” entre sus bornes de conexión de la carga. En un caso de implementación real, habría que utilizar un inversor que permitiera alternar entre la alimentación de la batería y los paneles, y la alimentación de la red eléctrica para que el sistema de bombeo de agua no dejara de funcionar en ninguna circunstancia.

Por último, tiene otros dos bornes con la inscripción BAT+ y BAT- donde se conectará la batería. Se añadirá un diodo 1N4007 entre el polo positivo de los paneles solares y el borne IN+ del módulo TP4056 para limitar el paso de corriente en una sola dirección y limitar el voltaje para proteger el circuito.

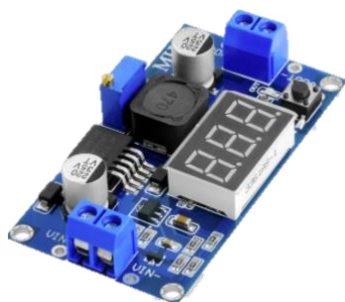


Figura 38. Step-up DC-DC Boost Converter.

Como se ha mencionado anteriormente, se ha utilizado una batería de 3.8V pero se va a regular su salida de tensión mediante un convertidor de tensión (Figura 38) que ajustará la salida de la batería para lograr alimentar la bomba a la potencia deseada.

Se trata de un convertidor elevador DC/DC.

El rango de tensión de entrada que soporta es de 2V a 24V y el de salida va de 5V a 28V (regulable). La corriente máxima de salida que va a suministrar es de 1A y su eficiencia de conversión puede alcanzar el 93%.

La tensión de salida va a regularse mediante el potenciómetro disponible en la placa y, a través del display incorporado, se podrán visualizar tanto la tensión de entrada como la de salida regulada. Esto facilitará su configuración.



Al ser un convertidor elevador, el voltaje de salida siempre será mayor que el de entrada. En este caso, se va a utilizar para fijar una tensión estable de 5V para alimentar la bomba desde la batería de 3.8V.

Se podría utilizar una batería de 5V directamente, sin necesidad de utilizar un convertidor elevador pero, debido a que en la práctica la tensión de entrada no tiene por qué ser totalmente constante y puede tener bajadas y subidas, se utiliza este dispositivo para fijar la tensión a un valor siempre constante independiente de la entrada.

Por tanto, el circuito final de alimentación de la bomba de agua queda tal y como se muestra en la Figura 39.

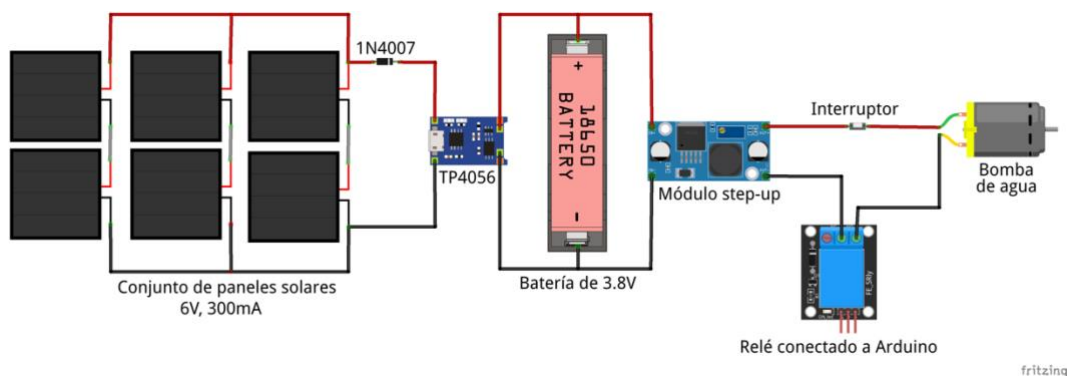


Figura 39. Circuito de alimentación externa para la bomba de agua. [26]

## 2.4. SISTEMA FINAL

Finalmente, tras la configuración de cada uno de los dispositivos se tiene la implementación del sistema final completo. A continuación, se describe brevemente su estructura básica y se representa la misma mediante el diagrama de flujo de la Figura 40.

En la página principal, se mostrarán la fecha y la hora actuales.

Una vez que se accede al menú principal, aparecen cuatro opciones diferentes por las que poder navegar:

1. Activar/desactivar el riego automático.
2. Activar/desactivar el riego manual.
3. Consultar el estado actual de la instalación, accediendo a la lectura de los sensores de temperatura y humedad. Además, en función de los

valores medidos, aparecerán una serie de mensajes como los que aparecen en la Tabla 6.

CONDICIÓN	MENSAJE
Humedad del suelo < 50%	“Suelo seco”.
50% < Humedad del suelo < 80%	“Suelo húmedo”.
Humedad del suelo > 80%	“Exceso de humedad”.
Nivel de agua < 30%	“Depósito vacío”.
30% < Nivel de agua < 60%	“Nivel bajo. Llenar”.
Nivel de agua > 80%	“Depósito lleno”.

Tabla 6. Condiciones de la instalación.

4. Consultar el estado del depósito gracias al sensor de nivel de agua que medirá el porcentaje de líquido que hay. También se puede abrir/cerrar el depósito de forma manual para que se llene, lo que se simulará con el movimiento de la hélice del servomotor.

El modo automático arrancará siempre y cuando se cumplan las siguientes condiciones:

- Es hora de regar, la cual puede configurarse en el código del programa.
- La humedad de la tierra está por debajo del 75%.
- El nivel del depósito está por encima del 30%.

A su vez, parará de regar cuando se den una serie de circunstancias:

- El nivel del depósito ha caído por debajo del 30%.
- La humedad está por encima del 75% y ya han pasado 3 minutos desde que comenzó el riego. Todos estos parámetros son modificables en el sketch de Arduino mediante la conexión de la tarjeta al ordenador.

En el [Capítulo 3.2.: Descripción del código](#), junto con la explicación del código programado, se describe en detalle el funcionamiento completo del sistema.

Además, en el [Capítulo 4.2.: Manual de funcionamiento](#) se adjunta un breve manual de uso que facilitará la comprensión del funcionamiento de la instalación.

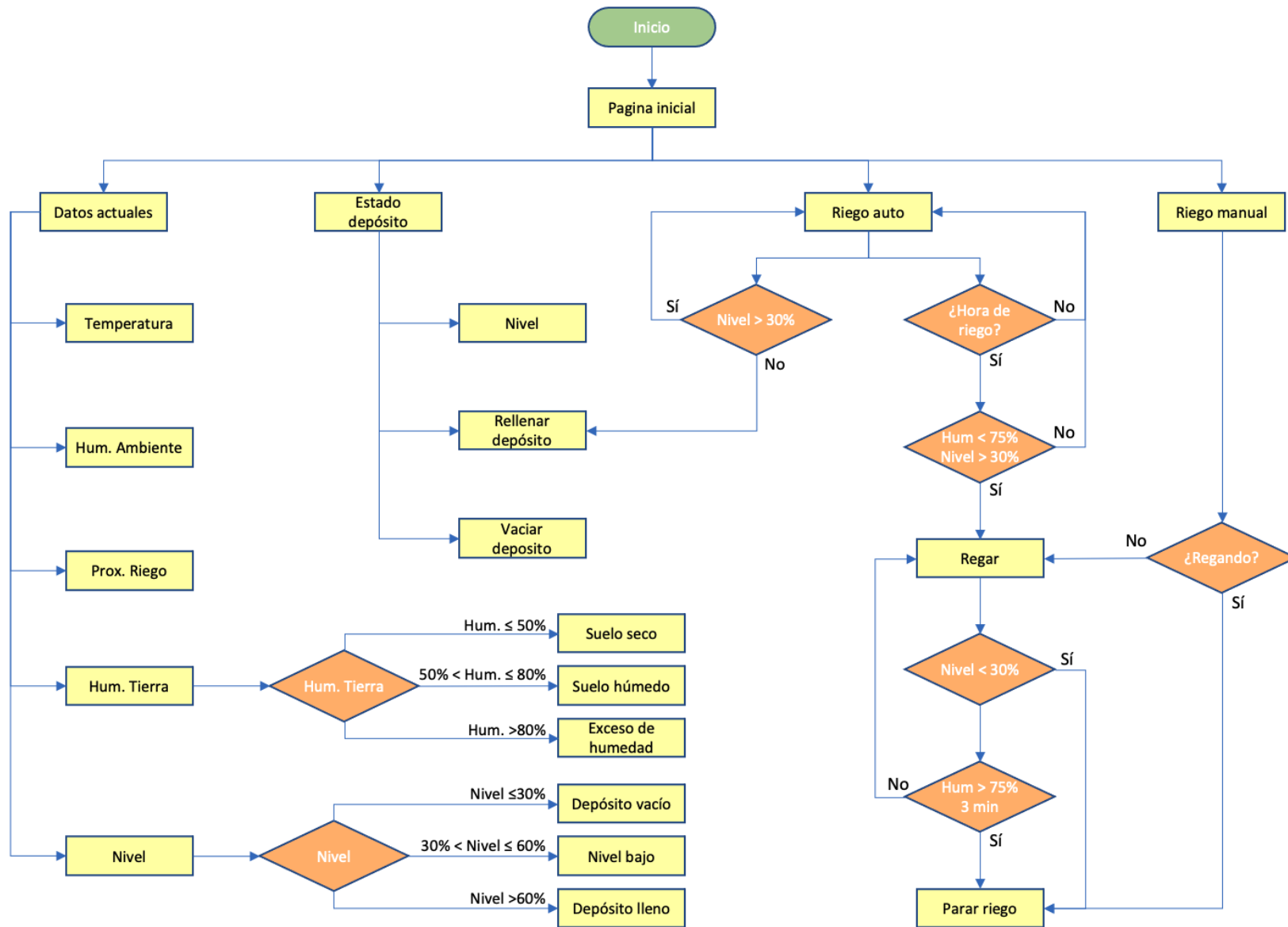


Figura 40. Diagrama de flujo del funcionamiento del sistema.

## CAPÍTULO 3: PROGRAMACIÓN

En este capítulo, se va a describir cada librería, función y variable que se ha utilizado para la implementación del código.

El sistema va a disponer de varias entradas, que serán los datos de reloj, las señales enviadas a través de los pulsadores y los valores leídos por los sensores. Estos últimos valores servirán para determinar el nivel de humedad de la tierra y el nivel de agua del depósito, que regularán el funcionamiento de la instalación. En función del nivel en el que se encuentren, el sistema determinará si es necesario regar o no, activando en los correspondientes casos el riego automático programado. También podrá activarse dicho riego de forma manual siempre que el usuario quiera.

Por otro lado se tiene una serie de salidas que pueden dividirse en actuadores y elementos de señalización o visualización.

Los actuadores, es decir, la bomba de agua y el servomotor, van a trabajar en función de los valores leídos en las entradas (sensores, reloj o pulsadores). De esta forma, en función de las instrucciones recibidas desde el modo de riego manual o desde el modo de riego automático, los actuadores serán o no activados. La bomba se encargará de la distribución del agua por el circuito de riego siempre que este esté activado y el servomotor simulará la apertura de una hipotética electroválvula que permitiría el llenado del tanque desde el que la bomba distribuye el agua.

Además, se dispone de una serie de elementos que facilitarán la comprensión del estado del sistema. En este caso, se han configurado tres diodos LED de distintos colores, rojo, verde y amarillo, que se encenderán cuando el depósito necesite ser llenado, cuando se esté regando o cuando sea necesario regar, respectivamente. Su funcionamiento se basará también en los valores leídos por los distintos sensores.

Por otro lado, se ha configurado una pantalla OLED que permite visualizar los datos medidos y el estado del sistema. Además, permite gestionar la activación y desactivación de los dos modos de riego disponibles (el manual y el automático) y gestionar el llenado del depósito de forma manual.

Por último, se tienen los diferentes sistemas de alimentación empleados, tanto el de la tarjeta Arduino como el de la bomba de agua. Estos sistemas no influyen de ninguna manera a la programación del código del programa, pero

es importante tenerlos en cuenta ya que son una parte imprescindible del sistema.

En el diagrama de bloques de la Figura 41, se puede observar la distribución simplificada del sistema final implementado.

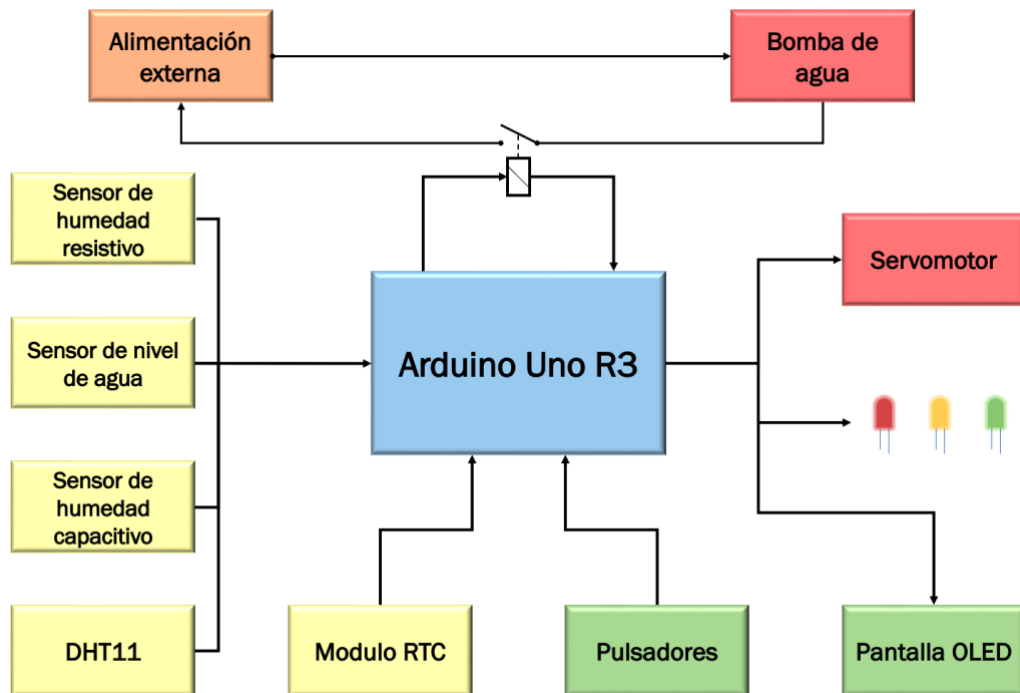


Figura 41. Diagrama de bloques de la instalación completa.

### 3.1. LIBRERÍAS UTILIZADAS

A continuación, se describe brevemente la función principal de cada una de las librerías utilizadas en la implementación del código.

```
#include <Wire.h> //Permite la comunicación por bus I2C
#include <OLED_I2C.h> //Pantalla OLED con comunicación I2C
#include <DHT.h> //Sensor de temperatura y humedad DHT11
#include <Servo.h> //Servomotor
//Control del reloj de tiempo real (módulo RTC)
#include <RTClib.h>
#include <SimpleTimer.h>
```

Figura 42. Declaración de librerías en la cabecera del código.

### 3.1.1 <Wire.h>

Esta librería [28] permite la comunicación con dispositivos I2C/TWI, funcionando como maestros a otros dispositivos o como esclavos que reciben peticiones y envían datos.

En el caso de la tarjeta Arduino UNO R3, que es la que se ha usado en este proyecto, los pines correspondientes a este medio de comunicación son los denominados A4 (SDA) y A5 (SCL).

### 3.1.2 <OLED I2C.h>

Esta librería [29] está diseñada para facilitar el uso y la configuración de la pantalla OLED empleada en el proyecto. Sus principales características y rangos de configuración se resumen a continuación:

- Personalizar la fuente utilizada para mostrar los datos en la pantalla.
- Declarar la pantalla como un objeto al que poder aplicar el resto de las funciones.
- Opción de inicializar dos tipos diferentes de pantalla OLED: la que tiene una resolución de 128x32 píxeles y la de 128x64 píxeles. Esta última es la empleada en el proyecto y no hace falta declararla en el sketch porque es la que utiliza la librería por defecto.
- Configuración del brillo de la pantalla.
- Función *sleepMode*, que permite encender y apagar la pantalla.
- Funciones *update* y *clrScr*, para actualizar los datos en la pantalla y limpiar la información mostrada, respectivamente.
- Funciones para invertir la pantalla, un texto o un determinado píxel, haciendo que si estaba encendido se apague y viceversa.
- Funciones para imprimir cadenas de caracteres (*print*) o cadenas de números de tipo entero (*printNumI*) o de tipo coma flotante (*printNumF*) por la pantalla.
- Funciones predeterminadas para dibujar rectas, figuras geométricas o mapas de bits.

### 3.1.3 <DHT.h>

Esta librería [30] permite utilizar los sensores DHT11 y DHT22, obteniendo sus valores de humedad y temperatura.

En primer lugar, se debe declarar una instancia de la clase DHT para poder acceder a las funciones del sensor y que recibe como parámetros el pin al que se conecta el sensor empleado y el tipo de sensor DHT que se va a utilizar (DHT11 en este caso).

### 3.1.4 <Servo.h>

Esta librería [31] permite a la tarjeta Arduino tomar control sobre una gran variedad de servomotores.

Además, solo es compatible con los pines 9 y 10 de la tarjeta Arduino, por lo que la conexión de la señal del servomotor empleado deberá realizarse en alguno de ellos. En este caso se ha utilizado el pin digital 9.

Mediante la función *write* de esta librería, se determina el ángulo de giro que realizará el servo medido en grados. También puede determinarse su posición en microsegundos mediante la función *writeMicroseconds*, pero se alcanza una menor precisión debido a que cada fabricante de servomotores utiliza unos valores diferentes.

### 3.1.5 <RTClib.h>

Esta librería [32] va a permitir conocer la hora y la fecha en cualquier momento gracias al módulo RTC vía hardware. Además, este módulo trabaja mediante comunicación I2C, por lo que será necesaria también la librería descrita anteriormente <Wire.h>.

Lo más importante de esta librería es la forma de establecer la hora por primera vez en el RTC. Para ello, la primera vez que se ejecute el programa deberá utilizarse la función *RTC.adjust(DateTime(F(\_\_DATE\_\_), F(\_\_TIME\_\_)))*, con la que se coordinará la hora y fecha del ordenador con el módulo de reloj en tiempo real.

### 3.1.6 <SimpleTimer.h>

Su función es, básicamente, controlar la gestión del tiempo. [33]

## 3.2. DESCRIPCIÓN DEL CÓDIGO

Se va a comentar en profundidad cada parte del código, explicando cada función y variables utilizadas. Además, en el propio código aparecen comentarios que lo describen brevemente.

Las librerías empleadas son lo primero que aparece en la cabecera del código, pero ya se han descrito en el apartado anterior.

En primer lugar, se ha asignado un pin a cada dispositivo conectado a la tarjeta Arduino de la forma que aparece en la Tabla 7.

DISPOSITIVO	VARIABLE	PIN	TIPO
Pantalla OLED	myOLED	SDA, SCL	Analógico
Sensor de humedad DHT11	sensorDHT	7	Digital
Sensor de humedad resistivo	sensorRes	A0	Analógico
Sensor de humedad capacitivo	sensorCap	A2	Analógico
Sensor de nivel de agua	sensorHW	A1	Analógico
Relé	bomba	8	Digital
Servomotor	pinServo	9	Digital
LED rojo	redLED	6	Digital
LED verde	greenLED	10	Digital
LED amarillo	yellLED	11	Digital
Pulsador "atrás"	z	4	Digital
Pulsador "OK"	y	3	Digital
Pulsador "siguiente"	x	2	Digital
Pulsador "ON/OFF"	w	5	Digital

Tabla 7. Declaración de pines en el código de Arduino.

A continuación, se han declarado las variables globales que se van a utilizar a lo largo del programa. En la Figura 43, aparecen todas ellas junto con una breve descripción de cada una de ellas



```

///////// Declaración de variables:
extern uint8_t SmallFont[]; //Fuente utilizada en la pantalla incluida en la librería OLED_I2C
int temp, humDHT; //Variables que guardan la temperatura y la humedad leídas por el sensor DHT11
int humedad; //Variable que guarda el valor de la humedad de la tierra
int nivel; //Variable para guardar el valor del nivel de agua del depósito
int i = 13; //Posición en y del cursor ">" en la pantalla
int page = 0; //Página por defecto
//Variables que guardan el estado de activación de cada elemento:
boolean releON = false, autoON = false;
boolean selecDown = true, selecBack = true, selecOK = true, selecSleep = true, selecRele = true;
boolean currentSleep = true; //Guarda el estado actual de la pantalla (encendida/apagada)
int pulsoMIN = 550; //Pulso mínimo del servo correspondiente al ángulo mínimo (0°)
int pulsoMAX = 2300; //Pulso máximo del servo correspondiente al ángulo máximo (180°)
//Variables para calcular el tiempo que ha pasado desde el inicio del programa
long currentMillis = 0;
long previousMillis = 0;
long autoMillis = 0;
boolean regando = false; //Indica si se está regando o no
// Horas de riego automático
int h_riego1 = 8; //Hora de comienzo del primer riego
int m_riego1 = 0; //Minuto de comienzo del primer riego
int h_riego2 = 21; //Hora de comienzo del segundo riego
int m_riego2 = 0; //Minuto de comienzo del segundo riego
int prox_riego = 0; //Guarda el tiempo que queda hasta el próximo riego programado

```

Figura 43. Declaración de variables globales.

### 3.2.1 Función `bcdToDec()`

Esta función [34] convierte un valor en formato BCD en su equivalente en formato decimal.

El módulo RTC devuelve un valor en formato BCD en un byte (8 bits) y la tarjeta Arduino va a interpretarlo como un único valor en binario. Como lo que se quiere conseguir es una cifra de dos dígitos, ese valor que lee Arduino debe dividirse en 4 (decenas) + 4 (unidades) bits. Por tanto, es necesario realizar esta conversión para obtener los valores correctos de fecha y hora.

```

byte bcdToDec(byte val) {
    return ( (val / 16 * 10) + (val % 16) );
}

```

Figura 44. Función `bcdToDec`.

### 3.2.2 Función `setup()`

Esta función es en la que se determinan las funciones principales que va a realizar el microcontrolador y es la primera función que se ejecuta en un programa de Arduino.

En ella se establecen las acciones que requieren una única ejecución. Por tanto, se han declarado las siguientes instrucciones:

- Inicialización de la pantalla OLED, del módulo RTC, del sensor DHT11 y del servomotor.
- Ajuste y carga de la hora y fecha del reloj en tiempo real (RTC).
- Declaración del pin correspondiente a la señal de relé como salida e inicialización del mismo relé en HIGH (apagado).
- Inicio del servomotor en 180°, que es la posición asignada a “depósito cerrado”.
- Inicio de la pantalla OLED apagada.

```
void setup() {
    myOLED.begin(); //Inicio de la pantalla OLED
    myOLED.clrScr(); //Limpiar la pantalla al inicio
    myOLED.setFont(SmallFont); //Establecer tipo de letra en la pantalla, cada caracter ocupa 6x8px

    RTC.begin(); //Inicio del módulo RTC
    //ajustar el reloj a la hora y fecha del ordenador
    //comentar esta línea después de cargar por primera vez el programa en la tarjeta Arduino
    RTC.adjust(DateTime(F(__DATE__), F(__TIME__)));

    dht.begin(); //Inicio del sensor de humedad DHT11

    pinMode(bomba, OUTPUT); //Asignar el pin 8 como salida
    digitalWrite(bomba, HIGH); //Inicia el relé apagado

    servo.attach(pinServo, pulsoMIN, pulsoMAX); //Declaración de parámetros del servomotor
    servo.write(180); //Inicia el deposito cerrado

    myOLED.sleepMode(SLEEP_ON); //La pantalla empieza apagada al cargar el programa
    myOLED.update();
}
```

Figura 45. Función setup().

### 3.2.3 Función loop()

En inglés, *loop* significa lazo o bucle. Por tanto, esta función es la que se ejecuta durante un número de veces infinito. Se ejecuta justo después de la función *setup()* y se repite de forma indefinida hasta que se apaga o reinicia el microcontrolador.

En primer lugar, se realiza la llamada a las funciones que se quieren tener en ejecución durante todo el programa. En este caso, se ejecutan las funciones que aparecen en la Figura 46. Más adelante se explicará que tarea realiza cada una.

```
void loop() {
    //Llamada de funciones
    lecturas();
    RiegoAuto();
    sleep();
    back();
}
```

Figura 46. Función loop(): llamada a funciones.

A continuación, se define el menú que va a aparecer en la pantalla OLED. Dicho menú, se divide en diferentes “páginas” y en cada una de ellas aparecerá una información diferente. La apariencia de dichas páginas se muestra en el [Capítulo 4.2.: Manual de Funcionamiento](#).

- Página 0: es la apariencia inicial de la interfaz. En ella se declara en qué lugar se quiere posicionar la palabra “MENU” y, mediante la función `pagina0()`, se muestra la fecha y la hora actuales. Además, si se pulsa el botón “OK” se accederá a la siguiente página (página 1).

```

if (page == 0) {
    myOLED.clrScr();
    myOLED.invertText(true);
    myOLED.print("    MENU    ", CENTER, 57);
    myOLED.invertText(false);
    pagina0(); //Llamada a la función de pág. 0
    myOLED.update();

    if (ok())
        page = 1; //Si se pulsa "OK" estando en la pág. 0, se accede a la pág. 1
}

```

Figura 47. Función `loop()`: configuración de página 0.

En la función `pagina0()`, que aparece en la Figura 48, se establece el formato de fecha y hora que se quiere mostrar por pantalla. Además, si el mes, el día, la hora o los minutos son menores que 0, se añade un 0 delante de ellos para que el formato sea siempre: dd/mm/aaaa, hh:mm.

```

void pagina0() {
    byte second, minute, hour, dayOfWeek, day, month, year;
    readDS3231time(&second, &minute, &hour, &dayOfWeek, &day, &month, &year);
    if (hour < 10) {
        myOLED.print("0", 47, 15); //Si la hora es <10, se añade un 0 delante para darle formato de 00:00
        myOLED.printNumI(hour, 53, 15);
    }
    else myOLED.printNumI(hour, 47, 15);
    myOLED.print(":", 59, 15);
    if (minute < 10) {
        myOLED.print("0", 65, 15);
        myOLED.printNumI(minute, 71, 15);
    }
    else myOLED.printNumI(minute, 65, 15);
    if (day < 10) {
        myOLED.print("0", 38, 30);
        myOLED.printNumI(day, 44, 30);
    }
    else myOLED.printNumI(day, 38, 30);
    myOLED.print("/", 50, 30);
    if (month < 10) {
        myOLED.print("0", 56, 30);
        myOLED.printNumI(month, 62, 30);
    }
    else myOLED.printNumI(month, 56, 30);
    myOLED.print("/", 68, 30);
    myOLED.printNumI(year, 74, 30);
}

```

Figura 48. Función `pagina0()`.

- Página 1: en ella se configura el menú principal. Aparecerán las distintas opciones por las que navegar para visualizar los datos recogidos por los sensores. Además, en ella se podrá activar/desactivar tanto el riego manual como el automático.

```

if (page == 1) {
    myOLED.clrScr();
    myOLED.invertText(true);
    myOLED.print("      MENU      ", CENTER, 0);
    myOLED.invertText(false);
    myOLED.print("RIEGO AUTO: ", 10, 13);
    myOLED.print("RIEGO MANUAL: ", 10, 27);
    myOLED.print("DATOS ACTUALES", 10, 41);
    myOLED.print("ESTADO DEPOSITO", 10, 55);
    //Indicador de ON/OFF del modo automático
    if (autoON == true && page == 1)
        myOLED.print(" ON", 85, 13);
    if (autoON == false && page == 1)
        myOLED.print(" OFF", 85, 13);
    //Indicador de ON/OFF del modo manual
    if (releON == true && page == 1)
        myOLED.print(" ON", 85, 27);
    if (releON == false && page == 1)
        myOLED.print(" OFF", 85, 27);
    //Situación del cursor indicador
    myOLED.print(">", 0, i);
    myOLED.update();
    down(4); //Navegar por 4 opciones distintas
    riegoManual(); //Llamada a la función de riego manual
}

```

Figura 49. Función loop(): configuración de página 1.

- Página 2: si estando en la página 1 se selecciona la opción “ESTADO DEPÓSITO”, se accede a esta página. En esta parte del código, se llama a las funciones *nivelAgua()* y *servoDep()* con las que se podrá visualizar en la pantalla el nivel del depósito del agua y abrir/cerrar el mismo mediante la selección de una de las dos opciones.

```

if (page == 2) {
    nivelAgua();
    servoDep();
    myOLED.clrScr();
    myOLED.invertText(true);
    myOLED.print("      ESTADO DEPOSITO      ", CENTER, 0);
    myOLED.invertText(false);
    myOLED.print("Nivel de agua: ", 10, 15);
    myOLED.printNumI(nivel, 106, 15);
    myOLED.print("%", 124, 15);
    myOLED.print("Abrir deposito", 10, 31);
    myOLED.print("Cerrar deposito", 10, 47);
    //Al entrar en la pág. 2 el cursor se sitúa en la posición y = 15+16:
    myOLED.print(">", 0, i + 16);
    myOLED.update();
    down(2); //En esta pág. las opciones seleccionables son 2
}

```

Figura 50. Función loop(): configuración de página 2.

- Página 3: si en la página 1 se selecciona la opción “DATOS ACTUALES”, accederemos a estas instrucciones (Figura 51). Aquí se llama a las funciones *leerDHT()* y *sensorHum()*, que son las encargadas de leer los datos de humedad y temperatura del ambiente y humedad del suelo. Se mostrarán dichos valores como un porcentaje.

```

if (page == 3 ) {

  leerDHT();
  sensorHum();
  myOLED.clrScr();
  myOLED.invertText(true);
  myOLED.print("  DATOS ACTUALES  ", CENTER, 0);
  myOLED.invertText(false);
  myOLED.print("T AMBIENTE: ", 10, 15);
  myOLED.printNumI(temp, 82, 15);
  myOLED.print("°C", 100, 15);
  myOLED.print("HUMEDAD AMB: ", 10, 31);
  myOLED.printNumI(humDHT, 86, 31);
  myOLED.print("%", 106, 31);
  myOLED.print("HUM TIERRA: ", 10, 47);
  myOLED.printNumI(humedad, 82, 47);
  myOLED.print("%", 106, 47);
  myOLED.print("->", 115, 55); //Se imprime una flecha -> para indicar la siguiente pág.
  myOLED.update();
}

```

Figura 51. Función *loop()*: configuración de página 3.

- Página 4: desde la página 3 y pulsando el botón “SIGUIENTE”, se accederá a esta página. En ella se llama a la función *pagina4()*, donde se han programado una serie de mensajes que saldrán por pantalla siempre que la lectura de los distintos sensores sea la correspondiente a cada estado de la instalación. También se ha configurado para que salga por pantalla el tiempo que queda hasta el próximo riego automático.

```

if (page == 4) {
  myOLED.clrScr();
  myOLED.invertText(true);
  myOLED.print("  DATOS ACTUALES  ", CENTER, 0);
  myOLED.invertText(false);
  pagina4(); //Llamada a la función que define la pág. 4
  myOLED.print("<-", 10, 55); //Se imprime una flecha -> para indicar pág. anterior
  myOLED.update();
}

```

Figura 52. Función *loop()*: configuración de página 4.

```

void pagina4() {
  byte second, minute, hour, dayOfWeek, day, month, year;
  readDS3231time(&second, &minute, &hour, &dayOfWeek, &day, &month, &year);
  sensorHum();
  nivelAgua();

  if (humedad <= 50) myOLED.print("Suelo seco.", 10, 15);
  else if (50 < humedad <= 80) myOLED.print("Suelo humedo.", 10, 15);
  else if (humedad > 80) myOLED.print("Exceso de humedad.", 10, 15);

  if (nivel <= 30) myOLED.print("Deposito vacio.", 10, 31);
  else if (30 < nivel <= 60) myOLED.print("Nivel bajo. Llenar.", 10, 31);
  else if (nivel > 60) myOLED.print("Deposito lleno.", 10, 31);

  if ((hour * 60 + minute > h_riego1 * 60 + m_riego1) && (hour * 60 + minute < h_riego2 * 60 + m_riego2))
    prox_riego = h_riego2 * 60 + m_riego2 - (hour * 60 + minute);
  else if (hour * 60 + minute > h_riego2 * 60 + m_riego2)
    prox_riego = 24 * 60 - (hour * 60 + minute) + (h_riego1 * 60 + m_riego1);
  else
    prox_riego = h_riego1 * 60 + m_riego1 - (hour * 60 + minute);

  myOLED.print("Riego en ", 23, 47);
  if (prox_riego % 60 < 10) {
    myOLED.print("0", 79, 47);
    myOLED.printNumI(prox_riego / 60, 85, 47);
  }
  else
    myOLED.printNumI(prox_riego / 60, 79, 47);
  myOLED.print(":", 91, 47);
  if (prox_riego % 60 < 10) {
    myOLED.print("0", 97, 47);
    myOLED.printNumI(prox_riego % 60, 103, 47);
  }
  else
    myOLED.printNumI(prox_riego % 60, 97, 47);
}

```

Figura 53. Función pagina4().

### 3.2.4 Función readDS3231time()

```

void readDS3231time(byte *second, byte *minute, byte *hour, byte *dayOfWeek, byte *dayOfMonth, byte *month, byte *year) {
  Wire.beginTransmission(DS3231_I2C_ADDRESS); //Empieza transmisión a RTC
  Wire.write(0); //El puntero de registro se pone a 0 para recibir los segundos
  Wire.endTransmission(); //Termina la transmisión
  Wire.requestFrom(DS3231_I2C_ADDRESS, 7); //Se piden 7 bytes del RTC
  *second = bcdToDec(Wire.read() & 0x7f); //Read devuelve el último byte transmitido del RTC
  *minute = bcdToDec(Wire.read()); //Cuando el RTC recibe el OK de que el byte se ha mandado
  *hour = bcdToDec(Wire.read() & 0x3f); //bien avanza el puntero hasta el siguiente y lo envía
  *dayOfWeek = bcdToDec(Wire.read()); //repetiendo el proceso para los 7 bytes.
  *dayOfMonth = bcdToDec(Wire.read()); //0x7f y 0x3f son filtros de dirección
  *month = bcdToDec(Wire.read()); //Como el RTC da los datos en formato BCD
  *year = bcdToDec(Wire.read()); //se pasan a decimal con la función bcdToDec
}

```

Figura 54. Función readDS3231time(). [34]

### 3.2.5 Función ok()

Esta función, define el funcionamiento del pulsador “OK” con el que se accede a las distintas opciones en el menú de la pantalla OLED. Es una función de tipo boolean que devolverá un 1 cuando esté pulsado el botón y un 0 cuando deje de pulsarse.

```

boolean ok() {
    if (y == 0)
        selecOk = 0;
    if (y == 1 && selecOk == 0) {
        selecOk = 1;
        return true;
    }
    return false;
}

```

Figura 55. Función ok().

### 3.2.6 Función *back()*

Esta función, establece el funcionamiento del botón “atrás”. Ejecutándola se podrá retroceder dentro del menú de una pantalla a la anterior. En función de la pantalla en la que se active y a la pantalla a la que vuelva, situará el cursor en una posición u otra. Además, se han limitado el número de pulsaciones de este botón para que la última página a la que pueda acceder sea la página 0.

```

void back() {
    if (z == 0)
        selecBack = 0;
    if (z == 1 && selecBack == 0) {
        i = 13;
        switch (page) {
            case 3:
                page = 1;
                break;
            default:
                if (page > 0)
                    page = page - 1;
        }
        selecBack = 1;
    }
}

```

Figura 56. Función back().

### 3.2.7 Función *down()*

La función *down()* permitirá navegar por las distintas opciones de cada página con el pulsador “siguiente”. Con la variable “n”, se define el número de opciones en cada caso para que, cuando llegue a la última de ellas, vuelva a situar el cursor en la primera.

```

void down(int n) {
    if (x == 0)
        selecDown = 0;

    if (x == 1 && selecDown == 0) {
        selecDown = 1;
        myOLED.clrScr();
        myOLED.update();

        if (page == 1) {
            i = i + 14;
            if (i > 13 + (n - 1) * 14)
                i = 13;
        }
        else {
            i = i + 16;
            if (i > 15 + (n - 1) * 16)
                i = 15;
        }
    }
}

```

Figura 57. Función down().

### 3.2.8 Función sleep()

Esta función gestiona el encendido y el apagado de la pantalla mediante el pulsador “ON/OFF”. En cualquier momento de la ejecución del programa se puede encender o apagar la pantalla, con lo que se producirá un menor consumo de corriente en la instalación.

```

void sleep() {
    if (w == 0)
        selecSleep = 0;

    if (w == 1 && selecSleep == 0) {
        selecSleep = 1;
        if (currentSleep == true) {
            myOLED.sleepMode(SLEEP_OFF);
            currentSleep = false;
        }
        else {
            myOLED.sleepMode(SLEEP_ON);
            currentSleep = true;
            page = 0;
        }
        myOLED.update();
    }
}

```

Figura 58. Función sleep().



### 3.2.9 Función leerDHT()

Cuando se llama a esta función durante la ejecución del programa, se accede al valor leído por el sensor de humedad DHT11. Con él, se obtienen los valores de temperatura y humedad del ambiente.

```
void leerDHT() {  
  humDHT = dht.readHumidity();  
  temp = dht.readTemperature();  
}
```

Figura 59. Función sensorDHT().

### 3.2.10 Función sensorHum()

A través de esta función, los sensores de humedad del suelo leen el valor de esta y mediante la función `map()` se realiza una transformación del valor a un porcentaje entre 0 y 100.

Como se tienen dos sensores, se suman ambas medidas y se dividen entre dos para obtener la humedad media del terreno.

Además, cuando la humedad es menor del 50%, se enciende un LED amarillo que avisará al usuario de que es necesario regar.

```
void sensorHum() {  
  int valorRes = 0;  
  int valorCap = 0;  
  valorRes = map(analogRead(sensorRes), 340, 1023, 100, 0);  
  valorCap = map(analogRead(sensorCap), 200, 523, 100, 0);  
  humedad = (valorRes + valorCap) / 2;  
  if (humedad < 0) humedad = 0;  
  if (humedad > 100) humedad = 100;  
  if (humedad < 50)  
    digitalWrite(yellLED, HIGH);  
  else digitalWrite(yellLED, LOW);  
}
```

Figura 60. Función sensorHum().

### 3.2.11 Función nivelAgua()

Esta función obtiene el valor de humedad medido por el sensor de nivel de agua y lo transforma en un porcentaje entre 0 y 100% mediante la función `map()`.

Además, si el depósito tiene un nivel de agua menor del 30%, se encenderá un LED rojo que indicará al usuario que necesita rellenarse.

```
void nivelAgua() {
  nivel = map(analogRead(sensorHW), 250, 580, 0, 100);
  if (nivel < 0) nivel = 0;
  if (nivel > 100) nivel = 100;
  if (nivel < 30)
    digitalWrite(redLED , HIGH);
  else digitalWrite(redLED , LOW);
}
```

Figura 61. Función nivelAgua().

### 3.2.12 Función servoDep()

Esta función se ha definido para poder simular la apertura y el cierre de un hipotético depósito de agua que se rellenaría de forma manual abriendo o cerrando un grifo o una electroválvula.

```
void servoDep() {
  if (i == 15 && page == 2 && ok()) {
    servo.write(0); //abrir depósito
  }
  else if (i == 31 && page == 2 && ok()) {
    servo.write(180); // cerrar depósito
  }
}
```

Figura 62. Función servoDep().

### 3.2.13 Función riegoManual()

Como se ha visto anteriormente, en la página 1 existe una opción llamada “RIEGO MANUAL” que se puede activar mediante el pulsador “OK” cuando el cursor “>” está situado sobre ella. Cambiando su estado a “ON”, se activará el relé que regula el funcionamiento de la bomba, activando de esta forma el riego.

	ESTADO	ESTADO LÓGICO
RELÉ	Encendido	LOW
	Apagado	HIGH
LED	Encendido	HIGH
	Apagado	LOW

Tabla 8. Estado de activación de relé y LED.

Además, cuando se está regando, se encenderá un LED verde para indicarlo.

```
void riegoManual() {
  if (i == 27 && page == 1 && ok()) {
    if (releON == false) {
      digitalWrite(bomba, LOW);
      digitalWrite(greenLED , HIGH);
      releON = true;
    }
    else {
      digitalWrite(bomba, HIGH);
      digitalWrite(greenLED , LOW);
      releON = false;
    }
  }
}
```

Figura 63. Función riegoManual().

### 3.2.14 Función lecturas()

Para limitar la ejecución del programa y evitar que los sensores estén leyendo información todo el rato y minimizar su deterioro, se ha programado esta función que mandará la señal de lectura cada 5 minutos.

Así, siempre y cuando no se acceda a alguna de las opciones que utilizan los datos leídos por los sensores, las lecturas se realizaran un número limitado de veces.

Para ello se ha utilizado el comando *millis()*, que se usa para medir el tiempo que ha pasado entre dos o más procesos.

```
void lecturas() {
  currentMillis = millis();
  if ((currentMillis - previousMillis) > 300000) {
    previousMillis = currentMillis;
    leerDHT();
    sensorHum();
    nivelAgua();
  }
}
```

Figura 64. Función lecturas().

### 3.2.15 Función RiegoAuto()

Por último, se ha definido la función principal del proyecto: la que va a gestionar el riego automático.

En primer lugar, para que este quede del todo configurado, se debe activar en la página 1 del menú de la pantalla OLED. Su estado cambiará de “OFF” a “ON” al pulsar sobre esta opción con el pulsador “OK”.

Una vez activado, se ejecutará a las horas establecidas mediante las variables h\_riego1, m\_riego1, h\_riego2 y m\_riego2 siempre y cuando sea necesario.

Para saber si se tiene que regar, se comprobarán los datos leídos por los diferentes sensores y se seguirán las instrucciones reflejadas a continuación:

- Si el nivel de humedad del suelo está por debajo del 75% y el nivel del depósito de agua está por encima del 30%: se regará en las horas establecidas.
- Si el nivel de humedad del suelo está por encima del 75% : no será necesario regar y, por tanto, si llega la hora programada el riego no se activará.
- Si se está regando, han pasado más de 3 minutos y la humedad supera el 75%, el riego parará.
- Si se está regando y el depósito baja del 30% de su capacidad, el riego también parará.
- Si el depósito baja del 30%, se rellenará de forma automática (en este caso, el servomotor se activará simulando ese llenado).

Todos estos valores, podrán ser modificados por el usuario en función de las necesidades de cada instalación.

```

void RiegoAuto() {
  byte second, minute, hour, dayOfWeek, day, month, year;
  readDS3231time(&second, &minute, &hour, &dayOfWeek, &day, &month, &year);

  if (i == 13 && page == 1 && ok()) {
    if (autoON == false)
      autoON = true;
    else
      autoON = false;
  }

  if (autoON == true) {
    if (regando == true) {
      sensorHum();
      nivelAgua();
    }

    if ((hour == h_riego1 && minute == m_riego1 || hour == h_riego2 && minute == m_riego2) && ~regando ) {
      sensorHum();
      nivelAgua();
      if (humedad < 75 && nivel > 30) {
        digitalWrite(bomba, LOW);
        digitalWrite(greenLED , HIGH);
        regando = true;
        autoMillis = millis();
      }
    }

    if ((( millis() - autoMillis > 180000) && regando && humedad > 75) || nivel < 30) {
      digitalWrite(bomba, HIGH);
      digitalWrite(greenLED , LOW);
      regando = false;
    }

    if (nivel < 30)
      servo.write(0);
    if (nivel > 90)
      servo.write(180);
  }
}

```

Figura 65. Función RiegoAuto().

### 3.3. ESPACIO EN MEMORIA

Como se indicó anteriormente, la memoria de la tarjeta Arduino UNO R3 se divide en distintas partes (Tabla 9). En la Figura 66 puede verse que, una vez compilado el código, se ha utilizado un 47% de la memoria Flash y un 41% de la memoria SRAM.

MEMORIA	DESCRIPCIÓN	CAPACIDAD
Flash	Destinada a almacenar el sketch.	32kB
SRAM	Almacena las variables declaradas en el sketch.	2kB
EEPROM	Almacena los datos que se quieren conservar ante posibles reinicios o fallos de alimentación.	1kB

Tabla 9. Memoria Arduino UNO R3.

Compilado
El Sketch usa 15400 bytes (47%) del espacio de almacenamiento de programa. El máximo es 32256 bytes. Las variables Globales usan 841 bytes (41%) de la memoria dinámica, dejando 1207 bytes para las variables locales. El máximo es 2048 bytes.

Figura 66. Espacio en memoria. [13]

## CAPÍTULO 4: DISEÑO

### 4.1. MAQUETA CONSTRUIDA

Para poder visualizar todo el proceso desarrollado a lo largo del proyecto, se ha construido una sencilla maqueta que permitirá probar el código programado y realizar una simulación de la instalación.

Como base, se ha utilizado una bandeja metálica de 50x29.5cm que contendrá la tierra y las plantas utilizadas para representar un pequeño jardín. Además, en uno de sus extremos se ha construido una caseta de cartón que contendrá toda la instalación eléctrica y electrónica. En la Figura 67 puede verse una vista general de la maqueta.



Figura 67. Maqueta construida.

En la parte frontal de dicha caseta, se ha colocado la pantalla OLED junto con los pulsadores y los LED indicadores de forma que se puedan visualizar y manipular fácilmente. En uno de los laterales, se ha colocado el sensor DHT11 para que detecte la temperatura y la humedad del ambiente (Figura 68).



Figura 68. Parte frontal (izquierda) y sensor DHT11 (derecha).

Como se ha mencionado a lo largo de la memoria, cada pulsador tiene una función diferente asignada y esta se ha reflejado en la Figura 69. Pulsadores de la interfaz..

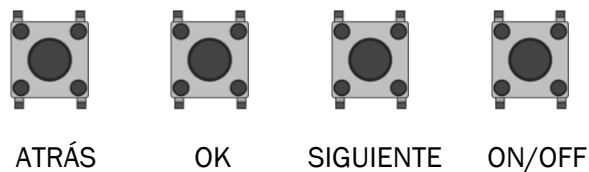


Figura 69. Pulsadores de la interfaz.

El servomotor que simula la apertura y el cierre de la electroválvula que llenaría el depósito en un caso real, también se ha colocado en la parte frontal para poder visualizar claramente su funcionamiento (Figura 70).



Figura 70. Servomotor.

La parte trasera (Figura 71) se ha dejado abierta para poder acceder a la tarjeta Arduino y a los distintos componentes que forman la instalación. Se han realizado dos agujeros en la parte inferior para poder llevar los cables de los sensores de humedad del suelo desde la tierra hasta la caseta.

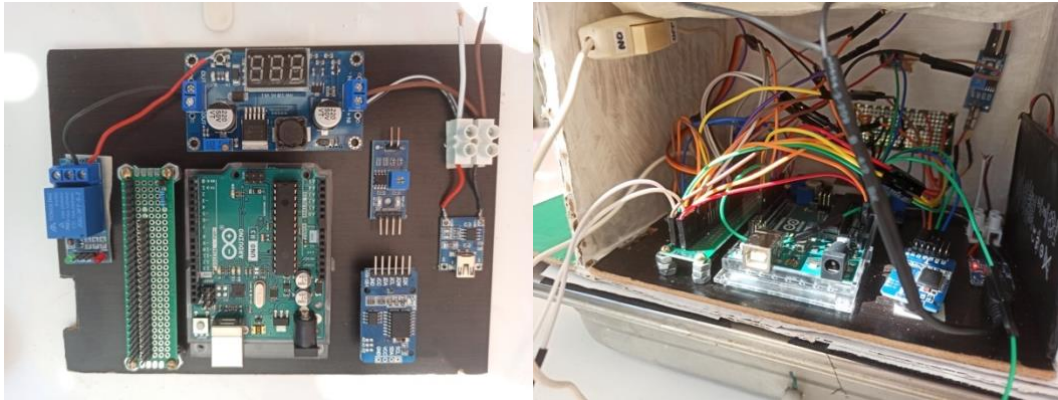


Figura 71. Instalación de control.

En la parte superior, se ha colocado una especie de tejadillo en el que se han instalado las placas solares que cargan la batería de alimentación de la bomba de agua para conseguir que incida sobre ellas una mayor cantidad de luz (Figura 72).



Figura 72. Paneles solares.

Por último, se ha utilizado una caja de plástico como depósito en la que se ha colocado la bomba de agua y el sensor de nivel (Figura 73). En esta caja se ha realizado un pequeño agujero para poder llevar los cables de conexión del sensor y la bomba hasta la tarjeta Arduino. Además, se ha realizado un orificio que permite llevar la manguera desde la bomba hasta el circuito de riego instalado en la bandeja. De esta forma, la bomba tomará el agua del depósito y la impulsará a través de la tubería con cierta presión hasta el circuito de riego por goteo. En un hipotético caso real, el depósito se llenaría desde la red mediante la apertura y el cierre de una electroválvula.





Figura 73. Depósito de agua con bomba y sensor de nivel.

Para construir el sistema de riego, se ha utilizado una tubería de 5mm de diámetro que sale de la bomba de agua hasta un enlace de 4 vías que distribuirá el agua entre las distintas ramas del riego por goteo.



Figura 74. Conexiones de tubería y goteros.

Como emisores, se han utilizado unos aspersores de riego por goteo en línea, cuyo caudal se puede regular entre 0 y 10 litros por hora.



Figura 75. Sistema de riego por goteo. Tubería, enlace de 4 vías y aspersor de riego por goteo.

## 4.2. MANUAL DE FUNCIONAMIENTO

Utilizar este sistema de riego es muy sencillo. A continuación, se describe paso a paso el funcionamiento de este.

Lo primero que hay que hacer, es asegurarse de que los parámetros de referencia establecidos en el código programado son los que se quieren tener en cuenta. Para modificar estos valores, se debe acceder a la plataforma de programación de la tarjeta *Arduino Genuino* a través de un ordenador. Una vez abierto el código, se podrán visualizar y modificar los diferentes parámetros establecidos tales como las horas de riego o los límites de humedad y de nivel del agua en el depósito.

Una vez establecidos los parámetros que deben tener en cuenta los sensores y actuadores de la instalación y después de haber cargado el código en la tarjeta Arduino, la interfaz está lista para usarse.

En primer lugar, habrá que encender la pantalla mediante el pulsador “ON/OFF”, ya que por defecto el sistema se inicia con ella apagada. Aparecerán los datos tal y como se muestra en la Tabla 10 (Página principal), donde se muestra la hora y la fecha actuales. En esa misma tabla, se representan las diferentes pantallas de navegación del sistema.

Para acceder al menú principal, hay que pulsar el botón “OK”. Una vez en él, se pueden visualizar cuatro opciones distintas, por las que se podrá navegar mediante los pulsadores de “atrás”, “OK” y “siguiente”.

- “RIEGO AUTO: OFF/ON”: en función de si la pantalla muestra ON o muestra OFF, el riego automático estará o no activado. El estado de dicho riego puede cambiarse situando el cursor “>” sobre esta opción y pulsando el botón “OK”. En función de los datos leídos por los distintos sensores y de

las horas de riego programadas, comenzará a regar o esperará hasta que se cumplan las condiciones establecidas en el código del programa.

- “RIEGO MANUAL: OFF/ON”: el estado de este modo de riego funciona exactamente igual que el riego automático pero, en este caso, el riego se encenderá directamente sin tener en cuenta ninguna lectura de los sensores.
- “ESTADO ACTUAL”: situando el cursor en esta opción y pulsando “OK” se puede acceder a los datos de temperatura y humedad leídos por los sensores. Además, pasando a la siguiente página se puede ver un resumen del estado de la instalación mediante una serie de mensajes que irán variando la información en función de las condiciones en las que se encuentra el sistema.
- “ESTADO DEPÓSITO”: en esta última opción, se visualiza el nivel de agua medido por el sensor. Se accede a ella de la misma forma que a las opciones anteriores: situando el cursor “>” sobre ella y pulsando “OK”. Una vez dentro, se podrá elegir también si abrir o cerrar el depósito para que se llene, lo que se simula mediante el movimiento de la hélice del servomotor.






ASPECTO DE LA PANTALLA	DESCRIPCIÓN
	<p><b>Página principal.</b> En ella se muestran la fecha y hora actuales. Además, aparece la palabra “MENU”, al que se podrá acceder pulsando el botón “OK”. Se ha establecido como página por defecto y aparecerá cada vez que se encienda la pantalla.</p>
	<p><b>Menú principal.</b> Muestra las distintas opciones por las que se puede navegar. Mediante el cursor “&gt;”, se indica la opción seleccionada, a la que se podrá acceder mediante el pulsador “OK”. Para navegar por el menú, basta con ir presionando el pulsador “SIGUIENTE”.</p>
	<p><b>Página 1 de “DATOS ACTUALES”.</b> En ella se muestra la temperatura y humedad ambiente, medidas por el sensor DHT11, y la humedad del suelo. En la parte inferior derecha aparece una flecha que indica la existencia de otra página en la misma opción.</p>
	<p><b>Página 2 de “DATOS ACTUALES”.</b> En ella se muestra un resumen del estado de la instalación, que irá variando en función de los datos medidos por los sensores. Además, se muestra el tiempo que falta hasta el próximo riego automático programado.</p>
	<p>Opción “<b>ESTADO DEPÓSITO</b>”. Aquí se puede visualizar el nivel de agua que queda en el depósito. Además, se puede abrir y cerrar de forma manual para poder iniciar o terminar el llenado de este.</p>

Tabla 10. Interfaz de la pantalla OLED. [35]

### 4.3. COSTE DEL EQUIPO

A continuación, se recoge el coste total aproximado del sistema implementado. En la Tabla 11 se desglosan de forma individual los costes referidos a la adquisición de todos los componentes. En el caso del presente proyecto, muchos de ellos no ha habido que comprarlos nuevos porque se han podido reciclar de otros proyectos.

La envolvente y los materiales utilizados en la construcción de la maqueta no se han valorado, ya que se ha construido con materiales reciclados o elementos ya adquiridos para otros fines y, además, representan el terreno y el espacio de trabajo ya existentes en un caso real.

DESCRIPCIÓN		Ud.	€/Ud.	TOTAL
<b>CAPÍTULO 01 ARDUINO</b>				
<b>01.1</b>	<b>Ud.</b>	<b>Arduino UNO R3</b>		
		1	20,00€	20,00€
<b>01.2</b>	<b>Ud.</b>	<b>Cable de conexión</b>		
		1	6,00 €	6,00€
<b>TOTAL CAPÍTULO 01</b>				<b>26,00 €</b>
<b>CAPÍTULO 02 SENSORES</b>				
<b>02.1</b>	<b>Ud.</b>	<b>Sensor DHT11</b>		
		1	2,80 €	2,80 €
<b>02.2</b>	<b>Ud.</b>	<b>Higrómetro FC-28</b>		
		1	2,60 €	2,60 €
<b>02.3</b>	<b>Ud.</b>	<b>Higrómetro capacitivo</b>		
		1	3,50 €	3,50 €
<b>02.4</b>	<b>Ud.</b>	<b>Detector de presencia de agua</b>		
		1	1,60 €	1,60 €
<b>02.5</b>	<b>Ud.</b>	<b>Pulsadores</b>		
		4	0,10 €	0,40 €
<b>TOTAL CAPÍTULO 02</b>				<b>10,90 €</b>
<b>CAPÍTULO 03 ACTUADORES Y VISUALIZADORES</b>				
<b>03.1</b>	<b>Ud.</b>	<b>Pantalla OLED</b>		
		1	4,98 €	4,98 €
<b>03.2</b>	<b>Ud.</b>	<b>Servomotor SG90</b>		
		1	2,60 €	2,60 €
<b>03.3</b>	<b>Ud.</b>	<b>Relé 5V KY-09</b>		
		1	2,50 €	2,50 €
<b>03.4</b>	<b>Ud.</b>	<b>Bomba de agua</b>		
		1	4,00 €	4,00 €
<b>03.5</b>	<b>Ud.</b>	<b>Diodo LED</b>		
		1	0,15 €	0,15 €
<b>03.6</b>	<b>Ud.</b>	<b>Diodo LED</b>		
		1	0,15 €	0,15 €
<b>03.7</b>	<b>Ud.</b>	<b>Diodo LED</b>		

Emisión de color y lente: amarillo. Tamaño: 5x9mm. Voltaje de funcionamiento: 1.8-2.1V. 100000 h.		1	0,15 €	0,15 €
<b>03.8</b>	<b>Ud.</b>	<b>Interruptor</b>		
Interruptor de paso blanco unipolar.		1	3,20 €	3,20 €
<b>TOTAL CAPÍTULO 03</b>				<b>17,73 €</b>
<b>CAPÍTULO 04 ALIMENTACIÓN</b>				
<b>04.1</b>	<b>Ud.</b>	<b>Batería</b>		
Batería Li-ion de 3.8V, 6500mAh.		1	5,00 €	5,00 €
<b>04.2</b>	<b>Ud.</b>	<b>Módulo TP4056</b>		
Controlador de carga Li-ion. Conector mini-USB 5V 1A.		1	1,25 €	1,25 €
<b>04.3</b>	<b>Ud.</b>	<b>Convertidor step-up DC/DC</b>		
Adaptador de fuente de alimentación. Convertidor elevador. Tensión de entrada: 2-24V. Tensión de salida: 5-28V.		1	3,00 €	3,00 €
<b>04.4</b>	<b>Ud.</b>	<b>Panel solar</b>		
Mini placas solares 3V 0.3W. Tamaño: 65x48mm.		6	2,00 €	12,00 €
<b>04.5</b>	<b>Ud.</b>	<b>Diodo rectificador</b>		
Modelo 1N4007. Permite el flujo de corriente en una sola dirección. Para circuitos con menos de 1A.		1	0,11 €	0,11 €
<b>04.6</b>	<b>Ud.</b>	<b>Fuente de alimentación</b>		
Adaptador de corriente AC/DC de 9V y 0.5A. Conector 5.5x2.1mm, con polo positivo en el interior.		1	7 €	7 €
<b>TOTAL CAPÍTULO 04</b>				<b>28,36 €</b>
<b>CAPÍTULO 05 RELOJ DE TIEMPO REAL</b>				
<b>05.1</b>	<b>Ud.</b>	<b>Módulo RTC</b>		
Reloj de precisión en tiempo real DS3231 I2C. Batería 3V incluida.		1	4,69 €	4,69 €
<b>TOTAL CAPÍTULO 05</b>				<b>4,69 €</b>
<b>CAPÍTULO 06 OTROS</b>				
<b>06.1</b>	<b>Ud.</b>	<b>Resistencia</b>		
Resistencia con montaje de orificio pasante de 220Ω.		3	0,07 €	0,21 €
<b>06.2</b>	<b>Ud.</b>	<b>Resistencia</b>		
Resistencia con montaje de orificio pasante de 5.1kΩ.		1	0,08 €	0,08 €
<b>06.3</b>	<b>Ud.</b>	<b>Resistencia</b>		
Resistencia con montaje de orificio pasante de 10kΩ.		4	0,10 €	0,40 €
<b>06.5</b>	<b>-</b>	<b>Cables de conexión</b>		
Cables puente, macho-hembra, macho-macho, hembra-hembra.		1	3,00 €	3,00 €
<b>06.6</b>	<b>Ud.</b>	<b>Placa PCB</b>		
Placa de circuito impreso de doble cara.		2	0,80 €	1,60 €
<b>TOTAL CAPÍTULO 06</b>				<b>5,29 €</b>
<b>CAPÍTULO 07 SISTEMA DE RIEGO</b>				
<b>07.1</b>	<b>Ud.</b>	<b>Aspersor riego por goteo</b>		
Goteros en línea regulables 0-10 l/h.		9	0,40 €	3,60 €
<b>07.2</b>	<b>Ud.</b>	<b>Enlace de 4 vías</b>		
Enlace de 4 vías 1/4" para tubería de Ø4-6mm.		1	0,30 €	0,30 €
<b>07.3</b>	<b>m</b>	<b>Tubería</b>		
Tubería de suministro Ø4-6mm.		3	0,23 €	0,69 €
<b>TOTAL CAPÍTULO 07</b>				<b>4,59 €</b>
<b>CAPÍTULO 08 ENVOLVENTE</b>				
<b>08.1</b>	<b>-</b>	<b>Envolvente</b>		
Material necesario para almacenar el agua y proteger la instalación eléctrica.		-	S/V	S/V
<b>TOTAL CAPÍTULO 08</b>				<b>S/V</b>
<b>TOTAL</b>				<b>97,56 €</b>

Tabla 11. Coste del equipo.

## CAPÍTULO 5: OTROS CÁLCULOS

En este capítulo, se reflejan los cálculos no especificados anteriormente, como los consumos de la instalación que determinan el tipo de alimentación necesaria, la composición de los paneles solares o las resistencias de polarización de los diodos LED señalizadores.

Los parámetros calculados cumplen con las exigencias de dicha instalación pero, cabe destacar que, al no haberse implementado en un sistema real y haber reutilizado y reciclado la mayoría de los materiales, los valores han resultado un tanto ajustados.

Aun así, toda la instalación ha sido probada y comprobada para evitar problemas por sobredimensionamiento sin resultar ningún inconveniente en la ejecución, simulación y construcción de esta.

### 5.1. CONSUMOS DE LA INSTALACIÓN

A la hora de elegir las fuentes de alimentación del sistema, se han tenido en cuenta los consumos máximos de cada componente utilizado. En la Tabla 12 se especifica el consumo máximo en mA de cada uno de ellos, obtenido de la hoja de características de cada componente.

DISPOSITIVO	CONSUMO MÁXIMO
Arduino UNO R3	46 mA
Pantalla OLED	6 mA
Sensor DHT11	1 mA
Higrómetro FC-28	35 mA
Sensor capacitivo de humedad	5 mA
Sensor de nivel de agua	15 mA
Servomotor	150 mA
Módulo RTC	0,3 mA
Relé 5V	15 mA
Diodo LED rojo	20 mA
Diodo LED verde	20 mA
Diodo LED amarillo	20 mA
<b>CONSUMO MÁXIMO TOTAL</b>	<b>333,3 mA</b>

Tabla 12. Consumo de cada dispositivo en mA.

Ese consumo de 333,3 mA no se va a dar nunca en la instalación actual, ya que para que esto ocurriera deberían funcionar todos los dispositivos a la vez a máximo rendimiento. La corriente que suministra el adaptador de

corriente utilizado para alimentar la tarjeta Arduino y, por ende, el resto de los componentes, es de 500mA, por lo que podrá abastecer sin ningún problema a toda la instalación.

Por otro lado, según la hoja de características de la bomba de agua, su consumo máximo va a ser de 300mA. La magnitud de la instalación de su alimentación se justifica en el apartado 5.3 de este mismo capítulo, dando como resultado una suficiente corriente de alimentación.

## 5.2. CÁLCULO DE RESISTENCIAS SERIE PARA LED

Se han utilizado tres diodos LED para indicar distintos estados de la instalación. Para poder conectarlos y utilizarlos con seguridad, se han utilizado tres resistencias que limitan la corriente que pasa por cada uno de ellos.

En el caso de aplicar un valor de corriente superior al recomendado, la vida útil del LED se puede ver afectada o hasta puede llegar a inutilizarlo totalmente.

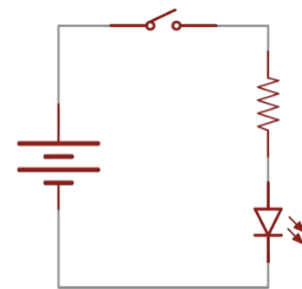


Figura 76. Circuito eléctrico diodo LED.

Para calcular dichas resistencias se ha recogido en la Tabla 13 las características eléctricas necesarias de cada uno de los LED. Además, se indica una breve descripción de las condiciones de activación de cada diodo.

Aplicando la Ley de Ohm (Ecuación 1) y teniendo en cuenta que la tensión de la fuente ( $V_f$ ) es de 5V, se obtiene el valor de la resistencia necesaria.

$$V = I \cdot R \rightarrow R = \frac{V}{I} = \frac{V_f - V_L}{I}$$

Ecuación 1. Ley de Ohm.

LED	QUÉ INDICA	CORRIENTE NOMINAL	TENSIÓN DE FUNCIONAMIENTO		RESISTENCIA TEÓRICA MÍNIMA	RESISTENCIA UTILIZADA
			Mín.	Máx.		
Rojo	Depósito vacío	20mA	1.9V	2.2V	140Ω	220Ω
Verde	Regando	20mA	2.2V	2.6V	120Ω	220Ω
Amarillo	Hace falta regar	20mA	1.9V	2.2V	140Ω	220Ω

Tabla 13. Características diodos LED.

Una vez calculadas dichas resistencias, se ha optado por elegir unas de mayor valor (de 220Ω) para evitar cualquier problema en caso de fallos de corriente y tensión. Además, aumentar dicho valor en las resistencias no afecta en el funcionamiento del circuito, únicamente puede variar ligeramente la



intensidad de brillo de cada diodo, pero se logrará aportar mayor seguridad y tiempo de vida a los diodos LED. La corriente que circulará, por tanto, con ese nuevo valor de resistencia por cada diodo LED será de aproximadamente 14mA, lo que es suficiente para que se iluminen.

### 5.3. SISTEMA DE ALIMENTACIÓN FOTOVOLTAICO

Para conseguir una tensión y corriente necesarias para cargar la batería y alimentar la bomba de agua, se deben configurar las placas solares de forma que cumplan con las necesidades mínimas.

Por ello, se ha realizado una conexión serie/paralelo combinando las seis placas disponibles. Un solo panel proporciona una tensión de 3V y una potencia de 0.3W. Por tanto, cuando esté funcionando a pleno rendimiento, proporcionará una corriente de 100mA (Ecuación 2).

$$P = V \cdot I \rightarrow I = \frac{P}{V} = \frac{0.3W}{3V} = 0.1A = 100mA$$

*Ecuación 2. Corriente proporcionada por un panel solar.*

Todos los paneles disponibles tienen las mismas características, resumidas en la Tabla 14. Por tanto, para poder cubrir tanto la carga de la batería como la alimentación de la bomba de agua, se deben aumentar la tensión y la corriente suministradas.

PANEL SOLAR	
Potencia máxima	0,3W
Voltaje a potencia máxima	3V
Intensidad a potencia máxima	100mA
Dimensiones	65x48mm

*Tabla 14. Características de un panel solar.*

En primer lugar, para aumentar la tensión de salida se han colocado dos placas en serie de forma que suministrarán una tensión final de 6V, ya que al conectar dos tensiones en serie dan como resultado la suma de ambas. En este caso, la corriente que suministran seguirá siendo de 100mA (Ecuación 3).

$$V_{total\_serie} = V_1 + V_2 = 3 + 3 = 6V$$

$$I_{total\_serie} = I_1 = I_2 = 100mA$$

*Ecuación 3. Tensión y corriente en serie.*

Al necesitar también aumentar la corriente, se conectarán ahora las placas en paralelo. De la operación anterior se han obtenido tres grupos de dos placas en serie cada uno, los cuales se van a conectar ahora en paralelo tal y como muestra la Figura 77.

De esta forma, se obtiene una tensión y una corriente resultantes de 6V y 300mA, respectivamente (Ecuación 4).

$$V_{total\text{paralelo}} = V_1 = V_2 = V_3 = 6V$$

$$I_{total\text{paralelo}} = I_1 + I_2 + I_3 = 100 + 100 + 100 = 300mA$$

Ecuación 4. Tensión y corriente en paralelo.

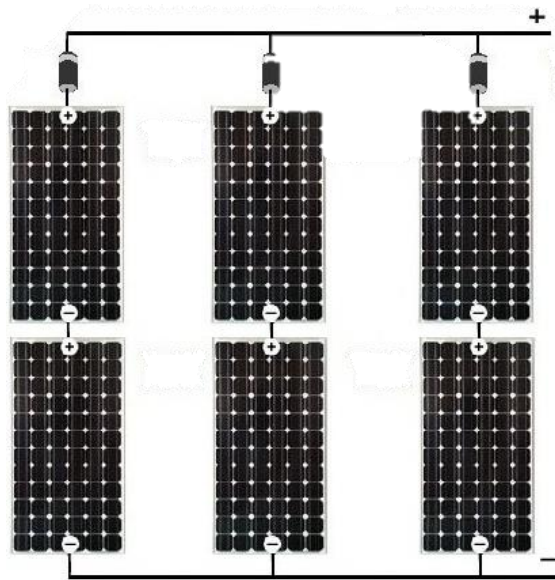


Figura 77. Configuración serie/paralelo de las placas solares. [36]

Como se ha mencionado anteriormente, la bomba de agua tiene un consumo de 300mA. Con los cálculos obtenidos anteriormente, se ha visto que los paneles solares serían capaces de alimentarla por sí solos cuando trabajasen a pleno rendimiento, es decir, cuando la incidencia del sol sobre ellos sea del 100%.

Por tanto, para lograr un mejor rendimiento con los materiales disponibles, el conjunto fotovoltaico va a utilizarse para cargar la batería que alimentará la bomba de agua.

La influencia de la corriente suministrada por las placas sobre la carga de la batería se verá reflejada en el tiempo de carga de esta. Dependiendo de la capacidad de la batería empleada y de la cantidad de sol que incida sobre las

placas, tardará más o menos tiempo en cargarse. En la Tabla 15, se da un tiempo estimado de carga para baterías con diferentes capacidades y con las placas funcionando a pleno rendimiento, calculado de la forma que aparece en la Ecuación 5 y siendo esa corriente suministrada por la alimentación de 300mA.

$$\text{Tiempo de carga} = \frac{\text{Capacidad de la batería}}{\text{Corriente suministrada por la alimentación}}$$

*Ecuación 5. Cálculo del tiempo de carga de una batería.*

CAPACIDAD DE LA BATERÍA	TIEMPO DE CARGA
1100mAh	3,6 horas
1850mAh	6,1 horas
4450mAh	14,8 horas
6500mAh	21,6 horas

*Tabla 15. Tiempos de carga para baterías con distintas capacidades.*

Para lograr una carga más rápida y un funcionamiento más eficiente, deberán utilizarse unas placas solares con mayor capacidad pero, para el presente proyecto, es suficiente con lo expuesto anteriormente, ya que dada la envergadura del TFG y que el principal cometido de este era diseñar única y exclusivamente el sistema de riego, lo que se ha hecho ha sido buscar módulos comerciales que pudieran satisfacer las necesidades sin entrar en fases de diseño y cálculo más avanzadas.

## CONCLUSIONES

### OBJETIVOS CUMPLIDOS

Finalmente se ha conseguido desarrollar el proyecto propuesto de manera satisfactoria y se ha logrado llevar a cabo la idea inicial del trabajo, tratando de hacerlo de la mejor forma posible.

Se ha construido un sistema de riego que puede funcionar tanto de manera manual como de forma autónoma, siendo esta última la idea principal del proyecto.

Tras muchas horas de investigación y documentación, se han estudiado diferentes alternativas que han servido para conocer la gran variedad de dispositivos y posibles vías de desarrollo que pueden formar un mismo sistema, terminando por elegir una serie de elementos que fuesen lo más sencillos y económicos posibles dentro de las exigencias del trabajo, no teniendo por qué ser las mejores ni únicas opciones para llevarlo a cabo pero sí unas opciones válidas que han permitido el desarrollo total de este.

Además, se ha conseguido profundizar en el estudio y conocimiento de los sistemas basados en la herramienta Arduino, la cual proporciona un gran número de posibilidades y oportunidades de desarrollo de proyectos en diferentes ámbitos.

Cabe destacar, también, que al utilizar circuitos auxiliares comerciales, tales como diferentes sensores y sistemas de alimentación, se ha conseguido crear un sistema flexible y modular, capaz de adaptarse a las exigencias requeridas por todos sus posibles usuarios.

### TRABAJO FUTURO

Tras realizar el proyecto, se han valorado una serie de ideas y propuestas para mejorar y hacer más eficiente el presente sistema en una posible implementación futura:

- Incorporación de algún tipo de sistema de monitorización online, ya sea vía web o mediante una aplicación móvil.
- Mejora de los elementos utilizados, como los sensores, para dotar de mayor precisión al sistema y conseguir un óptimo funcionamiento.

- Mejoras en el desarrollo y programación del código, añadiendo la posibilidad de configurar los distintos parámetros sin tener que entrar en el entorno de Arduino IDE.
- Búsqueda de un sistema más eficiente de alimentación basado en energías limpias que satisfaga las necesidades de toda la instalación.
- Desarrollo de algún método de recogida y recirculación del agua para los posibles excedentes que puedan derivar del riego y evitar así desperdiciarla.

## BIBLIOGRAFÍA

- [1] Losada, A., 1997. Glosario de términos. Ingeniería del Agua, 4(4).  
Departamento de Ingeniería Rural. Escuela Técnica superior de Ingenieros Agrónomos. Universidad Politécnica de Madrid. DOI:  
10.4995/ia.1997.2736
- [2] Demim, P., 2014. Aportes para el mejoramiento del manejo de los sistemas de riego. Métodos de riego: fundamentos, usos y adaptaciones. Instituto Nacional de Tecnología Agropecuaria. Ediciones INTA.
- [3] CultivoHidropónico.info, Sistema Hidropónico por Goteo.  
<https://cultivohidroponico.info/sistemas/por-goteo/> [Consultado en Julio 2022].
- [4] agroware, software agrícola. Tipos de riego y sus ventajas. ¿Cuál es el adecuado? <https://sistemaagricola.com.mx/blog/tipos-de-riego-en-la-agricultura-y-ventajas/> [Consultado en Julio 2022].
- [5] Be.green. ¿Qué humedad necesitan las plantas de interior?  
<https://be.green/es/blog/que-humedad-necesitan-las-plantas-de-interior1#toc-2> [Consultado en Julio 2022].
- [6] Torrente Artero, O., 2013. Arduino. Curso práctico de formación. Rc Libros.
- [7] Thorpe, E., 2020. Arduino: Advanced Methods and Strategies of Using Arduino.
- [8] Arduino.cl. n.d. ¿Qué es Arduino? | Arduino.cl. <https://arduino.cl/que-es-arduino/> [Consultado en Julio 2022].
- [9] Luis Llamas. 2016. El bus I2C en Arduino.  
<https://www.luisllamas.es/arduino-i2c/> [Consultado en Julio 2022].
- [10] Aprendiendo Arduino. 2016. Bus I2C/TWI.  
<https://aprendiendoarduino.wordpress.com/2016/11/14/bus-i2ctwi/>  
[Consultado en Julio 2022].
- [11] I2c-bus. 1995. [http://www.i2c-bus.org/fileadmin/ftp/i2c\\_bus\\_specification\\_1995.pdf](http://www.i2c-bus.org/fileadmin/ftp/i2c_bus_specification_1995.pdf) [Consultado en Julio 2022].

- [12] Arduino Official Store. Arduino Uno Rev3.  
<https://docs.arduino.cc/hardware/uno-rev3> [ Consultado en Junio 2022].
- [13] Guerra Carmenate, J., Arduino UNO R3 la revolución del hardware libre. Programar fácil con Arduino. <https://programarfácil.com/blog/arduino-blog/arduino-uno-r3/> [ Consultado en Junio 2022].
- [14] del Valle Hernández, L., 2019. Cómo utilizar el DHT11 para medir la temperatura y humedad con Arduino. Programar fácil con Arduino. <https://programarfácil.com/blog/arduino-blog/sensor-dht11-temperatura-humedad-arduino/> [ Consultado en Junio 2022].
- [15] Llamas, L., 2016. Medir la humedad del suelo con Arduino e higrómetro FC-28. Luis Llamas. <https://www.luisllamas.es/arduino-humedad-suelo-fc-28/> [Consultado en Junio 2022].
- [16] Llamas, L., 2022. Sensor de humedad del suelo capacitivo y Arduino. Luis Llamas. <https://www.luisllamas.es/sensor-de-humedad-del-suelo-capacitivo-y-arduino/> [Consultado en Junio 2022].
- [17] Sensor Detección de Agua para Arduino. Aprendiendo Arduino. <https://aprendiendoarduino.wordpress.com/2018/10/17/sensor-deteccion-de-agua-para-arduino/> [Consultado en Junio 2022].
- [18] Servomotor SG90 RC 9g - UNIT Electronics Arduino Micro Servo. <https://uelectronics.com/producto/servomotor-sg90-rc-9g/> [Consultado en Junio 2022].
- [19] del Valle Hernández, L., Reloj con Arduino, cómo controlar los tiempos con un RTC. Programar fácil con Arduino. <https://programarfácil.com/blog/arduino-blog/reloj-con-arduino-rtc/> [Consultado en Junio 2022].
- [20] Söderby, K., RTC (Real Time Clock) with MKR Zero and OLED Display. Docs.arduino.cc. <https://docs.arduino.cc/tutorials/mkr-zero/mkr-zero-rtc-clock> [Consultado en Junio 2022].
- [21] Vaello Sancho, J., 2022. El Relé. | Formación para la Industria 4.0 | Cursos gratuitos y online para la Industria. Instalador electricista, Automatas y Robótica. <https://automatismoindustrial.com/curso-carnet-instalador-baja-tension/d-automatizacion/1-2-control-de-potencia-aparamenta-electrica/el-rele/> [Consultado en Junio 2022].

- [22] Llamas, L., 2016. Encender una bomba de agua con Arduino. Luis Llamas. <https://www.luisllamas.es/bomba-de-agua-con-arduino/> [Consultado en Junio 2022].
- [23] del Valle Hernández, L., SSD1306 pantalla OLED con Arduino y ESP8266 I2C. Programar fácil con Arduino. <https://programarfácil.com/blog/arduino-blog/ssd1306-pantalla-oled-con-arduino/> [Consultado en Junio 2022].
- [24] G, A., 2019. Push Button con Arduino. Automatización para Todos. <https://www.automatizacionparatodos.com/push-button-con-arduino/> [Consultado en Junio 2022].
- [25] Geek Factory. 2017. Alimentar el Arduino: La guía definitiva - Geek Factory. <https://www.geekfactory.mx/tutoriales-arduino/alimentar-el-arduino-la-guia-definitiva/> [Consultado en Junio 2022].
- [26] Arduino Project Hub. 2021. Arduino con alimentación solar. <https://create.arduino.cc/projecthub/agriculturaelectronica/arduino-con-alimentacion-solar-a78891> [Consultado en Junio 2022].
- [27] Hardware libre. 2021. TP4056: el módulo para cargar baterías. [https://www.hwlibre.com/tp4056/#Todo\\_sobre\\_el\\_TP4056](https://www.hwlibre.com/tp4056/#Todo_sobre_el_TP4056) [Consultado en Junio 2022].
- [28] Arduino.cc. Wire - Arduino Reference. <https://www.arduino.cc/reference/en/language/functions/communication/wire/> [Consultado en Mayo 2022].
- [29] Rinkydinkelectronics.com. 2019. OLED\_I2C - Rinky-Dink Electronics. <http://www.rinkydinkelectronics.com/library.php?id=79> [Consultado en Mayo 2022].
- [30] Adafruit, DHT sensor library - Arduino Reference. Arduino.cc. <https://www.arduino.cc/reference/en/libraries/dht-sensor-library/> [Consultado en Mayo 2022].
- [31] Arduino.cc. Servo - Arduino Reference. <https://www.arduino.cc/reference/en/libraries/servo/> [Consultado en Mayo 2022].



- [32] Adafruit, RTCLib - Arduino Reference. Arduino.cc.  
<https://www.arduino.cc/reference/en/libraries/rtclib/> [Consultado en Mayo 2022].
- [33] Kiryanenko, A., SimpleTimer - Arduino Reference. Arduino.cc.  
<https://www.arduino.cc/reference/en/libraries/simpletimer/> [Consultado en Mayo 2022].
- [34] laurenceHR, GitHub Gist. 2016. RTC & LCD by I2C.  
<https://gist.github.com/laurenceHR/c08cd2a0944df78d168c> [Consultado en Mayo 2022].
- [35] Spannerandcode.co.uk. 2022. Text simulator for Adafruit OLED SSD1306 display. <https://spannerandcode.co.uk/arduino-tools/text-simulator-for-oled-display/> [Consultado en Julio 2022].
- [36] MPPT SOLAR. 2022. Paneles solares: Conexión en Paralelo.  
<https://www.mpptsolar.com/es/paneles-solares-paralelo.html> [Consultado en Mayo 2022].
- [37] Aguado da Costa, J., 2012. Desarrollo de un sistema automatizado para un invernadero. <https://uvadoc.uva.es/handle/10324/2930> [Consultado en 2022].
- [38] Arduino Project Hub. 2020. Greenhouse.  
<https://create.arduino.cc/projecthub/312663/greenhouse-1ec2ee>  
[Consultado en 2022].

## ANEXOS

### DATASHEETS

A continuación, se adjuntas las características más importantes de cada componente utilizado, extraídas de sus correspondientes *datasheets*. Estas hojas de características se incluyen completas en la carpeta de anexos junto con el código programado.

<a href="#">Arduino UNO R3</a> .....
<a href="#">Sensor de humedad DHT11</a> .....
<a href="#">Higrómetro FC-28: Comparador LM393</a> .....
<a href="#">Sensor de humedad capacitivo</a> .....
<a href="#">Sensor de nivel de agua</a> .....
<a href="#">Servomotor SG90</a> .....
<a href="#">Reloj en tiempo real. Módulo RTC</a> .....
<a href="#">Relé 5V</a> .....
<a href="#">Bomba de agua</a> .....
<a href="#">Pantalla OLED 128x64</a> .....
<a href="#">Módulo de carga de batería TP4056</a> .....
<a href="#">Convertidor step-up</a> .....
<a href="#">Diodos LED</a> .....

## Arduino UNO R3

### Features

- **ATMega328P Processor**
  - **Memory**
    - AVR CPU at up to 16 MHz
    - 32KB Flash
    - 2KB SRAM
    - 1KB EEPROM
  - **Security**
    - Power On Reset (POR)
    - Brown Out Detection (BOD)
  - **Peripherals**
    - 2x 8-bit Timer/Counter with a dedicated period register and compare channels
    - 1x 16-bit Timer/Counter with a dedicated period register, input capture and compare channels
    - 1x USART with fractional baud rate generator and start-of-frame detection
    - 1x controller/peripheral Serial Peripheral Interface (SPI)
    - 1x Dual mode controller/peripheral I2C
    - 1x Analog Comparator (AC) with a scalable reference input
    - Watchdog Timer with separate on-chip oscillator
    - Six PWM channels
    - Interrupt and wake-up on pin change
- **ATMega16U2 Processor**
  - 8-bit AVR® RISC-based microcontroller
- **Memory**
  - 16 KB ISP Flash
  - 512B EEPROM
  - 512B SRAM
  - debugWIRE interface for on-chip debugging and programming
- **Power**
  - 2.7-5.5 volts

## 2 Ratings

### 2.1 Recommended Operating Conditions

Symbol	Description	Min	Max
	Conservative thermal limits for the whole board:	-40 °C (-40°F)	85 °C ( 185°F)

**NOTE:** In extreme temperatures, EEPROM, voltage regulator, and the crystal oscillator, might not work as expected due to the extreme temperature conditions

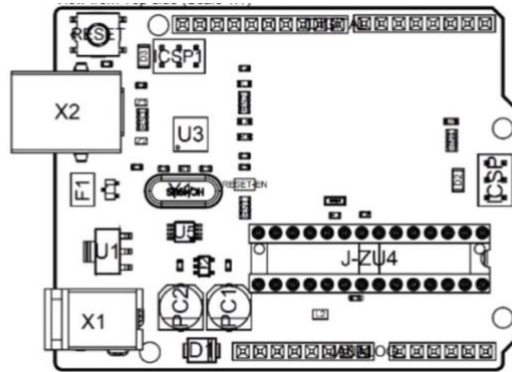
### 2.2 Power Consumption

Symbol	Description	Min	Typ	Max	Unit
VINMax	Maximum input voltage from VIN pad	6	-	20	V
VUSBMax	Maximum input voltage from USB connector		-	5.5	V
PMax	Maximum Power Consumption	-	-	xx	mA

### 3 Functional Overview

#### 3.1 Board Topology

Top view



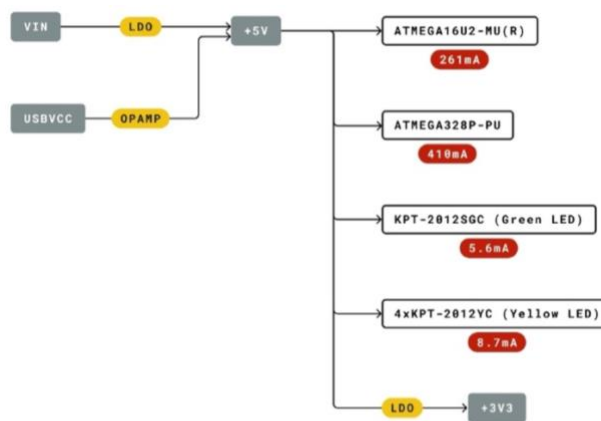
Board topology

Ref.	Description	Ref.	Description
X1	Power jack 2.1x5.5mm	U1	SPX1117M3-L-5 Regulator
X2	USB B Connector	U3	ATMEGA16U2 Module
PC1	EEE-1EA470WP 25V SMD Capacitor	U5	LMV358LIST-A.9 IC
PC2	EEE-1EA470WP 25V SMD Capacitor	F1	Chip Capacitor, High Density
D1	CGRA4007-G Rectifier	ICSP	Pin header connector (through hole 6)
J-ZU4	ATMEGA328P Module	ICSP1	Pin header connector (through hole 6)
Y1	ECS-160-20-4X-DU Oscillator		

#### 3.2 Processor

The Main Processor is ATmega328P running at up to 20 MHz. Most of its pins are connected to the external headers, however some are reserved for internal communication with the USB Bridge coprocessor.

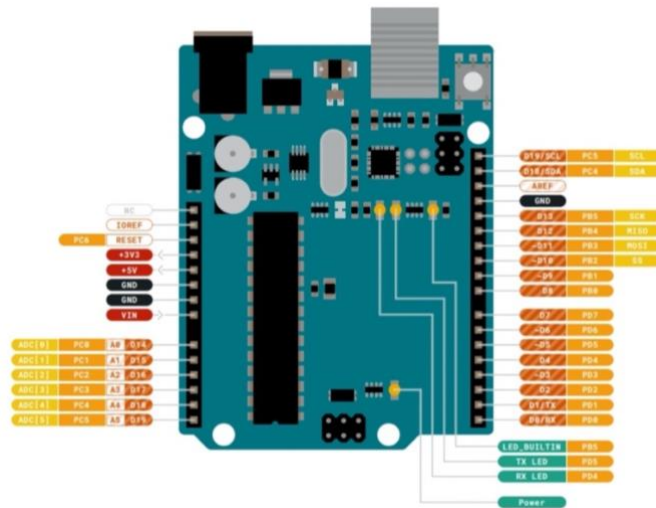
#### 3.3 Power Tree



Legend:  
 □ Component      ● Power I/O      ● Conversion Type  
 ● Max Current      ● Voltage Range

#### 4.6 Board Recovery

All Arduino boards have a built-in bootloader which allows flashing the board via USB. In case a sketch locks up the processor and the board is not reachable anymore via USB it is possible to enter bootloader mode by double-tapping the reset button right after power up.



Pinout

### 5.1 JANALOG

Pin	Function	Type	Description
1	NC	NC	Not connected
2	IOREF	IOREF	Reference for digital logic V - connected to 5V
3	Reset	Reset	Reset
4	+3V3	Power	+3V3 Power Rail
5	+5V	Power	+5V Power Rail
6	GND	Power	Ground
7	GND	Power	Ground
8	VIN	Power	Voltage Input
9	A0	Analog/GPIO	Analog input 0 /GPIO
10	A1	Analog/GPIO	Analog input 1 /GPIO
11	A2	Analog/GPIO	Analog input 2 /GPIO
12	A3	Analog/GPIO	Analog input 3 /GPIO
13	A4/SDA	Analog input/I2C	Analog input 4/I2C Data line
14	A5/SCL	Analog input/I2C	Analog input 5/I2C Clock line

### 5.2 JDIGITAL

Pin	Function	Type	Description
1	D0	Digital/GPIO	Digital pin 0/GPIO
2	D1	Digital/GPIO	Digital pin 1/GPIO
3	D2	Digital/GPIO	Digital pin 2/GPIO
4	D3	Digital/GPIO	Digital pin 3/GPIO
5	D4	Digital/GPIO	Digital pin 4/GPIO
6	D5	Digital/GPIO	Digital pin 5/GPIO
7	D6	Digital/GPIO	Digital pin 6/GPIO
8	D7	Digital/GPIO	Digital pin 7/GPIO
9	D8	Digital/GPIO	Digital pin 8/GPIO
10	D9	Digital/GPIO	Digital pin 9/GPIO
11	SS	Digital	SPI Chip Select
12	MOSI	Digital	SPI1 Main Out Secondary In
13	MISO	Digital	SPI Main In Secondary Out
14	SCK	Digital	SPI serial clock output
15	GND	Power	Ground
16	AREF	Digital	Analog reference voltage
17	A4/SD4	Digital	Analog input 4/I2C Data line (duplicated)
18	A5/SD5	Digital	Analog input 5/I2C Clock line (duplicated)

## Sensor DHT11

DHT11 Temperature & Humidity Sensor features a temperature & humidity sensor complex with a calibrated digital signal output. By using the exclusive digital-signal-acquisition technique and temperature & humidity sensing technology, it ensures high reliability and excellent long-term stability. This sensor includes a resistive-type humidity measurement component and an NTC temperature measurement component, and connects to a high-performance 8-bit microcontroller, offering excellent quality, fast response, anti-interference ability and cost-effectiveness.

Each DHT11 element is strictly calibrated in the laboratory that is extremely accurate on humidity calibration. The calibration coefficients are stored as programmes in the OTP memory, which are used by the sensor's internal signal detecting process. The single-wire serial interface makes system integration quick and easy. Its small size, low power consumption and up-to-20 meter signal transmission making it the best choice for various applications, including those most demanding ones. The component is 4-pin single row pin package. It is convenient to connect and special packages can be provided according to users' request.

## 2. Technical Specifications:

### Overview:

Item	Measurement Range	Humidity Accuracy	Temperature Accuracy	Resolution	Package
DHT11	20-90%RH 0-50 °C	±5%RH	±2°C	1	4 Pin Single Row

### Detailed Specifications:

Parameters	Conditions	Minimum	Typical	Maximum
<b>Humidity</b>				
Resolution		1%RH	1%RH 8 Bit	1%RH
Repeatability			±1%RH	
Accuracy	25°C		±4%RH	
	0-50°C			±5%RH
Interchangeability	Fully Interchangeable			
Measurement Range	0°C	30%RH		90%RH
	25°C	20%RH		90%RH
	50°C	20%RH		80%RH
Response Time (Seconds)	1/e(63%)25°C, 1m/s Air	6 S	10 S	15 S
Hysteresis			±1%RH	
Long-Term Stability	Typical		±1%RH/year	
<b>Temperature</b>				
Resolution		1°C	1°C	1°C
		8 Bit	8 Bit	8 Bit
Repeatability			±1°C	
Accuracy		±1°C		±2°C
Measurement Range		0°C		50°C
Response Time (Seconds)	1/e(63%)	6 S		30 S

### 3. Typical Application (Figure 1)

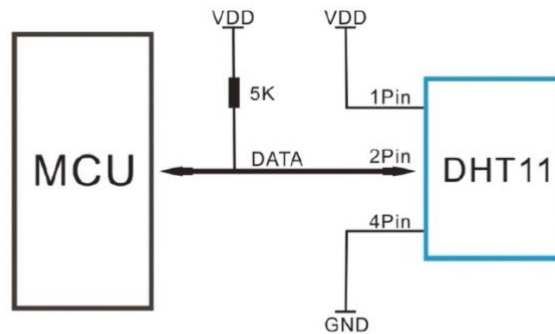


Figure 1 Typical Application

Note: 3Pin – Null; MCU = Micro-computer Unite or single chip Computer

When the connecting cable is shorter than 20 metres, a 5K pull-up resistor is recommended; when the connecting cable is longer than 20 metres, choose a appropriate pull-up resistor as needed.

### 4. Power and Pin

DHT11's power supply is 3-5.5V DC. When power is supplied to the sensor, do not send any instruction to the sensor in within one second in order to pass the unstable status. One capacitor valued 100nF can be added between VDD and GND for power filtering.

### 5. Communication Process: Serial Interface (Single-Wire Two-Way)

Single-bus data format is used for communication and synchronization between MCU and DHT11 sensor. One communication process is about 4ms.

Data consists of decimal and integral parts. A complete data transmission is **40bit**, and the sensor sends **higher data bit** first.

**Data format:** 8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data + 8bit check sum. If the data transmission is right, the check-sum should be the last 8bit of "8bit integral RH data + 8bit decimal RH data + 8bit integral T data + 8bit decimal T data".

#### 5.1 Overall Communication Process (Figure 2, below)

When MCU sends a start signal, DHT11 changes from the low-power-consumption mode to the running-mode, waiting for MCU completing the start signal. Once it is completed, DHT11 sends a response signal of 40-bit data that include the relative humidity and temperature information to MCU. Users can choose to collect (read) some data. Without the start signal from MCU, DHT11 will not give the response signal to MCU. Once data is collected, DHT11 will change to the low-power-consumption mode until it receives a start signal from MCU again.

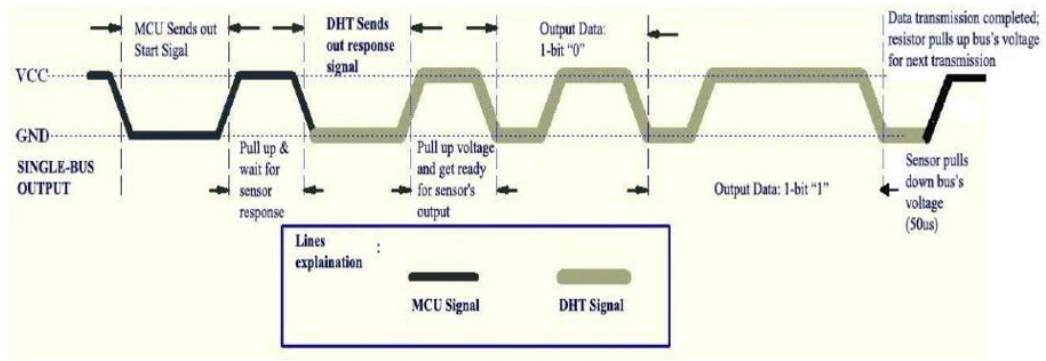


Figure 2 Overall Communication Process

### 5.2 MCU Sends out Start Signal to DHT (Figure 3, below)

Data Single-bus free status is at high voltage level. When the communication between MCU and DHT11 begins, the programme of MCU will set Data Single-bus voltage level from high to low and this process must take at least 18ms to ensure DHT's detection of MCU's signal, then MCU will pull up voltage and wait 20-40us for DHT's response.

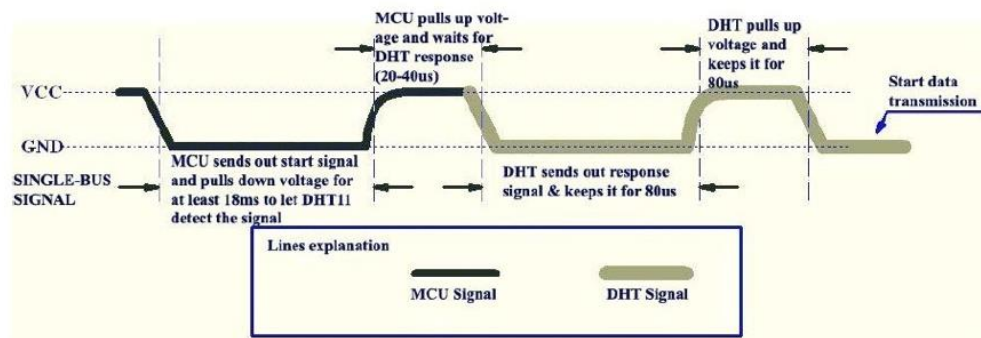


Figure 3 MCU Sends out Start Signal & DHT Responses

### 5.3 DHT Responses to MCU (Figure 3, above)

Once DHT detects the start signal, it will send out a low-voltage-level response signal, which lasts 80us. Then the programme of DHT sets Data Single-bus voltage level from low to high and keeps it for 80us for DHT's preparation for sending data.

When DATA Single-Bus is at the low voltage level, this means that DHT is sending the response signal. Once DHT sent out the response signal, it pulls up voltage and keeps it for 80us and prepares for data transmission.

When DHT is sending data to MCU, every bit of data begins with the 50us low-voltage-level and the length of the following high-voltage-level signal determines whether data bit is "0" or "1" (see Figures 4 and 5 below).



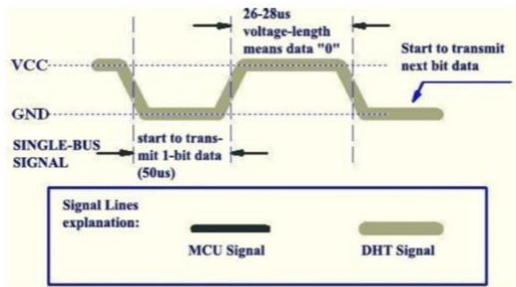


Figure 4 Data "0" Indication

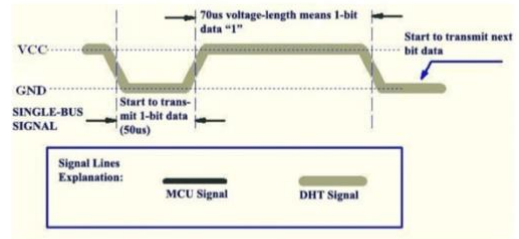


Figure 5 Data "1" Indication

If the response signal from DHT is always at high-voltage-level, it suggests that DHT is not responding properly and please check the connection. When the last bit data is transmitted, DHT11 pulls down the voltage level and keeps it for 50us. Then the Single-Bus voltage will be pulled up by the resistor to set it back to the free status.

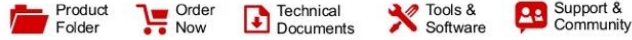
## 6. Electrical Characteristics

VDD=5V, T = 25 °C (unless otherwise stated)

	Conditions	Minimum	Typical	Maximum
Power Supply	DC	3V	5V	5.5V
Current Supply	Measuring	0.5mA		2.5mA
	Average	0.2mA		1mA
	Standby	100uA		150uA
Sampling period	Second	1		

Note: Sampling period at intervals should be no less than 1 second.

## Higrómetro FC-28: Comparador LM393



LM193-N, LM2903-N, LM293-N, LM393-N  
SNOSBJ6G – OCTOBER 1999 – REVISED OCTOBER 2018

### LMx93-N, LM2903-N Low-Power, Low-Offset Voltage, Dual Comparators

#### 1 Features

- Wide Supply
  - Voltage Range: 2.0 V to 36 V
  - Single or Dual Supplies:  $\pm 1.0$  V to  $\pm 18$  V
- Very Low Supply Current Drain (0.4 mA) — Independent of Supply Voltage
- Low Input Biasing Current: 25 nA
- Low Input Offset Current:  $\pm 5$  nA
- Maximum Offset voltage:  $\pm 3$  mV
- Input Common-Mode Voltage Range Includes Ground
- Differential Input Voltage Range Equal to the Power Supply Voltage
- Low Output Saturation Voltage: 250 mV at 4 mA
- Output Voltage Compatible with TTL, DTL, ECL, MOS and CMOS logic systems
- Available in the 8-Bump (12 mil) DSBGA Package
- See AN-1112 (SNVA009) for DSBGA Considerations
- Advantages
  - High Precision Comparators
  - Reduced  $V_{OS}$  Drift Over Temperature
  - Eliminates Need for Dual Supplies
  - Allows Sensing Near Ground
  - Compatible with All Forms of Logic
  - Power Drain Suitable for Battery Operation

#### 2 Applications

- Battery Powered Applications
- Industrial Applications

#### 3 Description

The LM193-N series consists of two independent precision voltage comparators with an offset voltage specification as low as 2.0 mV max for two comparators which were designed specifically to operate from a single power supply over a wide range of voltages. Operation from split power supplies is also possible and the low power supply current drain is independent of the magnitude of the power supply voltage. These comparators also have a unique characteristic in that the input common-mode voltage range includes ground, even though operated from a single power supply voltage.

Application areas include limit comparators, simple analog to digital converters; pulse, squarewave and time delay generators; wide range VCO; MOS clock timers; multivibrators and high voltage digital logic gates. The LM193-N series was designed to directly interface with TTL and CMOS. When operated from both plus and minus power supplies, the LM19-N series will directly interface with MOS logic where their low power drain is a distinct advantage over standard comparators.

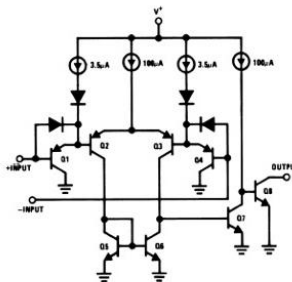
The LM393 and LM2903 parts are available in TI's innovative thin DSBGA package with 8 (12 mil) large bumps.

Device Information<sup>(1)</sup>

PART NUMBER	PACKAGE	BODY SIZE (NOM)
LM193-N	TO-99 (8)	9.08 mm x 9.08 mm
LM293-N	SOIC (8)	4.90 mm x 3.91 mm
LM393-N	DSBGA (8)	1.54 mm x 1.54 mm
	SOIC (8)	4.90 mm x 3.91 mm
LM2903-N	SOIC (8)	4.90 mm x 3.91 mm
	DSBGA (8)	1.54 mm x 1.54 mm

(1) For all available packages, see the orderable addendum at the end of the datasheet.

#### Simplified Schematic



An IMPORTANT NOTICE at the end of this data sheet addresses availability, warranty, changes, use in safety-critical applications, intellectual property matters and other important disclaimers. PRODUCTION DATA.

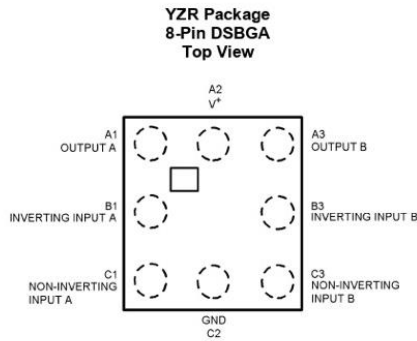
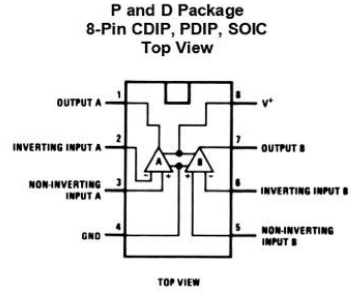
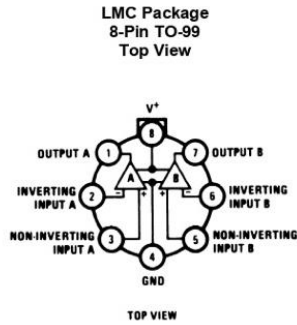


www.ti.com

LM193-N, LM2903-N, LM293-N, LM393-N

SNOSBJ6G—OCTOBER 1999—REVISED OCTOBER 2018

### 5 Pin Configuration and Functions



#### Pin Functions

NAME	PIN NO.		I/O	DESCRIPTION
	PDIP/SOIC/TO-99	DSBGA		
OUTA	1	A1	O	Output, Channel A
-INA	2	B1	I	Inverting Input, Channel A
+INA	3	C1	I	Noninverting Input, Channel A
GND	4	C2	P	Ground
+INB	5	C3	I	Noninverting Input, Channel B
-INB	6	B3	I	Inverting Input, Channel B
OUTB	7	A3	O	Output, Channel B
V+	8	A2	P	Positive power supply



## LM193-N, LM2903-N, LM293-N, LM393-N

SNOSBJ6G –OCTOBER 1999–REVISED OCTOBER 2018

www.ti.com

## 6 Specifications

### 6.1 Absolute Maximum Ratings

over operating free-air temperature range (unless otherwise noted)<sup>(1)(2)(3)</sup>

		MIN	MAX	UNIT
Differential Input Voltage <sup>(4)</sup>			36	V
Input Voltage		-0.3	36	V
Input Current ( $V_{IN} < -0.3$ V) <sup>(5)</sup>			50	mA
Power Dissipation <sup>(6)</sup>	PDIP		780	mW
	TO-99		660	mW
	SOIC		510	mW
	DSBGA		568	mW
Output Short-Circuit to Ground <sup>(7)</sup>			Continu ous	
Lead Temperature (Soldering, 10 seconds)			260	°C
Soldering Information	PDIP Package Soldering (10 seconds)		260	°C
	SOIC Package	Vapor Phase (60 seconds)	215	°C
		Infrared (15 seconds)	220	°C
Storage temperature, $T_{stg}$		-65	150	°C

- (1) *Absolute Maximum Ratings* indicate limits beyond which damage may occur. *Recommended Operating Conditions* indicate conditions for which the device is intended to be functional, but specific performance is not guaranteed. For guaranteed specifications and test conditions, see the Electrical Characteristics.
- (2) Refer to RETS193AX for LM193AH military specifications and to RETS193X for LM193H military specifications.
- (3) If Military/Aerospace specified devices are required, please contact the TI Sales Office/Distributors for availability and specifications.
- (4) Positive excursions of input voltage may exceed the power supply level. As long as the other voltage remains within the common-mode range, the comparator will provide a proper output state. The low input voltage state must not be less than  $-0.3V$  (or  $0.3V$  below the magnitude of the negative power supply, if used).
- (5) This input current will only exist when the voltage at any of the input leads is driven negative. It is due to the collector-base junction of the input PNP transistors becoming forward biased and thereby acting as input diode clamps. In addition to this diode action, there is also lateral NPN parasitic transistor action on the IC chip. This transistor action can cause the output voltages of the comparators to go to the  $V^+$  voltage level (or to ground for a large overdrive) for the time duration that an input is driven negative. This is not destructive and normal output states will re-establish when the input voltage, which was negative, again returns to a value greater than  $-0.3V$ .
- (6) For operating at high temperatures, the LM393 and LM2903 must be derated based on a  $125^{\circ}C$  maximum junction temperature and a thermal resistance of  $170^{\circ}C/W$  which applies for the device soldered in a printed circuit board, operating in a still air ambient. The LM193/LM193A/LM293 must be derated based on a  $150^{\circ}C$  maximum junction temperature. The low bias dissipation and the "ON-OFF" characteristic of the outputs keeps the chip dissipation very small ( $P_D \leq 100$  mW), provided the output transistors are allowed to saturate.
- (7) Short circuits from the output to  $V^+$  can cause excessive heating and eventual destruction. When considering short circuits to ground, the maximum output current is approximately 20 mA independent of the magnitude of  $V^+$ .

### 6.2 ESD Ratings

		VALUE	UNIT
$V_{ESD}$	Electrostatic discharge	Human-body model (HBM), per ANSI/ESDA/JEDEC JS-001 <sup>(1)</sup>	$\pm 1300$ V

- (1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.

### 6.3 Recommended Operating Conditions

over operating free-air temperature range (unless otherwise noted)

	MIN	NOM	MAX	UNIT
Supply Voltage ( $V^+$ ) - Single Supply	2.0		36	V
Supply Voltage ( $V^+$ ) - Dual Supply	$\pm 1.0$		$\pm 18$	V
Operating Input Voltage on ( $V_{IN}$ pin)	0		( $V^+$ ) -1.5V	V
Operating junction temperature, $T_J$ : LM193/LM193A	-55		125	°C
Operating junction temperature, $T_J$ : LM2903	-40		85	°C
Operating junction temperature, $T_J$ : LM293	-25		85	°C
Operating junction temperature, $T_J$ : LM393	0		70	°C



**LM193-N, LM2903-N, LM293-N, LM393-N**

SNOSBJ6G –OCTOBER 1999–REVISED OCTOBER 2018

www.ti.com

**6.7 Electrical Characteristics: LMx93 and LM2903 V<sup>+</sup> = 5 V, T<sub>A</sub> = 25°C**

Unless otherwise stated.

PARAMETER	TEST CONDITIONS	LM193-N			LM293-N, LM393-N			LM2903-N			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
Input Offset Voltage	See <sup>(1)</sup>		1.0	5.0		1.0	5.0		2.0	7.0	mV
Input Bias Current	I <sub>IN(+)</sub> or I <sub>IN(-)</sub> with Output In Linear Range, V <sub>CM</sub> = 0 V <sup>(2)</sup>		25	100		25	250		25	250	nA
Input Offset Current	I <sub>IN(+)</sub> - I <sub>IN(-)</sub> V <sub>CM</sub> = 0 V		3.0	25		5.0	50		5.0	50	nA
Input Common Mode Voltage Range	V <sub>+</sub> = 30 V <sup>(3)</sup>	0		V <sub>+</sub> - 1.5	0		V <sub>+</sub> - 1.5	0		V <sub>+</sub> - 1.5	V
Supply Current	R <sub>L</sub> = ∞		V <sup>+</sup> = 5 V			V <sup>+</sup> = 5 V			V <sup>+</sup> = 5 V		
			0.4	1		0.4	1		0.4	1.0	mA
			V <sup>+</sup> = 36 V			V <sup>+</sup> = 36 V			V <sup>+</sup> = 36 V		
			1	2.5		1	2.5		1	2.5	mA
Voltage Gain	R <sub>L</sub> ≥ 15 kΩ, V <sup>+</sup> = 15 V V <sub>O</sub> = 1 V to 11 V	50	200		50	200		25	100		V/mV
Large Signal Response Time	V <sub>IN</sub> = TTL Logic Swing, V <sub>REF</sub> = 1.4 V V <sub>RL</sub> = 5 V, R <sub>L</sub> = 5.1 kΩ		300			300			300		ns
Response Time	V <sub>RL</sub> = 5 V, R <sub>L</sub> = 5.1 kΩ <sup>(4)</sup>		1.3			1.3			1.5		μs
Output Sink Current	V <sub>IN(-)</sub> = 1 V, V <sub>IN(+)</sub> = 0, V <sub>O</sub> ≤ 1.5 V	6.0	16		6.0	16		6.0	16		mA
Saturation Voltage	V <sub>IN(-)</sub> = 1 V, V <sub>IN(+)</sub> = 0, I <sub>SINK</sub> ≤ 4 mA		250	400		250	400		250	400	mV
Output Leakage Current	V <sub>IN(-)</sub> = 0, V <sub>IN(+)</sub> = 1V, V <sub>O</sub> = 5 V		0.1			0.1			0.1		nA

- (1) At output switch point, V<sub>O</sub> = 1.4V, R<sub>S</sub> = 0 Ω with V<sup>+</sup> from 5V to 30V; and over the full input common-mode range (0V to V<sup>+</sup> - 1.5V), at 25°C.
- (2) The direction of the input current is out of the IC due to the PNP input stage. This current is essentially constant, independent of the state of the output so no loading change exists on the reference or input lines.
- (3) The input common-mode voltage or either input signal voltage should not be allowed to go negative by more than 0.3V. The upper end of the common-mode voltage range is V<sup>+</sup> - 1.5 V at 25°C, but either or both inputs can go to 36 V without damage, independent of the magnitude of V<sup>+</sup>.
- (4) The response time specified is for a 100 mV input step with 5 mV overdrive. For larger overdrive signals 300 ns can be obtained, see *LMx93 and LM193A Typical Characteristics*.



www.ti.com

LM193-N, LM2903-N, LM293-N, LM393-N

SNOSBJ6G—OCTOBER 1999—REVISED OCTOBER 2018

**6.8 Electrical Characteristics: LMx93 and LM2903 (V+ = 5 V)<sup>(1)</sup>**

PARAMETER	TEST CONDITIONS	LM193-N			LM293-N, LM393-N			LM290-N			UNIT
		MIN	TYP	MAX	MIN	TYP	MAX	MIN	TYP	MAX	
Input Offset Voltage	See <sup>(2)</sup>			9			9			15	mV
Input Offset Current	$I_{IN(+)} - I_{IN(-)}$ , $V_{CM} = 0$ V			100			150			50 200	nA
Input Bias Current	$I_{IN(+)}$ or $I_{IN(-)}$ with Output in Linear Range, $V_{CM} = 0$ V <sup>(3)</sup>			300			400			200 500	nA
Input Common Mode Voltage Range	$V^+ = 30$ V <sup>(4)</sup>	0		$V^+ - 2.0$	0		$V^+ - 2.0$	0		$V^+ - 2.0$	V
Saturation Voltage	$V_{IN(-)} = 1$ V, $V_{IN(+)} = 0$ , $I_{SINK} \leq 4$ mA			700			700			400 700	mV
Output Leakage Current	$V_{IN(-)} = 0$ , $V_{IN(+)} = 1$ V, $V_O = 30$ V			1.0			1.0			1.0	$\mu$ A
Differential Input Voltage	Keep All $V_{IN}$ 's $\geq 0$ V (or $V^-$ , if Used), <sup>(5)</sup>			36			36			36	V

- (1) These specifications are limited to  $-55^\circ\text{C} \leq T_A \leq +125^\circ\text{C}$ , for the LM193/LM193A. With the LM293 all temperature specifications are limited to  $-25^\circ\text{C} \leq T_A \leq +85^\circ\text{C}$  and the LM393 temperature specifications are limited to  $0^\circ\text{C} \leq T_A \leq +70^\circ\text{C}$ . The LM2903 is limited to  $-40^\circ\text{C} \leq T_A \leq +85^\circ\text{C}$ .
- (2) At output switch point,  $V_O = 1.4$  V,  $R_S = 0 \Omega$  with  $V^+$  from 5V to 30V; and over the full input common-mode range (0V to  $V^+ - 1.5$  V), at  $25^\circ\text{C}$ .
- (3) The direction of the input current is out of the IC due to the PNP input stage. This current is essentially constant, independent of the state of the output so no loading change exists on the reference or input lines.
- (4) The input common-mode voltage or either input signal voltage should not be allowed to go negative by more than 0.3V. The upper end of the common-mode voltage range is  $V^+ - 1.5$  V at  $25^\circ\text{C}$ , but either or both inputs can go to 36 V without damage, independent of the magnitude of  $V^+$ .
- (5) Positive excursions of input voltage may exceed the power supply level. As long as the other voltage remains within the common-mode range, the comparator will provide a proper output state. The low input voltage state must not be less than  $-0.3$  V (or 0.3V below the magnitude of the negative power supply, if used).

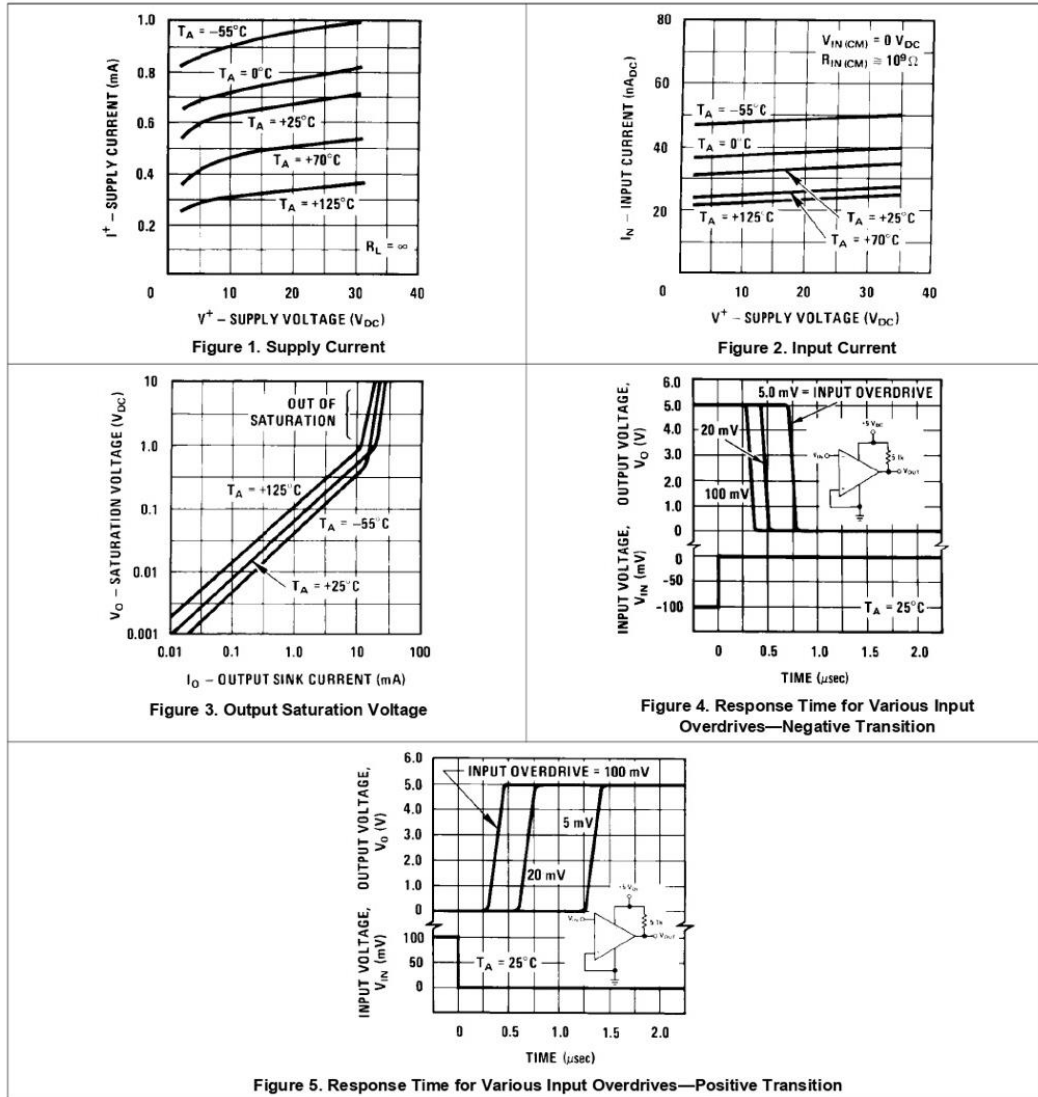


LM193-N, LM2903-N, LM293-N, LM393-N

SNOSBJ6G—OCTOBER 1999—REVISED OCTOBER 2018

www.ti.com

6.9 Typical Characteristics: LMx93 and LM193A



## 7 Detailed Description

### 7.1 Overview

The LM139 provides two independently functioning, high-precision, low  $V_{OS}$  drift, low input bias current comparators in a single package. The low power consumption of 0.4 mA at 5 V and the 2.0 V supply operation makes the LM139 suitable for battery powered applications.

### 7.2 Functional Block Diagram

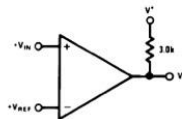


Figure 11. Basic Comparator

### 7.3 Feature Description

The input bias current of 25 nA enables the LM193 to use even very high impedance nodes as inputs. The differential voltage input range equals the supply voltage range.

The LM193 can be operated with a single supply, where  $V+$  can be from 2.0 V to 36 V, or in a dual supply voltage configuration, where GND pin is used as a  $V-$  supply. The supply current draws only 0.4 mA for both comparators.

The output of each comparator in the LM193 is the open collector of a grounded-emitter NPN output transistor which can typically draw up to 16 mA.

### 7.4 Device Functional Modes

A basic comparator circuit is used for converting analog signals to a digital output. The output is HIGH when the voltage on the non-inverting (+IN) input is greater than the inverting (-IN) input. The output is LOW when the voltage on the non-inverting (+IN) input is less than the inverting (-IN) input. The inverting input (-IN) is also commonly referred to as the "reference" or "VREF" input. All pins of any unused comparators should be tied to the negative supply.



## Higrómetro capacitivo



Part Number: SEN0193

Description: Gravity: Analog Capacitive Soil Moisture Sensor- Corrosion Resistant

### **INTRODUCTION**

This product measures soil moisture levels by capacitive sensing, rather than resistive sensing like other types of [moisture](#) sensor. It is made of a corrosion resistant material giving it a long service life. Insert it into soil and impress your friends with the real-time soil moisture data!

The product includes an on-board voltage regulator which gives it an operating voltage range of 3.3 ~ 5.5V. It is compatible with low-voltage MCUs (both 3.3V and 5V logic). To make it compatible with a [Raspberry Pi](#), an [ADC](#) converter is required.

This sensor is compatible with our 3-pin "Gravity" interface, which means it can be directly connected to the [Gravity I/O Expansion Shield](#).

The following image shows an example project - a device that notifies you when a plant needs to be watered.

### **FEATURES**

- Supports Gravity 3-Pin Interface
- Analog Output

### **APPLICATIONS**

- Gardening & Farming
- Moisture Detection
- Intelligent Agriculture

### **SPECIFICATION**

- Operating Voltage: 3.3 ~ 5.5 VDC
- Output Voltage: 1.2 ~ 2.5V
- Interface: PH2.0-3P
- Dimension: 98mm \* 23mm (3.86in x 0.905in)
- Weight: 15g

## Sensor de nivel de agua

### I. Notice

(1) did not carefully read the instructions before you do not give the driver board is powered !  
Avoid faulty wiring caused permanent damage to the drive plate .

(2) Please carefully check pin function , attention condensed identifier , correct wiring ! Do not  
reverse the power cord , resulting in  
Electronic devices burned.

### 2. the product introduction

2013 latest Water Sensor is a Easy to use, compact and lightweight , high cost of water , droplets  
identification and detection sensors. This sensor is working The principle is to measure the size  
of the trace amount of water droplets through the line with a series of parallel wires exposed .  
And domestic and foreign Products compared not only small , powerful, and cleverly designed  
with the following features : First, the amount of water to simulate Conversion ; Second, plasticity  
, based on the sensor output analog values ; Third, low power consumption , high sensitivity ;  
Fourth, can Directly connected to a microprocessor or other logic circuitry , and the controller  
board for a variety of , for example : ArduinoController , STC microcontroller , AVR microcontroller  
and so on.

### 3, the specification parameters

- 1 Product Name: water level sensor
- 2 Item :. K-0135
- 3 Operating voltage :. DC5V
- 4 Working current : less than 20mA
- 5 Sensor Type : Analog
- 6 detection area :. 40mm x16mm
- 7 Production process :. FR4 double-sided HASL
- 8 mounting hole size : 3.0mm
- 9 user-friendly design : half-moon -slip handle depression
- 10 Working temperature :. 10 °C -30 °C
- 11 Operating Humidity : 10% ~ 90 % non -condensing
- 12 Weight :. 3g
- 13 Product Dimensions : 65mm x 20mm x 8mm

### 4., the test Water Sensor Module

We use the Arduino controller to be tested , need to use hardware devices as follows :

- 1, Arduino controller × 1
- 2, Arduino sensor expansion board × 1
- 3, Water Sensor Module × 1
- 4, 3P sensor cable × 2
- 5, IR & LED Modue ( red ) × 1
- 6, USB data communication cable × 1

Water Sensor DuPont line will be connected to the Arduino sensor expansion board interface

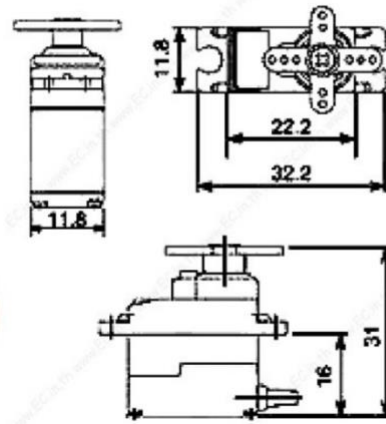
A1. The use of sensors

The red line will be connected to the Arduino piranha light sensor expansion board D8. After  
completing the hardware connection , the code is compiled

After downloading the Arduino inside .

## Servomotor SG90

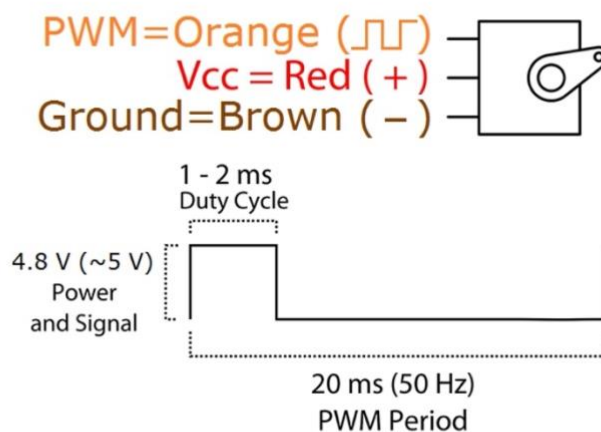
### SG90 9 g Micro Servo



Tiny and lightweight with high output power. Servo can rotate approximately 180 degrees (90 in each direction), and works just like the standard kinds but *smaller*. You can use any servo code, hardware or library to control these servos. Good for beginners who want to make stuff move without building a motor controller with feedback & gear box, especially since it will fit in small places. It comes with a 3 horns (arms) and hardware.

#### Specifications

- Weight: 9 g
- Dimension: 22.2 x 11.8 x 31 mm approx.
- Stall torque: 1.8 kgf·cm
- Operating speed: 0.1 s/60 degree
- Operating voltage: 4.8 V (~5V)
- Dead band width: 10  $\mu$ s
- Temperature range: 0 °C – 55 °C



Position "0" (1.5 ms pulse) is middle, "90" (~2 ms pulse) is all the way to the right, "-90" (~1 ms pulse) is all the way to the left.

## Módulo RTC

### DS3231

### Extremely Accurate I<sup>2</sup>C-Integrated RTC/TCXO/Crystal

#### General Description

The DS3231 is a low-cost, extremely accurate I<sup>2</sup>C real-time clock (RTC) with an integrated temperature-compensated crystal oscillator (TCXO) and crystal. The device incorporates a battery input, and maintains accurate timekeeping when main power to the device is interrupted. The integration of the crystal resonator enhances the long-term accuracy of the device as well as reduces the piece-part count in a manufacturing line. The DS3231 is available in commercial and industrial temperature ranges, and is offered in a 16-pin, 300-mil SO package.

The RTC maintains seconds, minutes, hours, day, date, month, and year information. The date at the end of the month is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with an AM/PM indicator. Two programmable time-of-day alarms and a programmable square-wave output are provided. Address and data are transferred serially through an I<sup>2</sup>C bidirectional bus.

A precision temperature-compensated voltage reference and comparator circuit monitors the status of V<sub>CC</sub> to detect power failures, to provide a reset output, and to automatically switch to the backup supply when necessary. Additionally, the RST pin is monitored as a pushbutton input for generating a  $\mu$ P reset.

#### Benefits and Features

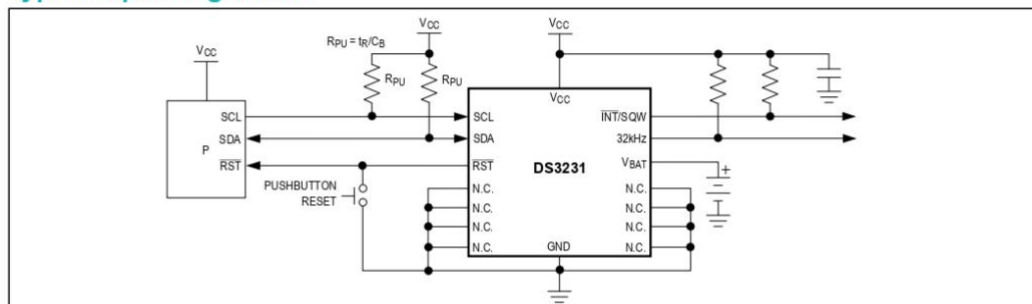
- Highly Accurate RTC Completely Manages All Timekeeping Functions
  - Real-Time Clock Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the Week, and Year, with Leap-Year Compensation Valid Up to 2100
  - Accuracy  $\pm 2$ ppm from 0°C to +40°C
  - Accuracy  $\pm 3.5$ ppm from -40°C to +85°C
  - Digital Temp Sensor Output:  $\pm 3$ °C Accuracy
  - Register for Aging Trim
  - RST Output/Pushbutton Reset Debounce Input
  - Two Time-of-Day Alarms
  - Programmable Square-Wave Output Signal
- Simple Serial Interface Connects to Most Microcontrollers
  - Fast (400kHz) I<sup>2</sup>C Interface
- Battery-Backup Input for Continuous Timekeeping
  - Low Power Operation Extends Battery-Backup Run Time
  - 3.3V Operation
- Operating Temperature Ranges: Commercial (0°C to +70°C) and Industrial (-40°C to +85°C)
- Underwriters Laboratories® (UL) Recognized

#### Applications

- Servers
- Telematics
- Utility Power Meters
- GPS

Ordering Information and Pin Configuration appear at end of data sheet.

#### Typical Operating Circuit



Underwriters Laboratories is a registered certification mark of Underwriters Laboratories Inc.

DS3231

Extremely Accurate I<sup>2</sup>C-Integrated  
RTC/TCXO/Crystal

### Absolute Maximum Ratings

Voltage Range on Any Pin Relative to Ground .....-0.3V to +6.0V  
 Junction-to-Ambient Thermal Resistance ( $\theta_{JA}$ ) (Note 1) 73°C/W  
 Junction-to-Case Thermal Resistance ( $\theta_{JC}$ ) (Note 1) .....23°C/W  
 Operating Temperature Range  
 DS3231S .....0°C to +70°C  
 DS3231SN .....-40°C to +85°C

Junction Temperature .....+125°C  
 Storage Temperature Range .....-40°C to +85°C  
 Lead Temperature (soldering, 10s) .....+260°C  
 Soldering Temperature (reflow, 2 times max) .....+260°C  
 (see the *Handling, PCB Layout, and Assembly* section)

**Note 1:** Package thermal resistances were obtained using the method described in JEDEC specification JESD51-7, using a four-layer board. For detailed information on package thermal considerations, refer to [www.maximintegrated.com/thermal-tutorial](http://www.maximintegrated.com/thermal-tutorial).

*Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

### Recommended Operating Conditions

( $T_A = T_{MIN}$  to  $T_{MAX}$ , unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Supply Voltage	V <sub>CC</sub>		2.3	3.3	5.5	V
	V <sub>BAT</sub>		2.3	3.0	5.5	V
Logic 1 Input SDA, SCL	V <sub>IH</sub>		0.7 x V <sub>CC</sub>		V <sub>CC</sub> + 0.3	V
Logic 0 Input SDA, SCL	V <sub>IL</sub>		-0.3		0.3 x V <sub>CC</sub>	V

### Electrical Characteristics

(V<sub>CC</sub> = 2.3V to 5.5V, V<sub>CC</sub> = Active Supply (see Table 1), T<sub>A</sub> = T<sub>MIN</sub> to T<sub>MAX</sub>, unless otherwise noted.) (Typical values are at V<sub>CC</sub> = 3.3V, V<sub>BAT</sub> = 3.0V, and T<sub>A</sub> = +25°C, unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Active Supply Current	I <sub>CCA</sub>	(Notes 4, 5)	V <sub>CC</sub> = 3.63V		200	μA
			V <sub>CC</sub> = 5.5V		300	
Standby Supply Current	I <sub>CCS</sub>	I <sup>2</sup> C bus inactive, 32kHz output on, SQW output off (Note 5)	V <sub>CC</sub> = 3.63V		110	μA
			V <sub>CC</sub> = 5.5V		170	
Temperature Conversion Current	I <sub>CCSCONV</sub>	I <sup>2</sup> C bus inactive, 32kHz output on, SQW output off	V <sub>CC</sub> = 3.63V		575	μA
			V <sub>CC</sub> = 5.5V		650	
Power-Fail Voltage	V <sub>PF</sub>		2.45	2.575	2.70	V
Logic 0 Output, 32kHz, INT/SQW, SDA	V <sub>OL</sub>	I <sub>OL</sub> = 3mA			0.4	V
Logic 0 Output, RST	V <sub>OL</sub>	I <sub>OL</sub> = 1mA			0.4	V
Output Leakage Current 32kHz, INT/SQW, SDA	I <sub>LO</sub>	Output high impedance	-1	0	+1	μA
Input Leakage SCL	I <sub>LI</sub>		-1		+1	μA
RST Pin I/O Leakage	I <sub>OL</sub>	RST high impedance (Note 6)	-200		+10	μA
V <sub>BAT</sub> Leakage Current (V <sub>CC</sub> Active)	I <sub>BATLKG</sub>			25	100	nA

DS3231

Extremely Accurate I<sup>2</sup>C-Integrated  
RTC/TCXO/Crystal

**Electrical Characteristics (continued)**

(V<sub>CC</sub> = 2.3V to 5.5V, V<sub>CC</sub> = Active Supply (see Table 1), T<sub>A</sub> = T<sub>MIN</sub> to T<sub>MAX</sub>, unless otherwise noted.) (Typical values are at V<sub>CC</sub> = 3.3V, V<sub>BAT</sub> = 3.0V, and T<sub>A</sub> = +25°C, unless otherwise noted.) (Notes 2, 3)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Output Frequency	f <sub>OUT</sub>	V <sub>CC</sub> = 3.3V or V <sub>BAT</sub> = 3.3V		32.768		kHz
Frequency Stability vs. Temperature (Commercial)	Δf/f <sub>OUT</sub>	V <sub>CC</sub> = 3.3V or V <sub>BAT</sub> = 3.3V, aging offset = 00h	0°C to +40°C		±2	ppm
			>40°C to +70°C		±3.5	
Frequency Stability vs. Temperature (Industrial)	Δf/f <sub>OUT</sub>	V <sub>CC</sub> = 3.3V or V <sub>BAT</sub> = 3.3V, aging offset = 00h	-40°C to <0°C		±3.5	ppm
			0°C to +40°C		±2	
Frequency Stability vs. Voltage	Δf/V			1		ppm/V
Trim Register Frequency Sensitivity per LSB	Δf/LSB	Specified at:	-40°C	0.7		ppm
			+25°C	0.1		
			+70°C	0.4		
			+85°C	0.8		
Temperature Accuracy	Temp	V <sub>CC</sub> = 3.3V or V <sub>BAT</sub> = 3.3V	-3		+3	°C
Crystal Aging	Δf/f <sub>O</sub>	After reflow, not production tested	First year	±1.0		ppm
			0–10 years	±5.0		

**Electrical Characteristics**

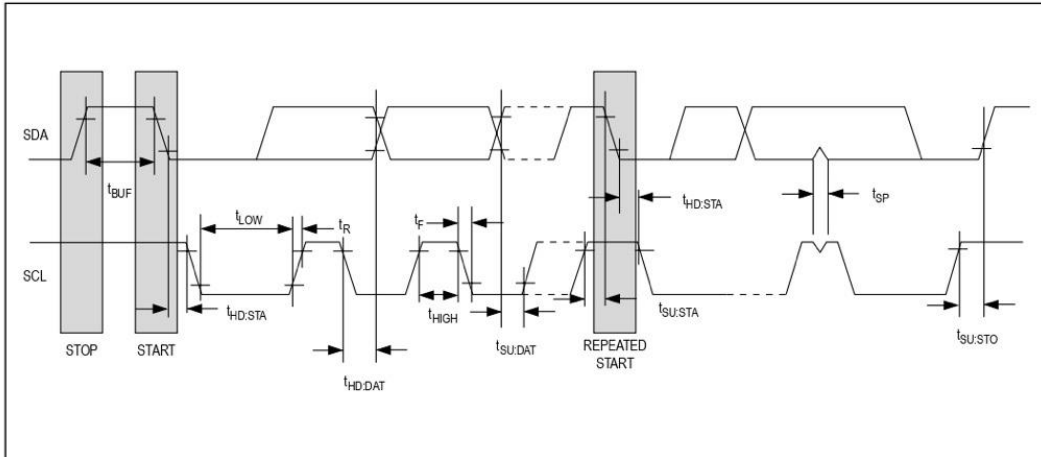
(V<sub>CC</sub> = 0V, V<sub>BAT</sub> = 2.3V to 5.5V, T<sub>A</sub> = T<sub>MIN</sub> to T<sub>MAX</sub>, unless otherwise noted.) (Note 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Active Battery Current	I <sub>BATA</sub>	E <sub>OSC</sub> = 0, BBSQW = 0, SCL = 400kHz (Note 5)	V <sub>BAT</sub> = 3.63V		70	μA
			V <sub>BAT</sub> = 5.5V		150	
Timekeeping Battery Current	I <sub>BATT</sub>	E <sub>OSC</sub> = 0, BBSQW = 0, EN32kHz = 1, SCL = SDA = 0V or SCL = SDA = V <sub>BAT</sub> (Note 5)	V <sub>BAT</sub> = 3.63V	0.84	3.0	μA
			V <sub>BAT</sub> = 5.5V	1.0	3.5	
Temperature Conversion Current	I <sub>BATTC</sub>	E <sub>OSC</sub> = 0, BBSQW = 0, SCL = SDA = 0V or SCL = SDA = V <sub>BAT</sub>	V <sub>BAT</sub> = 3.63V		575	μA
			V <sub>BAT</sub> = 5.5V		650	
Data-Retention Current	I <sub>BATTD</sub>	E <sub>OSC</sub> = 1, SCL = SDA = 0V, +25°C			100	nA

DS3231

Extremely Accurate I<sup>2</sup>C-Integrated  
RTC/TCXO/Crystal

Data Transfer on I<sup>2</sup>C Serial Bus



**WARNING:** Negative undershoots below -0.3V while the part is in battery-backed mode may cause loss of data.

**Note 2:** Limits at -40°C are guaranteed by design and not production tested.

**Note 3:** All voltages are referenced to ground.

**Note 4:** I<sub>CCA</sub>—SCL clocking at max frequency = 400kHz.

**Note 5:** Current is the averaged input current, which includes the temperature conversion current.

**Note 6:** The RST pin has an internal 50kΩ (nominal) pullup resistor to V<sub>CC</sub>.

**Note 7:** After this period, the first clock pulse is generated.

**Note 8:** A device must internally provide a hold time of at least 300ns for the SDA signal (referred to the V<sub>IH(MIN)</sub> of the SCL signal) to bridge the undefined region of the falling edge of SCL.

**Note 9:** The maximum t<sub>HD:DAT</sub> needs only to be met if the device does not stretch the low period (t<sub>LOW</sub>) of the SCL signal.

**Note 10:** A fast-mode device can be used in a standard-mode system, but the requirement t<sub>SU:DAT</sub> ≥ 250ns must then be met. This is automatically the case if the device does not stretch the low period of the SCL signal. If such a device does stretch the low period of the SCL signal, it must output the next data bit to the SDA line t<sub>R(MAX)</sub> + t<sub>SU:DAT</sub> = 1000 + 250 = 1250ns before the SCL line is released.

**Note 11:** C<sub>B</sub>—total capacitance of one bus line in pF.

**Note 12:** The parameter t<sub>OSF</sub> is the period of time the oscillator must be stopped for the OSF flag to be set over the voltage range of 0.0V ≤ V<sub>CC</sub> ≤ V<sub>CC(MAX)</sub> and 2.3V ≤ V<sub>BAT</sub> ≤ 3.4V.

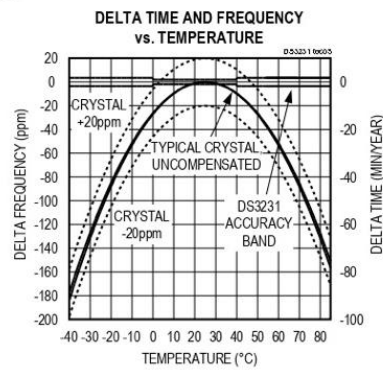
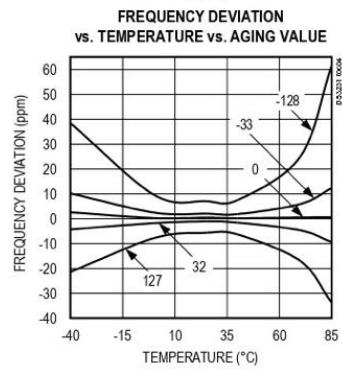
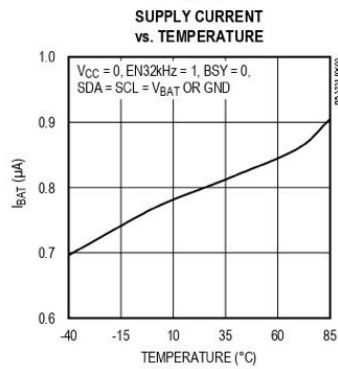
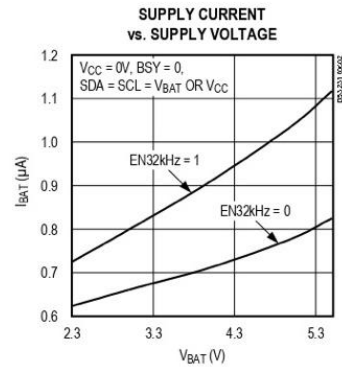
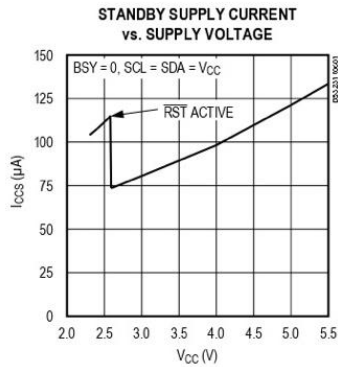
**Note 13:** This delay applies only if the oscillator is enabled and running. If the  $\overline{EOSC}$  bit is a 1, t<sub>REC</sub> is bypassed and  $\overline{RST}$  immediately goes high. The state of RST does not affect the I<sup>2</sup>C interface, RTC, or TCXO.

DS3231

Extremely Accurate I<sup>2</sup>C-Integrated RTC/TCXO/Crystal

**Typical Operating Characteristics**

(V<sub>CC</sub> = +3.3V, T<sub>A</sub> = +25°C, unless otherwise noted.)

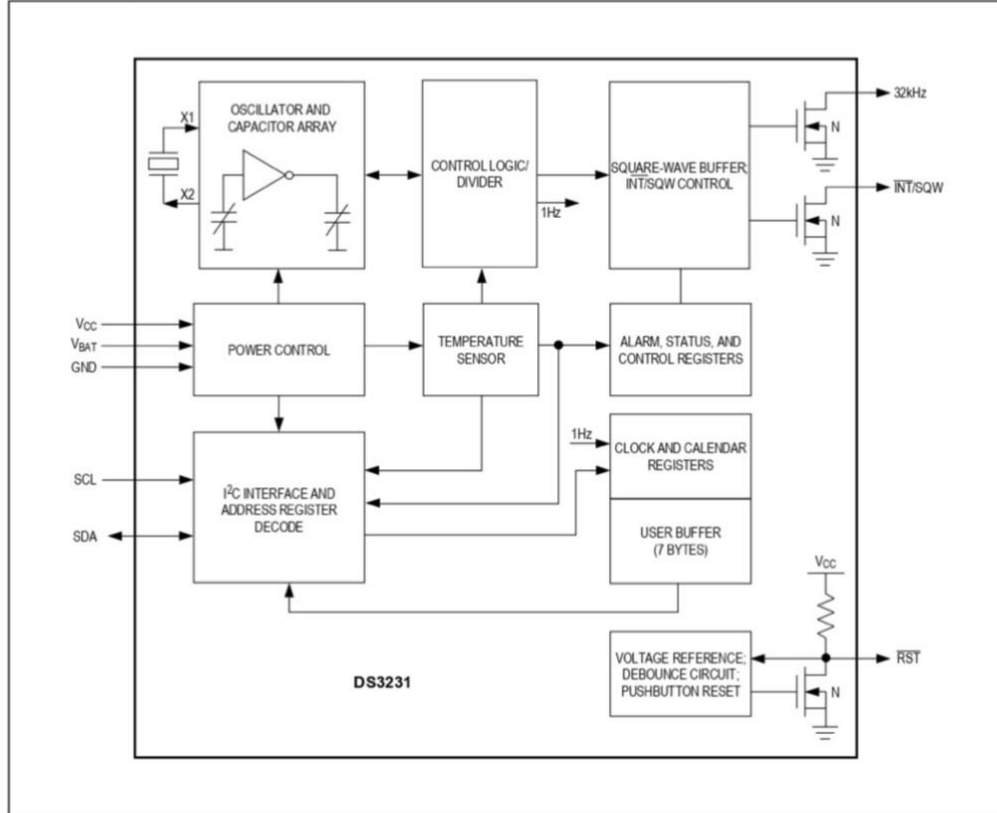




DS3231

Extremely Accurate I<sup>2</sup>C-Integrated  
RTC/TCXO/Crystal

Block Diagram



Pin Description

PIN	NAME	FUNCTION
1	32kHz	32kHz Output. This open-drain pin requires an external pullup resistor. When enabled, the output operates on either power supply. It may be left open if not used.
2	V <sub>CC</sub>	DC Power Pin for Primary Power Supply. This pin should be decoupled using a 0.1µF to 1.0µF capacitor. If not used, connect to ground.
3	INT/SQW	Active-Low Interrupt or Square-Wave Output. This open-drain pin requires an external pullup resistor connected to a supply at 5.5V or less. This multifunction pin is determined by the state of the INTCN bit in the Control Register (0Eh). When INTCN is set to logic 0, this pin outputs a square wave and its frequency is determined by RS2 and RS1 bits. When INTCN is set to logic 1, then a match between the timekeeping registers and either of the alarm registers activates the INT/SQW pin (if the alarm is enabled). Because the INTCN bit is set to logic 1 when power is first applied, the pin defaults to an interrupt output with alarms disabled. The pullup voltage can be up to 5.5V, regardless of the voltage on V <sub>CC</sub> . If not used, this pin can be left unconnected.
4	RST	Active-Low Reset. This pin is an open-drain input/output. It indicates the status of V <sub>CC</sub> relative to the V <sub>PF</sub> specification. As V <sub>CC</sub> falls below V <sub>PF</sub> , the RST pin is driven low. When V <sub>CC</sub> exceeds V <sub>PF</sub> , for t <sub>RST</sub> , the RST pin is pulled high by the internal pullup resistor. The active-low, open-drain output is combined with a debounced pushbutton input function. This pin can be activated by a pushbutton reset request. It has an internal 50kΩ nominal value pullup resistor to V <sub>CC</sub> . No external pullup resistors should be connected. If the oscillator is disabled, t <sub>REC</sub> is bypassed and RST immediately goes high.
5-12	N.C.	No Connection. Must be connected to ground.
13	GND	Ground
14	V <sub>BAT</sub>	Backup Power-Supply Input. When using the device with the V <sub>BAT</sub> input as the primary power source, this pin should be decoupled using a 0.1µF to 1.0µF low-leakage capacitor. When using the device with the V <sub>BAT</sub> input as the backup power source, the capacitor is not required. If V <sub>BAT</sub> is not used, connect to ground. The device is UL recognized to ensure against reverse charging when used with a primary lithium battery. Go to <a href="http://www.maximintegrated.com/qa/info/ul">www.maximintegrated.com/qa/info/ul</a> .
15	SDA	Serial Data Input/Output. This pin is the data input/output for the I <sup>2</sup> C serial interface. This open-drain pin requires an external pullup resistor. The pullup voltage can be up to 5.5V, regardless of the voltage on V <sub>CC</sub> .
16	SCL	Serial Clock Input. This pin is the clock input for the I <sup>2</sup> C serial interface and is used to synchronize data movement on the serial interface. Up to 5.5V can be used for this pin, regardless of the voltage on V <sub>CC</sub> .

Relé 5V



# Handson Technology

User Guide

## 1 Channel 5V Optical Isolated Relay Module

This is a LOW Level 5V 1-channel relay interface board, needs a 15-20mA driver current. It can be used to control various appliances and equipment with large current. It is equipped with high-current relays that work under AC250V 10A or DC30V 10A. It has a standard interface that can be controlled directly by microcontroller. This module is optically isolated from high voltage side for safety requirement and also prevent ground loop when interface to microcontroller.

**Brief Data:**

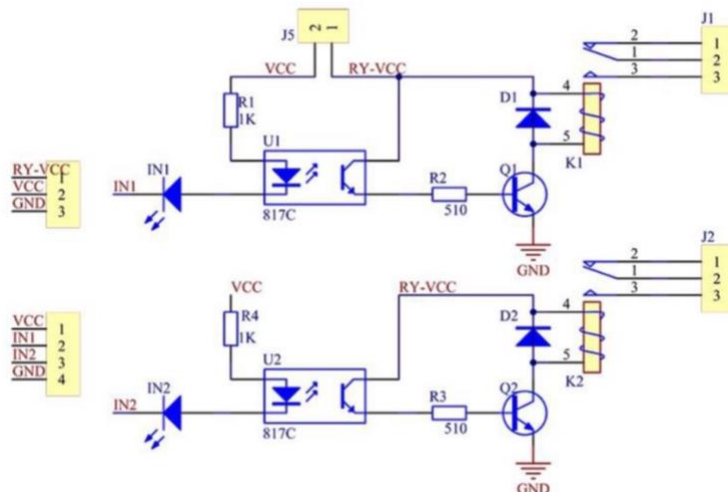
- Operating Voltage: 5Vdc.
- Relay Maximum output: DC 30V/10A, AC 250V/10A.
- 1 Channel Relay Module with Opto-coupler. LOW Level Trigger expansion board, which is compatible with Arduino control board.
- Standard interface that can be controlled directly by microcontroller ( 8051, AVR, \*PIC, DSP, ARM, ARM, MSP430, TTL logic).
- Relay of high quality low noise relays SPDT. A common terminal, a normally open, one normally closed terminal.
- Opto-Coupler isolation, for high voltage safety and prevent ground loop with microcontroller.

**Schematic:**

VCC and RY-VCC are also the power supply of the relay module. When you need to drive a large power load, you can take the jumper cap off and connect an extra power to RY-VCC to supply the relay; connect VCC to 5V of the MCU board to supply input signals.

NOTES: If you want complete optical isolation, connect "Vcc" to Arduino +5 volts but do NOT connect Arduino Ground. Remove the Vcc to JD-Vcc jumper. Connect a separate +5 supply to "JD-Vcc" and board Gnd. This will supply power to the transistor drivers and relay coils.

If relay isolation is enough for your application, connect Arduino +5 and Gnd, and leave Vcc to JD-Vcc jumper in place.

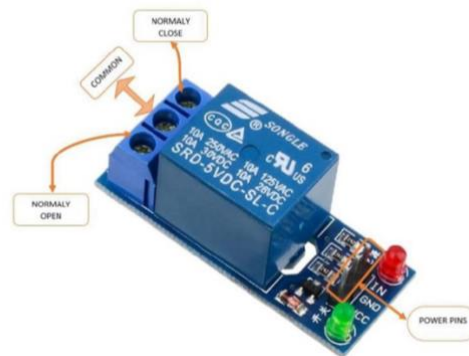


It is sometimes possible to use this relay boards with 3.3V signals, if the JD-VCC (Relay Power) is provided from a +5V supply and the VCC to JD-VCC jumper is removed. That 5V relay supply could be totally isolated from the 3.3V device, or have a common ground if opto-isolation is not needed. If used with isolated 3.3V signals, VCC (To the input of the opto-isolator, next to the IN pins) should be connected to the 3.3V device's +3.3V supply.

NOTE: Some Raspberry-Pi users have found that some relays are reliable and others do not actuate sometimes. It may be necessary to change the value of R1 from 1000 ohms to something like 220 ohms, or supply +5V to the VCC connection.

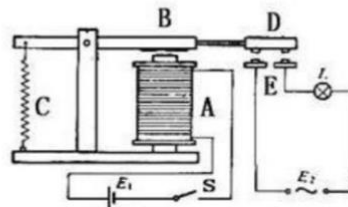
NOTE: The digital inputs from Arduino are Active LOW: The relay actuates and LED lights when the input pin is LOW, and turns off on HIGH.

### **Module Layout:**



### **Operating Principle:**

See the picture below: A is an electromagnet, B armature, C spring, D moving contact, and E fixed contacts. There are two fixed contacts, a normally closed one and a normally open one. When the coil is not energized, the normally open contact is the one that is off, while the normally closed one is the other that is on.



Supply voltage to the coil and some currents will pass through the coil thus generating the electromagnetic effect. So the armature overcomes the tension of the spring and is attracted to the core, thus closing the moving contact of the armature and the normally open (NO) contact or you may say releasing the former and the normally closed (NC) contact. After the coil is de-energized, the electromagnetic force disappears and the armature moves back to the original position, releasing the moving contact and normally closed contact. The closing and releasing of the contacts results in power on and off of the circuit.

### **Input:**

VCC : Connected to positive supply voltage (supply power according to relay voltage)

GND : Connected to supply ground.

IN1: Signal triggering terminal 1 of relay module

### **Output:**

Each module of the relay has one NC (normally close), one NO (normally open) and one COM (Common) terminal. So there are 2 NC, 2 NO and 2 COM of the channel relay in total. NC stands for the normal close port contact and the state without power. NO stands for the normal open port contact and the state with power. COM means the common port. You can choose NC port or NO port according to whether power or not.

## Bomba de agua



### Descripción

Esta mini bomba es totalmente sumergible y muy funcional para aplicaciones donde requieras extraer agua. Puede funcionar con un rango de tensión de 2.5 a 6 VCD. Permite tener un flujo de hasta de dos litros de agua por minuto (80-120 l/h).

Internamente está conformado por un motor DC de 800mA, además con su hélice te permitirá impulsar el agua que recibe en la entrada, obteniendo a la salida una presión aún mayor.

El motor eléctrico viene incluido en sus terminales unos capacitores cerámicos 104 que los permitirán mejorar la estabilidad de voltaje en el circuito. Pues previene picos y el ruido eléctrico que la fuente de tensión puede generar.

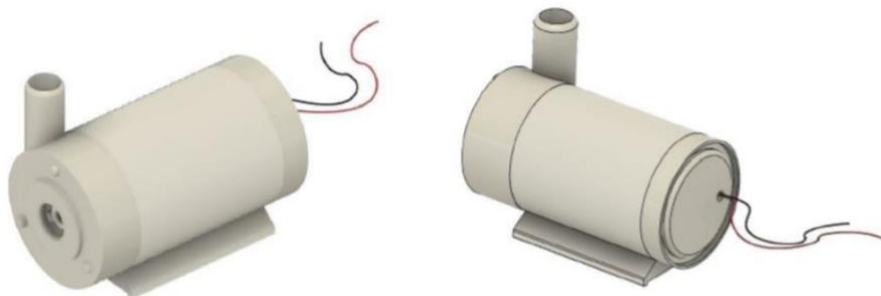


Imagen-.1

### Especificaciones técnicas

- Voltaje de trabajo: 2.5 – 6 VCD
- Altura Bombeo máximo: 0.4 – 1.1 metros
- Caudal Bombeo máximo: 80-120 l/H
- Corriente: 800mA
- Potencia: 0.4 -1.5 W
- Horas de trabajo continuo: 500 horas
- Longitud cable: 20 cm
- No cuenta con manguera
- SKU: AR0366
- Color: Amarillento
- Dimensiones: 45x34x23 aproximadamente

### Aplicaciones

- Filtros o flujo de acuarios o peceras pequeñas (máx 100 litros)
- Cascadas o fuentes
- Regaderas de pasto

### Elementos de la bomba de agua sumergible



Imagen-. 4

Pantalla OLED

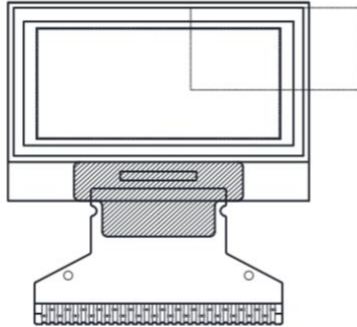


www.vishay.com

OLED-128O064D-WPP3N00000

Vishay

128 x 64 Graphic OLED



FEATURES

- Type: graphic
- Display format: 128 x 64 dots
- Built-in controller: SSD1306BZ
- Duty cycle: 1/64
- +3 V power supply
- Interface: 6800, 8080, serial, and I<sup>2</sup>C
- Material categorization: for definitions of compliance please see [www.vishay.com/doc?99912](http://www.vishay.com/doc?99912)



MECHANICAL DATA		
ITEM	STANDARD VALUE	UNIT
Module dimension	26.7 x 19.26 x 1.65	mm
Viewing area	23.938 x 12.058	
Active area	21.738 x 10.858	
Dot size	0.148 x 0.148	
Dot pitch	0.17 x 0.17	
Mounting hole	n/a	

ABSOLUTE MAXIMUM RATINGS				
ITEM	SYMBOL	STANDARD VALUE		UNIT
		MIN.	MAX.	
Supply voltage for logic <sup>(1)(2)</sup>	V <sub>DD</sub>	0	4	V
Supply voltage for display <sup>(1)(2)</sup>	V <sub>CC</sub>	0	15	
Operating temperature	T <sub>OP</sub>	-40	+80	°C
Storage temperature	T <sub>STG</sub>	-40	+80	

Notes

- (1) All the above voltages are on the basis of "V<sub>SS</sub> = 0 V".
- (2) When this module is used beyond the above absolute maximum ratings, permanent breakage of the module may occur. Also, for normal operations, it is desirable to use this module under the conditions according to section 6 "Electrical Characteristics". If this module is used beyond these conditions, malfunctioning of the module can occur and the reliability of the module may deteriorate.

ELECTRICAL CHARACTERISTICS						
ITEM	SYMBOL	CONDITION	STANDARD VALUE			UNIT
			MIN.	TYP.	MAX.	
Supply voltage for logic	V <sub>DD</sub>	-	2.8	3.0	3.3	V
Supply voltage for display	V <sub>CC</sub>	-	10	12	15	
Input high voltage	V <sub>IH</sub>	-	0.8 V <sub>DD</sub>	-	V <sub>DDIO</sub>	
Input low voltage	V <sub>IL</sub>	-	0	-	0.2 V <sub>DD</sub>	
Output high voltage	V <sub>OH</sub>	-	0.9 V <sub>DD</sub>	-	V <sub>DDIO</sub>	
Output low voltage	V <sub>OL</sub>	-	0	-	0.1 V <sub>DD</sub>	
50 % check board operating current	I <sub>CC</sub>	V <sub>CC</sub> = 12 V	9	10	12	mA
CIE <sub>x</sub> (white)		(CIE1931)	0.26	0.28	0.30	
CIE <sub>y</sub> (white)		(CIE1931)	0.30	0.32	0.34	

OPTIONS				
EMITTING COLOR				
YELLOW	GREEN	RED	BLUE	WHITE
-	-	-	-	Yes

Revision: 14-Dec-16

1

Document Number: 37905

For technical questions, contact: [displays@vishay.com](mailto:displays@vishay.com)

THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE. THE PRODUCTS DESCRIBED HEREIN AND THIS DOCUMENT ARE SUBJECT TO SPECIFIC DISCLAIMERS, SET FORTH AT [www.vishay.com/doc?91000](http://www.vishay.com/doc?91000)



www.vishay.com

OLED-128O064D-WPP3N00000

Vishay

INTERFACE PIN FUNCTION																														
PIN NO.	SYMBOL	FUNCTION																												
1	NC (GND)	Reserved pin (supporting pin) The supporting pins can reduce the influences from stresses on the function pins. These pins must be connected to external ground.																												
2	C2N	Positive terminal of the flying inverting capacitor negative terminal of the flying boost capacitor The charge-pump capacitors are required between the terminals. They must be floated when the converter is not used.																												
3	C2P																													
4	C1P																													
5	C1N																													
6	V <sub>BAT</sub>	Power supply for DC/DC converter circuit This is the power supply pin for the internal buffer of the DC/DC voltage converter. It must be connected to external source when the converter is used. It should be connected to V <sub>DD</sub> when the converter is not used.																												
7	NC	NC																												
8	V <sub>SS</sub>	Ground of logic circuit This is a ground pin. It also acts as a reference for the logic pins. It must be connected to external ground.																												
9	V <sub>DD</sub>	Power supply for logic This is a voltage supply pin. It must be connected to external source.																												
10	BS0	Communicating protocol select These pins are MCU interface selection input. See the following table:																												
11	BS1	<table border="1"> <thead> <tr> <th></th> <th>I<sup>2</sup>C</th> <th>3-wire SPI</th> <th>4-wire SPI</th> <th>8-bit 68XX parallel</th> <th>8-bit 80XX parallel</th> </tr> </thead> <tbody> <tr> <td>BS0</td> <td>0</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>BS1</td> <td>1</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> </tr> <tr> <td>BS2</td> <td>0</td> <td>0</td> <td>0</td> <td>1</td> <td>1</td> </tr> </tbody> </table>						I <sup>2</sup> C	3-wire SPI	4-wire SPI	8-bit 68XX parallel	8-bit 80XX parallel	BS0	0	1	0	0	0	BS1	1	0	0	0	1	BS2	0	0	0	1	1
	I <sup>2</sup> C	3-wire SPI	4-wire SPI	8-bit 68XX parallel	8-bit 80XX parallel																									
BS0	0	1	0	0	0																									
BS1	1	0	0	0	1																									
BS2	0	0	0	1	1																									
12	BS2																													
13	CS#	Chip select This pin is the chip select input. The chip is enabled for MCU communication only when CS# is pulled low.																												
14	RES#	Power reset for controller and driver This pin is reset signal input. When the pin is low, initialization of the chip is executed.																												
15	D / C#	Data / command control This pin is data / command control pin. When the pin is pulled high, the input at D7 to D0 is treated as display data. When the pin is pulled low, the input at D7 to D0 will be transferred to the command register. For detail relationship to MCU interface signals, please refer to the timing characteristics diagrams. When the pin is pulled high and serial interface mode is selected, the data at SDIN is treated as data. When it is pulled low, the data at SDIN will be transferred to the command register. In I <sup>2</sup> C mode, this pin acts as SA0 for slave address selection.																												
16	R / W#	Read / write select or write This pin is MCU interface input. When interfacing to a 68XX-series microprocessor, this pin will be used as read / write (R/W#) selection input. Pull this pin to "high" for read mode and pull it to "low" for write mode. When 80XX interface mode is selected, this pin will be the write (WR#) input. Data write operation is initiated when this pin is pulled low and the CS# is pulled low.																												
17	E / RD#	Read / write enable or read This pin is MCU interface input. When interfacing to a 68XX-series microprocessor, this pin will be used as the enable (E) signal. Read / write operation is initiated when this pin is pulled high and the CS# is pulled low. When connecting to an 80XX-microprocessor, this pin receives the read (RD#) signal. Data read operation is initiated when this pin is pulled low and CS# is pulled low.																												
18 to 25	D0 to D7	Host data input / output bus These pins are 8-bit bi-directional data bus to be connected to the microprocessor's data bus. When serial mode is selected, D1 will be the serial data input SDIN and D0 will be the serial clock input SCLK. When I <sup>2</sup> C mode is selected, D2 and D1 should be tied together and serve as SDA <sub>out</sub> and SDA <sub>in</sub> in application and D0 is the serial clock input SCL.																												
26	I <sub>REF</sub>	Current reference for brightness adjustment This pin is segment current reference pin. A resistor should be connected between this pin and V <sub>SS</sub> . Set the current lower than 12.5 μA.																												
27	V <sub>COMH</sub>	Voltage output high level for COM signal This pin is the input pin for the voltage output high level for COM signals. A capacitor should be connected between this pin and V <sub>SS</sub> .																												
28	V <sub>CC</sub>	Power supply for OEL panel This is the most positive voltage supply pin of the chip. A stabilization capacitor should be connected between this pin and V <sub>SS</sub> when the converter is used. It must be connected to external source when the converter is not used.																												
29	V <sub>LSS</sub>	Ground of analog circuit This is an analog ground pin. It should be connected to V <sub>SS</sub> externally.																												
30	NC (GND)	Reserved pin (supporting pin) The supporting pins can reduce the influences from stresses on the function pins. These pins must be connected to external ground.																												

Revision: 14-Dec-16

2

Document Number: 37905

For technical questions, contact: [displays@vishay.com](mailto:displays@vishay.com)

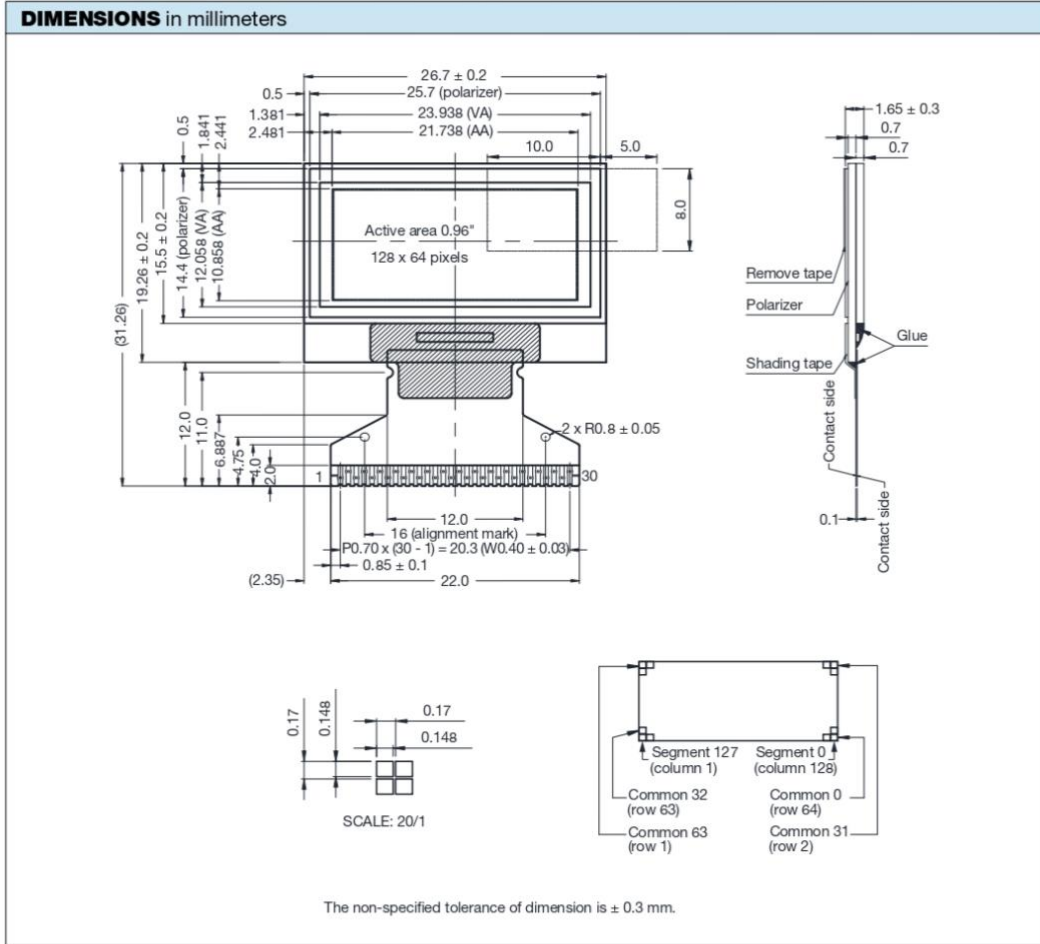
THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE. THE PRODUCTS DESCRIBED HEREIN AND THIS DOCUMENT ARE SUBJECT TO SPECIFIC DISCLAIMERS, SET FORTH AT [www.vishay.com/doc?91000](http://www.vishay.com/doc?91000)



www.vishay.com

**OLED-128O064D-WPP3N00000**

Vishay





## Módulo TP4056



南京拓微集成电路有限公司  
NanJing Top Power ASIC Corp.

### TP4056 1A Standalone Linear Li-Ion Battery Charger with Thermal Regulation in SOP-8

#### DESCRIPTION

The TP4056 is a complete constant-current/constant-voltage linear charger for single cell lithium-ion batteries. Its SOP package and low external component count make the TP4056 ideally suited for portable applications. Furthermore, the TP4056 can work within USB and wall adapter.

No blocking diode is required due to the internal PMOSFET architecture and have prevent to negative Charge Current Circuit. Thermal feedback regulates the charge current to limit the die temperature during high power operation or high ambient temperature. The charge voltage is fixed at 4.2V, and the charge current can be programmed externally with a single resistor. The TP4056 automatically terminates the charge cycle when the charge current drops to 1/10th the programmed value after the final float voltage is reached.

TP4056 Other features include current monitor, under voltage lockout, automatic recharge and two status pin to indicate charge termination and the presence of an input voltage.

#### FEATURES

- Programmable Charge Current Up to 1000mA
- No MOSFET, Sense Resistor or Blocking Diode Required
- Complete Linear Charger in SOP-8 Package for Single Cell Lithium-Ion Batteries
- Constant-Current/Constant-Voltage
- Charges Single Cell Li-Ion Batteries Directly from USB Port
- Preset 4.2V Charge Voltage with 1.5% Accuracy
- Automatic Recharge
- two Charge **Status** Output Pins
- C/10 Charge Termination
- 2.9V Trickle Charge Threshold (TP4056)
- Soft-Start Limits Inrush Current
- Available Radiator in 8-Lead SOP Package, the Radiator need connect GND or impending

#### ABSOLUTE MAXIMUM RATINGS

- Input Supply Voltage( $V_{CC}$ ):  $-0.3V \sim 8V$
- TEMP:  $-0.3V \sim 10V$
- CE:  $-0.3V \sim 10V$
- BAT Short-Circuit Duration: Continuous
- BAT Pin Current: 1200mA
- PROG Pin Current: 1200uA
- Maximum Junction Temperature:  $145^{\circ}C$
- Operating Ambient Temperature Range:  $-40^{\circ}C \sim 85^{\circ}C$
- Lead Temp.(Soldering, 10sec):  $260^{\circ}C$

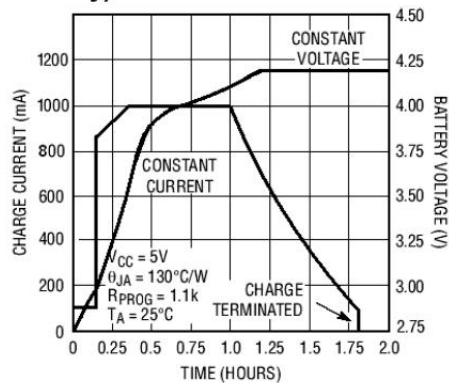
#### APPLICATIONS

- Cellular Telephones, PDAs, GPS
- Charging Docks and Cradles
- Digital Still Cameras, Portable Devices
- USB Bus-Powered Chargers,Chargers

#### PACKAGE/ORDER INFORMATION

<p>SOP-8</p>	
photo	<b>ORDER PART NUMBER</b> TP4056-42-SOP8-PP
	<b>PART MARKING</b> TP4056

#### Complete Charge Cycle (1000mAh Battery)





南京拓微集成电路有限公司  
NanJing Top Power ASIC Corp.

**TEMP(Pin 1) :Temperature Sense Input** Connecting TEMP pin to NTC thermistor's output in Lithium ion battery pack. If TEMP pin's voltage is below 45% or above 80% of supply voltage  $V_{IN}$  for more than 0.15S, this means that battery's temperature is too high or too low, charging is suspended. The temperature sense function can be disabled by grounding the TEMP pin.

**PROG(Pin 2): Constant Charge Current Setting and Charge Current Monitor Pin** charge current is set by connecting a resistor  $R_{ISET}$  from this pin to GND. When in precharge mode, the ISET pin's voltage is regulated to 0.2V. When in constant charge current mode, the ISET pin's voltage is regulated to 2V. In all modes during charging, the voltage on ISET pin can be used to measure the charge current as follows:

$$I_{BAT} = \frac{V_{PROG}}{R_{PROG}} \times 1200 \quad (V_{PROG}=1V)$$

**GND(Pin3): Ground Terminal**

**Vcc(Pin 4): Positive Input Supply Voltage**  $V_{IN}$  is the power supply to the internal circuit. When  $V_{IN}$  drops to within 30mv of the BAT pin voltage, TP4056 enters low power sleep mode, dropping BAT pin's current to less than 2uA.

**BAT(Pin5): Battery Connection Pin.** Connect the positive terminal of the battery to BAT pin. BAT pin draws less than 2uA current in chip disable mode or in sleep mode. BAT pin provides charge current to the battery and provides regulation voltage of 4.2V.

**STDBY(Pin6): Open Drain Charge Status Output** When the battery Charge Termination, the  $\overline{STDBY}$  pin is pulled low by an internal switch, otherwise  $\overline{STDBY}$  pin is in high impedance state.

**CHRG (Pin7): Open Drain Charge Status Output** When the battery is being charged, the  $\overline{CHRG}$  pin is pulled low by an internal switch, otherwise  $\overline{CHRG}$  pin is in high impedance state.

**CE(Pin8): Chip Enable Input.** A high input will put the device in the normal operating mode.

Pulling the CE pin to low level will put the YP4056 into disable mode. The CE pin can be driven by TTL or CMOS logic level.

### ELECTRICAL CHARACTERISTICS

The ● denotes specifications which apply over the full operating temperature range, otherwise specifications are at  $T_A=25^\circ\text{C}$ ,  $V_{CC}=5V$ , unless otherwise noted.

SYMBOL	PARAMETER	CONDITIONS	MIN	TYP	MAX	UNITS
$V_{CC}$	Input Supply Voltage		● 4.0	5	8.0	V
$I_{CC}$	Input Supply Current	Charge Mode, $R_{PROG} = 1.2k$	●	150	500	$\mu\text{A}$
		StandbyMode(Charge Terminated)	●	55	100	$\mu\text{A}$
		Shutdown Mode ( $R_{PROG}$ Not Connected, $V_{CC} < V_{BAT}$ , or $V_{CC} < V_{UV}$ )	●	55	100	$\mu\text{A}$
$V_{FLOAL}$	Regulated Output (Float) Voltage	$0^\circ\text{C} \leq T_A \leq 85^\circ\text{C}$ , $I_{BAT}=40\text{mA}$	4.137	4.2	4.263	V
$I_{BAT}$	BAT Pin Current Text condition: $V_{BAT}=4.0V$	$R_{PROG} = 2.4k$ , Current Mode	● 450	500	550	mA
		$R_{PROG} = 1.2k$ , Current Mode	● 950	1000	1050	mA
		Standby Mode, $V_{BAT} = 4.2V$	● 0	-2.5	-6	$\mu\text{A}$
$I_{TRIKL}$	Trickle Charge Current	$V_{BAT} < V_{TRIKL}$ , $R_{PROG}=1.2K$	● 120	130	140	mA
$V_{TRIKL}$	Trickle Charge Threshold Voltage	$R_{PROG}=1.2K$ , $V_{BAT}$ Rising	2.8	2.9	3.0	V
$V_{TRHYS}$	Trickle Charge Hysteresis Voltage	$R_{PROG}=1.2K$	60	80	100	mV
$T_{LIM}$	Junction Temperature in Constant Temperature Mode			145		$^\circ\text{C}$



南京拓微集成电路有限公司  
NanJing Top Power ASIC Corp.

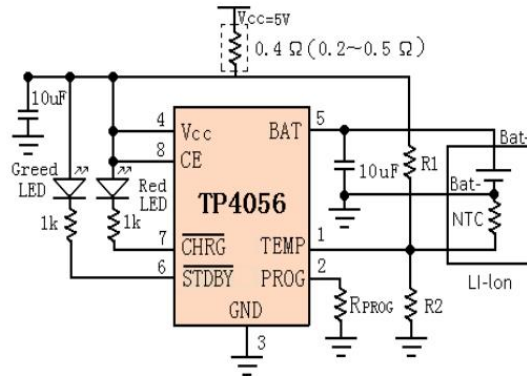
**indicator light state**

Charge state	Red LED $\overline{\text{CHRG}}$	Green LED $\overline{\text{STDBY}}$
charging	bright	extinguish
Charge Termination	extinguish	bright
Vin too low; Temperature of battery too low or too high; no battery	extinguish	extinguish
BAT PIN Connect 10u Capacitance; No battery	Green LED bright, Red LED Coruscate T=1-4 S	

**Rprog Current Setting**

R <sub>PROG</sub> (k)	I <sub>BAT</sub> (mA)
10	130
5	250
4	300
3	400
2	580
1.66	690
1.5	780
1.33	900
1.2	1000

**TYPICAL APPLICATIONS**



Convertidor step-up



**AEROSEMI**

**MT3608**

**High Efficiency 1.2MHz  
2A Step Up Converter**

**FEATURES**

- Integrated 80mΩ Power MOSFET
- 2V to 24V Input Voltage
- 1.2MHz Fixed Switching Frequency
- Internal 4A Switch Current Limit
- Adjustable Output Voltage
- Internal Compensation
- Up to 28V Output Voltage
- Automatic Pulse Frequency Modulation Mode at Light Loads
- up to 97% Efficiency
- Available in a 6-Pin SOT23-6 Package

**APPLICATIONS**

- Battery-Powered Equipment
- Set-Top Boxed
- LCD Bais Supply
- DSL and Cable Modems and Routers
- Networking cards powered from PCI or PCI express slots

**GENERAL DESCRIPTION**

The MT3608 is a constant frequency, 6-pin SOT23 current mode step-up converter intended for small, low power applications. The MT3608 switches at 1.2MHz and allows the use of tiny, low cost capacitors and inductors 2mm or less in height. Internal soft-start results in small inrush current and extends battery life.

The MT3608 features automatic shifting to pulse frequency modulation mode at light loads. The MT3608 includes under-voltage lockout, current limiting, and thermal overload protection to prevent damage in the event of an output overload. The MT3608 is available in a small 6-pin SOT-23 package.

**TYPICAL APPLICATION**

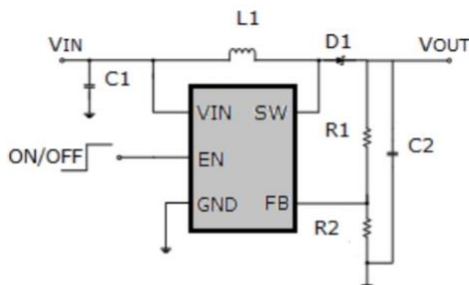


Figure 1. Basic Application Circuit

**Efficiency**

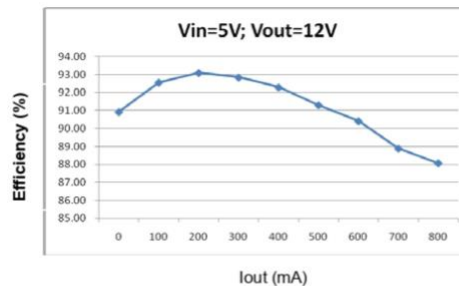


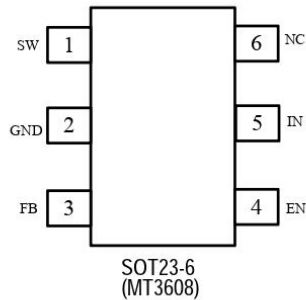
Figure 2. Efficiency Curve

## MT3608

### ABSOLUTE MAXIMUM RATINGS

IN, EN voltages .....	-0.3V to 26V	SW Voltage .....	-0.3V to 30V
Operating Temperature.....	-40°C to +85°C	Storage Temperature Range	-65°C to 150°C
FB Voltages .....	-0.3V to 6V	Peak SW Sink and Source Current .....	4A
Junction Temperature .....	160°C	Lead Temperature (Soldering, 10s) ...	+300°C

### PACKAGE/ORDER INFORMATION



### PIN DESCRIPTION

PIN	NAME	FUNCTION
1	SW	Power Switch Output. SW is the drain of the internal MOSFET switch. Connect the power inductor and output rectifier to SW. SW can swing between GND and 28V.
2	GND	Ground Pin
3	FB	Feedback Input. The FB voltage is 0.6V. Connect a resistor divider to FB.
4	EN	Regulator On/Off Control Input. A high input at EN turns on the converter, and a low input turns it off. When not used, connect EN to the input supply for automatic startup.
5	IN	Input Supply Pin. Must be locally bypassed.
6	NC	NC

## MT3608

**ELECTRICAL CHARACTERISTICS**(V<sub>IN</sub>=V<sub>EN</sub>=5V, T<sub>A</sub> = 25°C, unless otherwise noted.)

Parameter	Conditions	MIN	TYP	MAX	unit
Operating Input Voltage		2		24	V
Under Voltage Lockout				1.98	V
Under Voltage Lockout Hysteresis			100		mV
Current (Shutdown)	V <sub>EN</sub> = 0V		0.1	1	μA
Quiescent Current (PFM)	V <sub>FB</sub> =0.7V, No switch		100	200	μA
Quiescent Current (PWM)	V <sub>FB</sub> =0.5V, switch		1.6	2.2	mA
Switching Frequency			1.2		MHz
Maximum Duty Cycle	V <sub>FB</sub> = 0V	90			%
EN Input High Voltage		1.5			V
EN Input Low Voltage				0.4	V
FB Voltage		0.588	0.6	0.612	V
FB Input Bias Current	V <sub>FB</sub> = 0.6V	-50	-10		nA
SW On Resistance (1)			80	150	mΩ
SW Current Limit (1)	V <sub>IN</sub> = 5V, Duty cycle=50%		4		A
SW Leakage	V <sub>SW</sub> = 20V			1	μA
Thermal Shutdown			155		°C

**Note:**

1) Guaranteed by design, not tested.

## MT3608

### OPERATION

The MT3608 uses a fixed frequency, peak current mode boost regulator architecture to regulate voltage at the feedback pin. The operation of the MT3608 can be understood by referring to the block diagram of Figure 3. At the start of each oscillator cycle the MOSFET is turned on through the control circuitry. To prevent sub-harmonic oscillations at duty cycles greater than 50 percent, a stabilizing ramp is added to the output of the current sense amplifier and the result is fed into the negative input of the PWM comparator. When this voltage equals

The output voltage of the error amplifier the power MOSFET is turned off. The voltage at the output of the error amplifier is an amplified version of the difference between the 0.6V bandgap reference voltage and the feedback voltage. In this way the peak current level keeps the output in regulation. If the feedback voltage starts to drop, the output of the error amplifier increases. These results in more current to flow through the power MOSFET, thus increasing the power delivered to the output. The MT3608 has internal soft start to limit the amount of input current at startup and to also limit the amount of overshoot on the output.

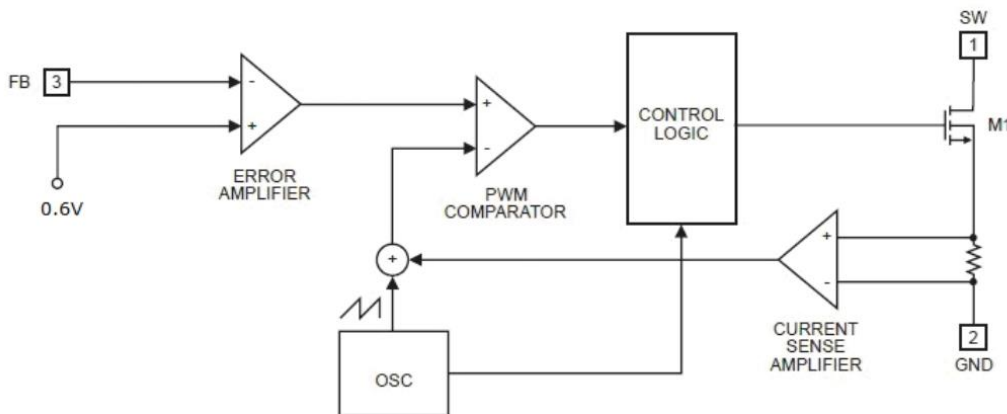
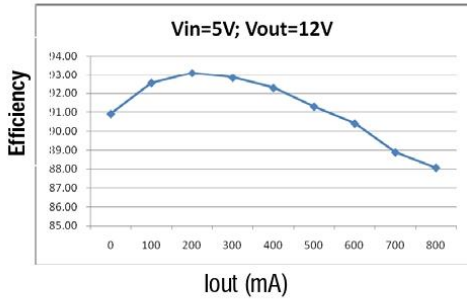


Figure 3. Functional Block Diagram

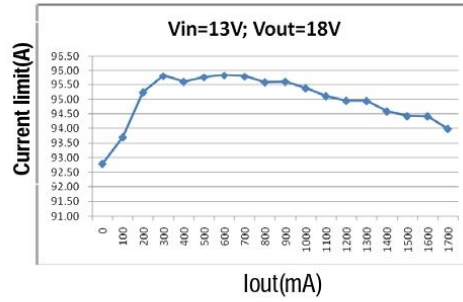
MT3608

**TYPICAL OPERATING CHARACTERISTICS**

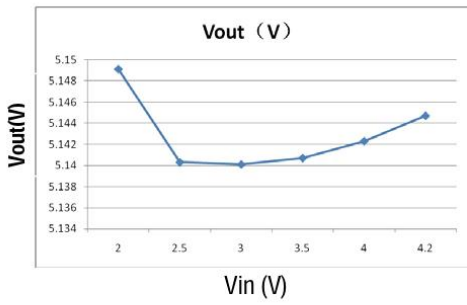
**Efficiency Curve**



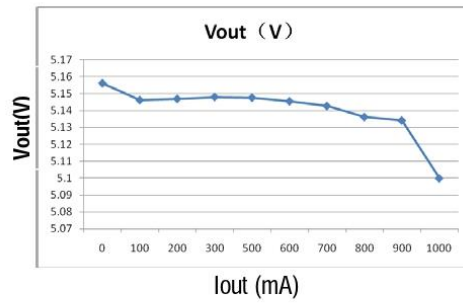
**Efficiency Curve**



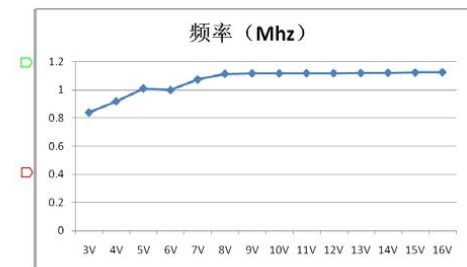
**line Regulation**



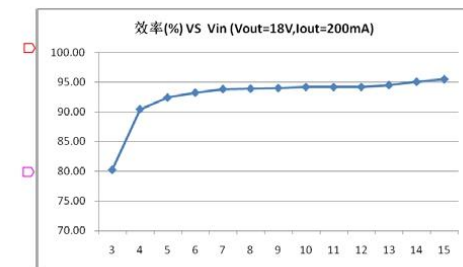
**Load regulation**



**Freq VS Vin**



**Efficiency VS Vin**





## MT3608

### APPLICATION INFORMATION

#### Setting the Output Voltage

The internal reference  $V_{REF}$  is 0.6V (Typical). The output voltage is divided by a resistor divider,  $R_1$  and  $R_2$  to the FB pin. The output voltage is given by

$$V_{OUT} = V_{REF} \times \left(1 + \frac{R_1}{R_2}\right)$$

#### Inductor Selection

The recommended values of inductor are 4.7 to 22 $\mu$ H. Small size and better efficiency are the major concerns for portable device, such as MT3608 used for mobile phone. The inductor should have low core loss at 1.2MHz and low DCR for better efficiency. To avoid inductor saturation current rating should be considered.

#### Capacitor Selection

Input and output ceramic capacitors of 22 $\mu$ F are recommended for MT3608 applications. For better voltage filtering, ceramic capacitors with low ESR are recommended. X5R and X7R types are suitable because of their wider voltage and temperature ranges.

#### Diode Selection

Schottky diode is a good choice for MT3608 because of its low forward voltage drop and fast reverse recovery. Using Schottky diode can get better efficiency. The high speed rectification is also a good characteristic of Schottky diode for high switching frequency. Current rating of the diode must meet the root mean square of the peak current and output average current multiplication as following :

$$I_D(RMS) \approx \sqrt{I_{OUT} \times I_{PEAK}}$$

The diode's reverse breakdown voltage should be larger than the output voltage.

#### Layout Consideration

For best performance of the MT3608, the following guidelines must be strictly followed.

- Input and Output capacitors should be placed close to the IC and connected to ground plane to reduce noise coupling.
- The GND should be connected to a strong ground plane for heat sinking and noise protection.
- Keep the main current traces as possible as short and wide.
- SW node of DC-DC converter is with high frequency voltage swing. It should be kept at a small area.
- Place the feedback components as close as possible to the IC and keep away from the noisy devices.

## Diodos LED



### Diodos leds estándar

En nuestro catálogo también disponemos de una completa selección de **leds de inserción y smd**, ya sea para aplicaciones de señalización, pilotaje de una función o las propias donde se requieren leds de muy alto rendimiento.

En esta sección encontrará diodos leds de tipo "piraña", de 3 mm, 5 mm y 10 mm en distintos todos de color y con varias luminosidades, así como versiones para montaje superficial SMD de tipo 0603, 0805 y 1206.

Así mismo, disponemos de leds de 5 mm de diámetro con cinco chips internos que constituyen los modelos que más luminosidad ofrecen. También están disponibles leds de emisión ultravioleta e infrarrojos.

### Standard leds

Our standard leds selection is one of the most complete range available in market. **insertion and smd models** are suitable for simple applications or for highest lighting level designs where powerful leds are necessary.

Our list of models comprises insertion super-flux models, 3 mm, 5 mm and 10 mm leds, all of them available in different lighting rates. Also SMT models in sizes as 0603, 0805 and 1206 are available in our product line. Besides, multichip 5mm leds are best option for projects where highest lighting level is necessary. Ultraviolet and infrared diodes are available too.

#### Led de señal estándar

#### Standard signal leds



Cápsula tintada difusa.  
Diffused tinted lens.

REFERENCIA Y COLOR	CÁPSULA (DIÁMETRO)	LUMINOSIDAD	LONGITUD DE ONDA	ÁNGULO	CORRIENTE DE ALIMENTACIÓN	TENSIÓN DE ALIMENTACIÓN
PART NUMBER AND COLOR	PACKAGE	LUMINOUS INTENSITY	WAVELENGTH	VIEWING ANGLE	FORWARD CURRENT	FORWARD VOLTAGE
BL-B5141	3 mm	10 mcd	700 nm	40°	20 mA	1,9-2,2 VDC
BL-B2141	3 mm	40 mcd	568 nm	35°	20 mA	1,9-2,2 VDC
BL-B3141	3 mm	30 mcd	585 nm	35°	20 mA	1,9-2,2 VDC
BL-B5134	5 mm	12 mcd	700 nm	35°	20 mA	2,2-2,6 VDC
BL-B2134	5 mm	80 mcd	568 nm	35°	20 mA	2,2-2,6 VDC
BL-B3134	5 mm	70 mcd	585 nm	35°	20 mA	2,2-2,6 VDC

#### Led de señal especiales

#### Special signal leds



Cápsula tintada difusa.  
Diffused tinted lens.

REFERENCIA Y COLOR	CÁPSULA (DIÁMETRO)	LUMINOSIDAD	Tº COLOR / LONGITUD DE ONDA	ÁNGULO	CORRIENTE DE ALIMENTACIÓN	TENSIÓN DE ALIMENTACIÓN
PART NUMBER AND COLOR	PACKAGE	LUMINOUS INTENSITY	CCT / WAVELENGTH	VIEWING ANGLE	FORWARD CURRENT	FORWARD VOLTAGE
RF-WW03A3SWI2-B3	3 mm	1300 mcd	5500k	45°	20 mA	3,2-4,0 VDC
RF-WW03A3SRA1-B3	3 mm	120 mcd	630 nm	45°	20 mA	2,0-2,4 VDC
RF-WW03A3SBB1-B3	3 mm	270 mcd	475 nm	45°	20 mA	3,2-4,0 VDC
RF-WW03A3SGI1-B3	3 mm	1000 mcd	525 nm	45°	20 mA	3,2-4,0 VDC
RF-WW03A3AYA1-B3	3 mm	120 mcd	595 nm	45°	20 mA	2,0-2,4 VDC
RF-WW05A3SWC2-B3	5 mm	1000 mcd	5500k	45°	20 mA	3,2-4,0 VDC
RF-WW05A3SRA1-B3	5 mm	120 mcd	630 nm	45°	20 mA	2,0-2,4 VDC
RF-WW05A3SBB1-B3	5 mm	270 mcd	475 nm	45°	20 mA	3,2-4,0 VDC
RF-WW05A3SGI1-B3	5 mm	780 mcd	525 nm	45°	20 mA	3,2-4,0 VDC
RF-WW05A3AYA1-B3	5 mm	120 mcd	595 nm	45°	20 mA	2,0-2,4 VDC

#### Leds de alta luminosidad

#### Ultra bright leds



Cápsula transparente.  
Water clear lens.

REFERENCIA Y COLOR	CÁPSULA (DIÁMETRO)	LUMINOSIDAD	Tº COLOR / LONGITUD DE ONDA	ÁNGULO	CORRIENTE DE ALIMENTACIÓN	TENSIÓN DE ALIMENTACIÓN
PART NUMBER AND COLOR	PACKAGE	LUMINOUS INTENSITY	CCT / WAVELENGTH	VIEWING ANGLE	FORWARD CURRENT	FORWARD VOLTAGE
RF-WW03A3SWQ4-N2	3 mm	6300 mcd	5500k	25°	20 mA	3,2-4,0 VDC
RF-WW03A3SRN4-B2	3 mm	3000 mcd	630 nm	40°	20 mA	2,0-2,4 VDC
RF-WW03A3SRQ4-N2	3 mm	3700 mcd	630 nm	25°	20 mA	2,0-2,4 VDC
RF-WW03A3SBI4-N2	3 mm	1700 mcd	475 nm	25°	20 mA	3,2-4,0 VDC
RF-WW03A3SBQ4-N2	3 mm	2200 mcd	475 nm	25°	20 mA	3,2-4,0 VDC
RF-WW03A3SGG4-N2	3 mm	4900 mcd	525 nm	25°	20 mA	3,2-4,0 VDC
RF-WW03A3AYQ4-N2	3 mm	4900 mcd	595 nm	15°	20 mA	2,0-2,4 VDC

Nota: Contamos con una gama de diodos leds más extensa con modelos de distintas características y niveles de luminosidad. Consulte con nuestro departamento comercial para ampliar información.

Note: We have a wider led range with models of different specifications and lighting rating. Check with our sales department for more information.

○ Blanco frío / Natural white   ● Rojo / Red   ● Azul / Blue   ● Verde / Green   ● Ambar / Amber