



**Escuela de Doctorado
y Estudios de Posgrado**
Universidad de La Laguna

Trabajo Final de Máster **Máster en Informática Industrial y Robótica**

Sistema de Riego Automatizado de un Huerto con IoT
Automated Vegetable Garden Irrigation System with IoT

AUTOR: Íñigo Martínez Sánchez
TUTOR: Santiago Torres Álvarez & Juan Albino Méndez Pérez
Julio 2024

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

1. TABLA DE CONTENIDOS	
1.1 Índice de figuras.....	4
1.2 Lista de acrónimos utilizados en la memoria.....	5
2. RESUMEN	6
3. ABSTRACT	7
4. INTRODUCCIÓN.....	8
4.1 Estado actual del sistema	9
4.2 Estado de la Técnica	10
4.2.1 Tecnologías Actuales en la Agricultura	10
4.2.2 Tecnologías Actuales en la Meteorología	11
4.3 Propuestas de Implementación	13
4.4 Objetivos	14
4.5 Estructura del documento	15
5. TECNOLOGÍAS UTILIZADAS	16
5.1 Hardware	16
5.2 Software	34
6. IMPLEMENTACIÓN.....	37
6.1 Implementación del Hardware.....	38
6.1.1 Circuito Adaptador de Velocidad del Viento.....	42
6.1.2 Lecturas de sensores con señales analógicas.....	45
6.1.3 Lecturas de sensores con señales digitales	46
6.2 Implementación del Software.....	47
6.2.1 Comunicación.....	54
6.2.2 Integración de Servidor con Dash y Bootstrap.....	57
7. PRUEBAS DE FUNCIONAMIENTO.....	63
8. RESULTADOS OBTENIDOS.....	69
9. CONCLUSIONES Y LÍNEAS FUTURAS.....	71
10. PRESUPUESTO.....	73
11. BIBLIOGRAFÍA.....	75
12. ANEXOS.....	77

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

1. Tabla de Contenidos

1.1 Índice de figuras

Figura 1: Programador Básico Riego	9
Figura 2: Concepto de sistema de huerto IoT.....	14
Figura 3: Placa Arduino MEGA	17
Figura 4: Shield Tornillo Arduino MEGA.....	18
Figura 5: Placa Raspberry Pi 4B	18
Figura 6: Sensor BMP180	20
Figura 7: Sensor AM2302.....	21
Figura 8: Sensor GUVA-S12SD	22
Figura 9: Sensor Viento Young 03002	23
Figura 10: Sensor XKC-Y23A	24
Figura 11: Sensor JSN-SR04T	25
Figura 12: Sensor YF-S201.....	26
Figura 13: Sensor Humedad Suelo	27
Figura 14: Válvula Solenoide 12VDC.....	28
Figura 15: Bomba STERWINS 900w	29
Figura 16: Control automático bomba	30
Figura 17: Luminaria LED 30w	31
Figura 18: Módulo Relé 5v 4 vías	32
Figura 19: Artículos para montaje de sistema de riego	33
Figura 20: Arquitectura Sistema IoT	36
Figura 21: Esquema Conexión Sensores y Actuadores	37
Figura 22: Caja Sistema Central con Display LCD	38
Figura 23: Esquema Sistema Central	39
Figura 24: Comunicación Arduino / Raspberry Pi	39
Figura 25: Level Shifter.....	40
Figura 26: Interior caja Sistema Central	40
Figura 27: Circuito Activación Electroválvula	41
Figura 28: Sensor Temperatura y Humedad	41
Figura 29: Piezoeléctrico Sensor Barométrico	42
Figura 30: Anemómetro Young 03002	42
Figura 31: Señal AC bobina anemómetro	43
Figura 32: Circuito adaptador de señal	43
Figura 33: Señal AC vs Señal Pulsada	44
Figura 34: Relación Frecuencia Señal AC / velocidad Viento.....	44
Figura 35: Preamplificador sensor humedad sustrato.....	45
Figura 36: Conexión Sensor Dirección Viento.....	45
Figura 37: Figura 37: Preamplificador sensor UV.....	46
Figura 38: Figura 38: Señal Pulsada Sensor Caudal.....	46
Figura 39: Prueba lectura de sensores.....	47
Figura 40: Señal de datos Sensor AM2302	48
Figura 41: Tabla de Índice UV Sensor GUVA-S12SD.....	49
Figura 42: Principio Funcionamiento sensor US	51
Figura 43: Sistema central instalado en pared	63
Figura 44: Navegación Menús LCD	63
Figura 45: Anemómetro Instalado	64
Figura 46: Pantalla Status del Servidor Web	65
Figura 47: Pantalla Monitor - Gráfico Temperatura	65
Figura 48: Pantalla Monitor - Gráfico Humedad	66
Figura 49: Pantalla Actuadores - EV Riego, purga, llenado y Foco	66
Figura 50: Activación de Relés	67
Figura 51: Activación EV Riego	67
Figura 52: Manguera de riego por exudación.	65
Figura 53: Evaporación según climatología (simulación).....	67

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

1.2 Lista de acrónimos utilizados en la memoria

- IoT** - Internet of Things (Internet de las Cosas)
- IA** - Inteligencia Artificial
- I2C** - Inter-Integrated Circuit (Circuito Inter-integrado)
- API** - Application Programming Interface (Interfaz de Programación de Aplicaciones)
- HTTP** - Hypertext Transfer Protocol (Protocolo de Transferencia de Hipertexto)
- HTML** - Hypertext Markup Language (Lenguaje de Marcado de Hipertexto)
- RPI** - Raspberry Pi
- Raspberry Pi** - Computadora de placa única de bajo costo desarrolladas por la Fundación Raspberry Pi y que por lo general funciona bajo SO Linux
- SO** - Sistema Operativo
- DB** - Database (Base de Datos)
- SQL** - Structured Query Language (Lenguaje de Consulta Estructurado)
- PWM** - Pulse Width Modulation (Modulación por Ancho de Pulso)
- ADC** - Analog-to-Digital Converter (Convertidor de Analógico a Digital)
- DHT** - Digital Humidity and Temperature (Humedad y Temperatura Digital)
- GUI** - Graphical User Interface (Interfaz Gráfica de Usuario)
- LAN** - Local Area Network (Red de Área Local)
- WAN** - Wide Area Network (Red de Área Amplia)
- SD** - Secure Digital (Tarjeta de Memoria)
- USB** - Universal Serial Bus (Bus Serial Universal)
- LED** - Light Emitting Diode (Diodo Emisor de Luz)
- Arduino** - Placa de microcontrolador de código abierto basada en el microcontrolador ATmega2560.
- EV** - Electroválvula.
- Raspbian** - Sistema Operativo Linux diseñado para Raspberry Pi.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

2. RESUMEN

El cambio climático y el crecimiento de la población mundial han incrementado la presión sobre los recursos hídricos, especialmente en el sector agrícola, que es uno de los mayores consumidores de agua. La agricultura sostenible se presenta como una solución necesaria para enfrentar estos desafíos, promoviendo el uso eficiente de los recursos naturales y minimizando el impacto ambiental. En este contexto, la integración de tecnologías avanzadas en la agricultura ofrece nuevas oportunidades para mejorar la eficiencia del riego y reducir el consumo de agua.

Tenerife, la mayor de las Islas Canarias, presenta un clima subtropical con una notable variabilidad en la precipitación a lo largo del año. Mientras que los inviernos pueden ser relativamente húmedos, los veranos son generalmente secos y calurosos, lo que plantea desafíos significativos para la gestión del agua en la agricultura. La escasez de aguas pluviales durante los meses de verano hace necesario el uso eficiente de los recursos hídricos disponibles para asegurar la sostenibilidad del huerto.

Este proyecto de fin de máster se centra en el desarrollo de un sistema de riego inteligente, que aprovecha el uso de las tecnologías, dispositivos y software actuales (sensores ambientales, Arduino, Raspberry Pi, Internet de las Cosas (IoT) y técnicas de programación en Arduino y Python), para la digitalización y la optimización del uso de agua en el cultivo de alimentos. El objetivo principal es implementar una solución que permita monitorizar y controlar el riego de manera precisa y remota, basándose en un análisis exhaustivo de datos climáticos y del suelo.

El sistema propuesto combina datos obtenidos de una estación meteorológica y sensores humedad de sustrato, incluyendo temperatura, humedad, dirección y velocidad del viento, variaciones en la presión atmosférica y radiación ultravioleta, entre otros. Estos datos se procesan para analizar las condiciones climáticas actuales y futuras y así determinar la cantidad óptima de agua necesaria para el riego. Además, se desarrolla una aplicación web que facilita la monitorización y el control remoto del sistema, ofreciendo una interfaz amigable para el usuario.

La implementación de este sistema no solo busca mejorar la eficiencia del riego, sino también dar la posibilidad de dar acceso a controlar y monitorizar el estado del espacio agrícola desde cualquier parte del mundo.

Además, el agua utilizada para el riego puede ser recolectada de la lluvia o provenir de fuentes de abastecimiento, lo que añade un componente adicional de sostenibilidad al proyecto.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

3. ABSTRACT

Climate change and global population growth have increased pressure on water resources, especially in the agricultural sector, which is one of the largest consumers of water. Sustainable agriculture presents itself as a necessary solution to face these challenges, promoting the efficient use of natural resources and minimizing environmental impact. In this context, the integration of advanced technologies in agriculture offers new opportunities to improve irrigation efficiency and reduce water consumption.

Tenerife, the largest of the Canary Islands, has a subtropical climate with a remarkable variability in rainfall throughout the year. While winters can be relatively wet, summers are generally dry and hot, posing significant challenges for agricultural water management. The scarcity of rainwater during the summer months necessitates the efficient use of available water resources to ensure the sustainability of the orchard.

This master's thesis project focuses on the development of a smart irrigation system, which leverages the use of current technologies, devices and software (environmental sensors, Arduino, Raspberry Pi, Internet of Things (IoT) and programming techniques in Arduino and Python), for the digitisation and optimisation of water use in food cultivation. The main objective is to implement a solution to monitor and control irrigation accurately and remotely, based on a comprehensive analysis of climate and soil data.

The proposed system combines data obtained from a weather station and soil moisture sensors, including temperature, humidity, wind direction and speed, atmospheric pressure variations and ultraviolet radiation, among others. These data are processed to analyze current and future climatic conditions and thus determine the optimal amount of water needed for irrigation. In addition, a web application is developed to facilitate the monitoring and remote control of the system, offering a user-friendly interface.

The implementation of this system not only aims to improve irrigation efficiency, but also to give the possibility to provide access to control and monitor the state of the agricultural space from anywhere in the world.

In addition, the water used for irrigation can be collected from rainfall or from water supply sources, which adds an additional component of sustainability to the project.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

4. INTRODUCCIÓN

Éste documento se estructura en varias secciones que abordan desde el estado actual del sistema de riego utilizado en el huerto, hasta las propuestas de implementación y los objetivos del proyecto. En primer lugar, se analiza el estado actual del sistema, identificando sus limitaciones y áreas de mejora. A continuación, se exploran las tecnologías modernas disponibles en el ámbito agrícola y meteorológico, que pueden ser integradas para desarrollar un sistema más eficiente y adaptable.

Finalmente, se presentan las propuestas de implementación específicas, detallando cómo estas tecnologías pueden ser aplicadas para optimizar el riego en el huerto, y se establecen los objetivos del proyecto para guiar su desarrollo y evaluación.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

4.1 Estado Actual del Sistema

El huerto que se va a utilizar para realizar la implementación de este proyecto actualmente utiliza un sistema de riego por goteo controlado por un programador básico, que emplea exclusivamente agua de abastecimiento.

Aunque este método es más eficiente que otros métodos de riego tradicionales, presenta varias limitaciones que afectan tanto a la eficiencia del uso del agua como a la salud de los cultivos. Ya que el programador básico permite establecer horarios fijos para el riego, pero no se adapta a las condiciones ambientales cambiantes. Esto resulta en un riego uniforme que no considera variables críticas como la humedad del suelo, la temperatura, o las previsiones de lluvia, lo que a menudo conduce a un uso innecesario y excesivo del agua. En periodos de lluvia, el sistema sigue funcionando, causando desperdicio de agua y problemas de sobre-irrigación que pueden dañar los cultivos.

Además, la dependencia exclusiva del agua de abastecimiento implica un costo constante y una dependencia de una fuente que puede no ser sostenible a largo plazo, especialmente en regiones donde el agua es escasa o cara. La posible capacidad para recolectar y utilizar agua de lluvia representa una oportunidad para reducir el consumo de agua de abastecimiento y mejorar la sostenibilidad del huerto.



Figura 1: Programador Básico Riego

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

4.2 Estado de la Técnica

En la actualidad, las tecnologías utilizadas para mejorar la agricultura y la ciencia de la meteorología han avanzado significativamente, integrando diversas herramientas y técnicas para optimizar la eficiencia, sostenibilidad y precisión en ambos campos. A continuación, se describen algunas de las tecnologías más relevantes:

4.2.1 Tecnologías Actuales en la Agricultura

Sensores Ambientales y de Suelo:

- Sensores de Humedad del Suelo: Miden la cantidad de agua presente en el suelo, ayudando a determinar cuándo y cuánto regar.
- Sensores de Temperatura y Humedad del Aire: Monitorizan las condiciones ambientales.
- Sensores de Nutrientes: Detectan los niveles de nutrientes en el suelo, permitiendo ajustes en la fertilización.

Drones y Satélites:

- Drones: Utilizados para monitorear cultivos, realizar mapeo del terreno, o aplicar pesticidas y fertilizantes de manera precisa.
- Satélites: Proporcionan imágenes de alta resolución para la monitorización de grandes extensiones de tierra, ayudando en la gestión de cultivos y predicción de rendimientos.

Sistemas de Riego Inteligente:

- Riego por Goteo Automatizado: Controlado por sensores y programadores inteligentes que ajustan el riego en función de las condiciones del suelo y clima.
- Sistemas de Gestión del Agua: Utilizan datos en tiempo real para optimizar el uso del agua, reduciendo el desperdicio.

Internet de las Cosas (IoT):

- Dispositivos IoT: Conectan sensores, actuadores y sistemas de control, permitiendo la recolección de datos y el control remoto desde cualquier parte del mundo.
- Plataformas de Gestión: herramientas de software diseñadas para ayudar a las organizaciones a gestionar diversos aspectos de sus operaciones de manera más eficiente.

10

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

Big Data y Análisis Predictivo:

- Modelos de Análisis: Utilizan datos históricos y en tiempo real para predecir rendimientos de cultivos, necesidades de riego y aparición de plagas...
- Machine Learning: Algoritmos que mejoran continuamente la precisión de las predicciones y recomendaciones agrícolas.

4.2.2 Tecnologías Actuales en la Meteorología

Estaciones Meteorológicas:

- Estaciones Automáticas: Miden variables como temperatura, humedad, velocidad y dirección del viento, presión atmosférica, y radiación solar.
- Redes de Estaciones: Proporcionan datos a gran escala, mejorando las predicciones climáticas regionales y globales.

Modelos Climáticos:

- Modelos Numéricos: Simulan el comportamiento de la atmósfera, los océanos y la superficie terrestre para prever el clima futuro.
- Sistemas de Predicción a Corto y Largo Plazo: Ofrecen pronósticos meteorológicos diarios y predicciones climáticas a largo plazo.

Satélites Meteorológicos:

- Observación por Satélite: Monitorean variables climáticas globales, como la temperatura del mar, el nivel de hielo, y la concentración de gases de efecto invernadero.
- Imágenes en Tiempo Real: Ayudan en la detección de fenómenos meteorológicos extremos y en la gestión de desastres.

Sensores Remotos:

- LIDAR y RADAR: Utilizados para medir la altitud de las nubes, la precipitación y la estructura de las tormentas.
- Sondas Atmosféricas: Recopilan datos desde diferentes capas de la atmósfera, proporcionando información crucial para los modelos climáticos.

11

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

Supercomputadoras:

- Simulaciones Climáticas: Ejecutan complejos modelos numéricos que requieren grandes capacidades de procesamiento.
- Análisis de Big Data: Manejan y analizan enormes volúmenes de datos para mejorar la precisión de las predicciones climáticas.

Estas tecnologías, combinadas con técnicas avanzadas de análisis de datos, están revolucionando tanto la agricultura como la ciencia del clima, permitiendo una gestión más precisa y eficiente de los recursos y mejorando nuestra capacidad para adaptarnos a las condiciones cambiantes del medio ambiente.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

4.3 Propuestas de implementación

Gracias a las técnicas anteriormente descritas, para mejorar el sistema de riego actual, es necesario agregar y modificar diferentes componentes que optimicen el funcionamiento bajo las condiciones reales del entorno. La integración de sensores que monitorean variables ambientales en tiempo real, combinados con otros sensores que monitorean el estado del suelo y el sistema de riego, y una plataforma de control remota, nos permitiría crear un sistema más preciso y eficiente. Además, la incorporación de sistemas de recolección de agua de lluvia podría reducir la dependencia del agua de abastecimiento, promoviendo una práctica más sostenible y económica. Por último, el uso de algoritmos de inteligencia artificial para analizar los datos recopilados por los sensores puede ofrecer recomendaciones automatizadas para ajustar los parámetros del riego, garantizando así un uso óptimo de los recursos hídricos y mejorando la salud y el rendimiento de los cultivos.

Por todo ello se propone la incorporación de sensores ambientales y de suelo que nos permita una monitorización más precisa de las condiciones del campo. Estos sensores podrán medir variables críticas como la humedad del suelo, la temperatura, la radiación solar, la velocidad y dirección del viento, y la presión atmosférica, proporcionando datos en tiempo real que pueden ser utilizados para tomar decisiones informadas sobre el riego.

La integración de un sistema con Internet de las Cosas (IoT) nos facilitará la interconexión de estos dispositivos, permitiendo la creación de sistemas integrados que recopilan, transmiten y analizan datos de manera continua. Además nos permitirá activar diferentes actuadores de forma remota. Para ello haremos uso de plataformas de hardware como Arduino y Raspberry Pi que jugarán un papel crucial en esta transformación, ofreciendo soluciones accesibles y personalizables para el desarrollo de sistemas de riego inteligentes.

En paralelo, el uso de lenguajes de programación como Python, nos permitirá el desarrollo de una aplicación web con una interfaz gráfica que nos permita conectar con el sistema de forma amigable. Estas tecnologías nos permitirán en un futuro implementar nuevas herramientas que añadan aún más valor al sistema.

Se trata de una solución prometedora para enfrentar los desafíos actuales de la agricultura y asegurar la sostenibilidad de los recursos hídricos a largo plazo. Además de facilitar las labores de este sector primario.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

4.4 Objetivos

Los siguientes objetivos que se proponen para este proyecto, buscan servir como un modelo replicable para otras áreas agrícolas, contribuyendo a la adopción de prácticas más sostenibles y eficientes en las labores y gestión de los recursos naturales.

Monitorización del Entorno:

- Instalar y calibrar una red de sensores ambientales y de suelo para medir variables críticas como la humedad del suelo, la temperatura, la humedad del aire, el viento y la radiación solar en tiempo real.

Monitorización del consumo de recursos hídricos:

- Instalar y calibrar una red de sensores que permitan monitorizar los diferentes parámetros relacionados con el consumo de recursos hídricos. Cantidad de agua recolectada en aljibe mediante sensor de nivel, así como los consumos de agua realizados mediante el riego.

Desarrollo de una Aplicación Web (IoT):

- Desarrollar un sistema IoT que nos permita controlar un sistema de riego por exudación permitiendo activar y desactivar diferentes actuadores de forma remota, a través de una aplicación web con una interfaz gráfica amigable. Además de monitorizar las lecturas de los sensores desde cualquier parte del mundo.

Integración de Fuentes de Agua Alternativas:

- Incorporar la capacidad de recolectar agua de lluvia y dar la opción de seleccionar la fuente de agua que se utilizará para el riego (recolectada o de abastecimiento), reduciendo la dependencia del agua y promoviendo una gestión más sostenible de los recursos hídricos.

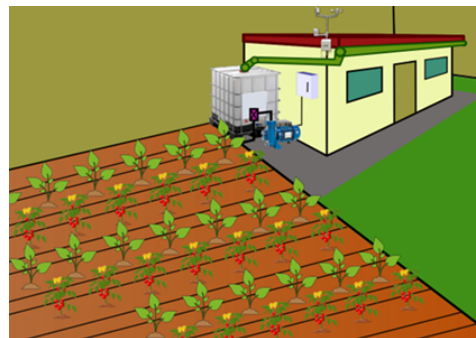


Figura 2: Concepto de sistema de huerto IoT

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

4.5 Estructura del documento

Después de una introducción a los temas principales que se tratarán en esta memoria y plantear los principales objetivos que se tratarán de llevar a cabo en este proyecto, procederemos a definir la estructura y los apartados de las siguientes páginas del documento, ofreciendo una guía clara y detallada del contenido que se desarrollará.

En primer lugar, se presentarán las tecnologías y componentes que se utilizarán para la elaboración del sistema de riego inteligente, diferenciando claramente entre el hardware y el software utilizado. Se detallará exhaustivamente cada uno de los componentes físicos, como sensores, actuadores, y las plataformas de desarrollo (Arduino y Raspberry Pi), explicando su función específica dentro del sistema.

En cuanto al software, se proporcionará una descripción pormenorizada de las partes del código utilizadas para el manejo de sensores y actuadores, y por otro lado, las partes de código importantes para llevar a cabo la implementación de la interfaz de usuario de la aplicación web. Cada sección de código será explicada con claridad, destacando los pasos seguidos en su desarrollo e integración.

Seguidamente, se abordará la implementación del sistema, describiendo el proceso paso a paso, desde la instalación de los componentes hardware hasta la configuración y programación del software. Esta sección incluirá diagramas de circuitos, esquemas de conexión y capturas de pantalla de la interfaz web, proporcionando una visión completa del proceso de desarrollo.

Una vez implementado el sistema, en el apartado 7 se procederá a las pruebas de funcionamiento, donde se evaluará su desempeño bajo diferentes condiciones ambientales y de cultivo. Los resultados obtenidos de estas pruebas serán presentados y analizados en detalle, comparándolos con los métodos de riego tradicionales para destacar las mejoras en términos de eficiencia y ahorro de agua.

A continuación, en el apartado 8 se discutirán las conclusiones del proyecto, resaltando los logros alcanzados y las ventajas del sistema de riego inteligente en la gestión de recursos hídricos. También se identificarán las limitaciones encontradas durante el desarrollo e implementación, y se propondrán líneas futuras de investigación y mejora que podrían implementarse para optimizar aún más el sistema.

Para concluir, se presentará un presupuesto completo de la implementación del sistema, desglosando los costos de cada componente y etapa del proyecto. Esta sección permitirá evaluar la viabilidad económica del sistema y su potencial aplicación en diferentes contextos agrícolas.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

Finalmente, se incluirá una bibliografía de las fuentes consultadas para el desarrollo del proyecto. En los anexos, se proporcionarán los códigos completos de los diferentes sistemas implementados, ofreciendo recursos adicionales para aquellos interesados en replicar o mejorar el sistema desarrollado.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

5. TECNOLOGÍA UTILIZADA

El montaje del sistema de riego inteligente se basa en una combinación de hardware y software cuidadosamente seleccionada para optimizar la eficiencia y la funcionalidad del riego. A continuación, se detallan las tecnologías empleadas, diferenciando entre los componentes de hardware y software.

5.1 HARDWARE

- Placa Arduino MEGA

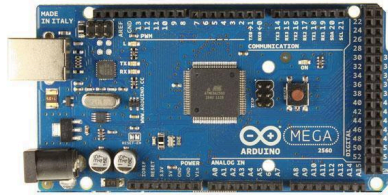


Figura 3: Placa Arduino MEGA (Imagen comercial)

El **Arduino Mega** compuesta por un microcontrolador ATmega2560, está diseñado para proyectos que requieren una gran cantidad de pines de entrada/salida y una mayor capacidad de memoria. Es ideal para aplicaciones complejas y exigentes que involucran múltiples sensores, actuadores y periféricos.

Características Principales

- **Microcontrolador:** ATmega2560.
- **Pines de E/S:** 54 pines digitales de entrada/salida (15 de los cuales pueden usarse como salidas PWM), 16 entradas analógicas y 4 UARTs (puertos serie hardware).
- **Memoria:** 256 KB de memoria flash para almacenar programas, 8 KB de SRAM y 4 KB de EEPROM.
- **Voltaje de Operación:** 5V.
- **Conectividad:** Incluye un conector USB para programación y comunicación con un ordenador, un conector de alimentación externa, un conector ICSP y un botón de reset.
- **Frecuencia de reloj:** 16 MHz.

Aplicación

El uso que le daremos al Arduino en este proyecto será el de interconectar los diferentes sensores y actuadores a utilizar, así como el resto de componentes de hardware (display LCD, pulsadores, ...). Además nos dará la opción de establecer una comunicación fluida con la Raspberry Pi a través de uno de los puertos Serial.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

- Shield para Arduino MEGA de terminales de tornillo

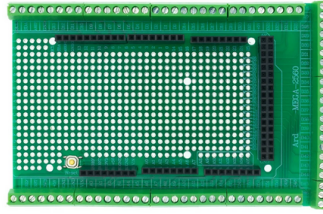


Figura 4: Shield Tornillo Arduino MEGA (Imagen comercial)

El **Arduino Mega Screw Terminal Shield** es una expansión diseñada específicamente para la placa Arduino Mega, que facilita las conexiones mediante terminales de tornillo. Este shield convierte los pines del Arduino Mega en terminales de tornillo robustos, permitiendo conexiones seguras de los cables. Además nos permite soldar diferentes componentes en la placa de puntos que contiene en su interior.

Características Principales

- **Terminales de Tornillo:** Proporciona terminales de tornillo para cada pin del Arduino Mega, lo que permite conexiones firmes y seguras.
- **Compatibilidad:** Diseñado específicamente para ajustarse a la placa Arduino Mega.
- **Acceso Completo a Pines:** Todos los pines digitales y analógicos del Arduino Mega están disponibles a través de terminales de tornillo.
- **Facilidad de Uso:** Simplifica la conexión de cables y componentes.
- **Robustez:** Para entornos donde las conexiones necesitan ser más duraderas y resistentes a manipulaciones.

Aplicación

El uso que le daremos en este proyecto es el de facilitar el conexionado de los cables de los diversos periféricos que se conectarán al Arduino. Además el que tenga una PCB de puntos en su interior nos dará la opción de formar circuitos auxiliares en ella.

- Placa Raspberry Pi 4B

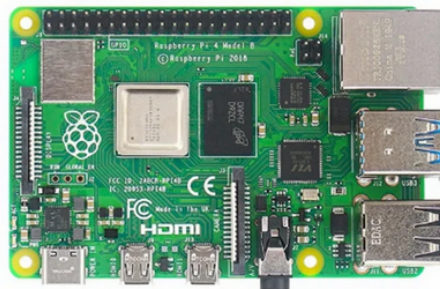


Figura 5: Placa Raspberry Pi 4B (Imagen comercial)

18

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970

Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez
UNIVERSIDAD DE LA LAGUNA

Fecha: 20/08/2024 23:04:24

Santiago Torres Álvarez
UNIVERSIDAD DE LA LAGUNA

20/08/2024 23:05:12

Juan Albino Méndez Pérez
UNIVERSIDAD DE LA LAGUNA

26/08/2024 12:55:49

La **Raspberry Pi 4 Model B** es una potente y versátil computadora de placa única que ofrece un rendimiento aceptable en relación a su tamaño. Diseñada para una amplia gama de aplicaciones, desde la educación hasta proyectos de IoT y automatización industrial, esta versión con 2GB de RAM es ideal para proyectos que requieren un equilibrio entre capacidad de procesamiento y eficiencia de costos.

Características Principales

- **Procesador:** Broadcom BCM2711, Cortex-A72 (ARM v8) de 64 bits a 1.5GHz, de cuatro núcleos.
- **Memoria RAM:** 2GB LPDDR4-3200 SDRAM.
- **Conectividad:**
 - **Wi-Fi:** 802.11ac de doble banda (2.4GHz y 5.0GHz).
 - **Bluetooth:** 5.0, BLE.
 - **Ethernet:** Gigabit Ethernet.
- **Puertos USB:** 2 puertos USB 3.0 y 2 puertos USB 2.0.
- **Salidas de Video:** 2 puertos micro-HDMI que soportan hasta 4K a 60 fps.
- **Almacenamiento:** Ranura para tarjeta microSD para almacenamiento y sistema operativo.
- **GPIO:** 40 pines GPIO para conexión a sensores, actuadores y otros dispositivos.
- **Audio:** Salida de audio estéreo y soporte para audio digital a través de HDMI.
- **Alimentación:** Puerto USB-C para alimentación, con una demanda de 5V/3A.

Aplicación

En este proyecto la Raspberry Pi se utilizará como centro de procesamiento y control del sistema de riego inteligente. Su aplicación incluirá la recepción y procesamiento de datos en tiempo real desde el Arduino Mega, la ejecución de un servidor local con una interfaz web interactiva mediante el uso de herramientas de programación como Python y Dash. Además, la Raspberry Pi enviará comandos al Arduino Mega para activar y desactivar los diferentes actuadores del sistema de riego. Esta integración centralizada permitirá una gestión eficiente y adaptativa del sistema.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

- Sensor de Presión Barométrica BMP180

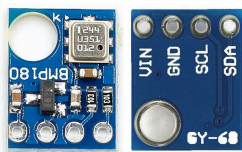


Figura 6: Sensor BMP180 (Imagen Comercial)

El **BMP180** es un sensor digital de presión barométrica y temperatura desarrollado por Bosch Sensortec. Es conocido por su alta precisión, bajo consumo de energía y tamaño compacto, lo que lo hace ideal para aplicaciones móviles y proyectos de electrónica como estaciones meteorológicas.

Características Principales

- **Medición de Presión:** Capaz de medir la presión barométrica en un rango de 300 hPa a 1100 hPa.
- **Medición de Temperatura:** Integra un sensor de temperatura que mide en un rango de -40°C a $+85^{\circ}\text{C}$.
- **Precisión:** Alta precisión con una resolución de hasta 0.02 hPa.
- **Consumo de Energía:** Muy bajo consumo de energía, adecuado para dispositivos alimentados por batería.
- **Interfaz:** Comunicación a través de interfaces I2C o SPI, lo que facilita la integración con microcontroladores como Arduino y Raspberry Pi.
- **Calibración:** Viene pre-calibrado de fábrica, lo que simplifica su uso en proyectos.

Aplicaciones

En este proyecto, el sensor BMP180 se utilizará para medir la presión atmosférica y la temperatura ambiente en el huerto. La información proporcionada por el BMP180 será crucial para predecir en un futuro los cambios climáticos. La presión atmosférica puede indicar la aproximación de lluvias o cambios significativos en el clima, permitiéndonos decidir el uso de recursos hídricos que utilizaremos en nuestro sistema.

- Sensor de Temperatura y Humedad AM2302



Figura 9: Sensor AM2302 (Imagen Comercial)

El **AM2302**, también conocido como **DHT22**, es un sensor digital de alta precisión para medir la temperatura y la humedad del ambiente. Es ampliamente utilizado en aplicaciones que requieren monitoreo ambiental debido a su fiabilidad, precisión y facilidad de uso.

Características Principales

- **Medición de Humedad:** Capaz de medir la humedad relativa en un rango de 0% a 100% con una precisión del $\pm 2-5\%$.
- **Medición de Temperatura:** Mide la temperatura en un rango de -40°C a $+80^{\circ}\text{C}$ con una precisión de $\pm 0.5^{\circ}\text{C}$.
- **Salida Digital:** Proporciona una salida digital calibrada, eliminando la necesidad de componentes adicionales para el procesamiento de señales.
- **Interfaz:** Comunicación a través de un solo pin digital, lo que simplifica la integración con microcontroladores como Arduino.
- **Calibración:** Pre-calibrado de fábrica

- Aplicación

En este proyecto, el sensor AM2302 se utilizará para medir la humedad y la temperatura ambiente en el huerto. La información precisa y en tiempo real proporcionada por el AM2302 nos permitirá tomar decisiones de ajuste del riego de manera óptima según las condiciones ambientales actuales.

- Sensor de Radiación UV GUVA-S12SD

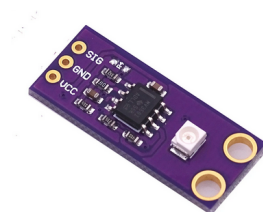


Figura 8: Sensor GUVA-S12SD (Imagen Comercial)

El **GUVA-S12SD** es un sensor de radiación ultravioleta (UV) que mide la intensidad de la luz UV en el rango de 200 a 370 nm. Es ampliamente utilizado en aplicaciones que requieren monitoreo de la exposición a los rayos UV, como estaciones meteorológicas, dispositivos portátiles y sistemas de salud ambiental.

Características Principales

- **Rango de Medición:** Capaz de detectar luz UV en el rango de 200 nm a 370 nm.
- **Salida Analógica:** Proporciona una salida de voltaje analógica proporcional a la intensidad de la luz UV detectada.
- **Alta sensibilidad:** Detecta bajos niveles de radiación UV con alta precisión.
- **Bajo Consumo de Energía:** Ideal para dispositivos alimentados por batería.
- **Tamaño Compacto:** Fácil de integrar en proyectos donde el espacio es limitado.

Aplicación

En este proyecto el sensor GUVA-S12SD se utilizará para medir la radiación ultravioleta (UV) en el huerto. La información sobre la intensidad de la radiación UV nos permitirá intuir la velocidad de evaporación de la humedad del sustrato según las condiciones de luz solar, ya que una alta radiación puede aumentar la evaporación del agua del suelo y las plantas, incrementando su necesidad de agua.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

- Sensor de Viento Young, Modelo 03002



Figura 9: Sensor Viento Young 03002 (Imagen Comercial)

El **Wind Sentry 03002** de Young es un sensor de viento que mide tanto la velocidad como la dirección del viento. Este sensor es ampliamente utilizado en aplicaciones meteorológicas y medioambientales debido a su precisión, durabilidad y fiabilidad en condiciones extremas.

Características Principales

- **Medición de Velocidad del Viento:** Utiliza un anemómetro de cazoletas para medir la velocidad del viento en un rango de 0 a 50 m/s (0 a 180 Km/h) con una precisión de ± 0.5 m/s.
- **Medición de Dirección del Viento:** Emplea una veleta para determinar la dirección del viento en un rango de 360° , con una resolución de 5° y una precisión de $\pm 5^\circ$.
- **Construcción Duradera:** Fabricado con materiales resistentes a la corrosión, adecuado para operar en condiciones climáticas adversas.
- **Salida Analógica:** Proporciona señales analógicas proporcionales a la velocidad y dirección del viento, facilitando la integración con sistemas de adquisición de datos y microcontroladores haciendo uso de ADCs.
- **Bajo Mantenimiento:** Diseño robusto que requiere un mantenimiento mínimo.

Aplicación

En este proyecto el anemómetro se utilizará para medir la velocidad y dirección del viento en el huerto, ya que el viento puede influir significativamente en la evaporación del agua y la dispersión de la humedad. Al conocer las condiciones del viento, podremos decidir posponer el riego

para evitar desperdicios en días ventosos y asegurar que el agua aplicada llegue eficazmente a las plantas.

- Sensor de Nivel/Presencia de Líquido XKC-Y23A



Figura 10: Sensor XKC-Y23A (Imagen Comercial)

El **XKC-Y23A** es un sensor de nivel de líquido sin contacto que utiliza tecnología capacitiva para detectar la presencia de líquido a través de la pared de un contenedor. Es especialmente útil en aplicaciones donde el contacto directo con el líquido no es deseable o posible.

Características Principales

- **Detección Sin Contacto:** Capaz de detectar líquidos a través de la mayoría de los materiales no metálicos, como plástico o vidrio, con un grosor de hasta 20 mm.
- **Alta Sensibilidad:** Ofrece una detección precisa del nivel de líquido, evitando falsos positivos y negativos.
- **Fácil de Usar:** Instalación sencilla y sin necesidad de perforar el contenedor. Se monta externamente al tanque o tubería.
- **Salida Digital:** Proporciona una señal de salida digital que indica la presencia o ausencia de líquido, facilitando la integración con microcontroladores como Arduino.
- **Resistente al Agua:** Diseño impermeable, para entornos húmedos o aplicaciones donde puede haber salpicaduras.
- **Bajo Consumo de Energía:** Ideal para sistemas alimentados por baterías.

Aplicación

El sensor capacitivo de nivel de agua XKC-Y23A se utilizará para detectar la presencia de agua en las tuberías de riego, protegiendo la bomba de posibles daños por funcionamiento en seco. Este sensor asegurará que siempre haya suficiente agua en el sistema antes de activar la bomba, evitando su operación cuando las tuberías están vacías.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

- Sensor de Ultrasonidos JSN-SR04T



Figura 11: Sensor JSN-SR04T (Imagen Comercial)

El **JSN-SR04T** es un sensor de distancia ultrasónico resistente al agua, diseñado para medir distancias utilizando ondas ultrasónicas. Es conocido por su durabilidad y capacidad para operar en ambientes húmedos y exteriores, lo que lo hace ideal para una amplia gama de aplicaciones.

Características Principales

- **Medición de Distancia:** Capaz de medir distancias en un rango de 20 cm a 600 cm (0.2 m a 6 m) con una precisión de aproximadamente 2 cm.
- **Resistente al Agua:** Su diseño impermeable lo hace adecuado para aplicaciones en exteriores y entornos húmedos.
- **Salida Digital:** Proporciona una señal de salida digital que indica la distancia medida, facilitando la integración con microcontroladores como Arduino.
- **Bajo Consumo de Energía:** Para aplicaciones de bajo consumo energético, adecuado para sistemas alimentados por baterías.
- **Interfaz Fácil de Usar:** Utiliza una interfaz de 4 pines (VCC, Trig, Echo, GND).

Aplicación

El sensor de ultrasonido JSN-SR04T se utilizará para medir el nivel de agua en el depósito de riego. Este sensor proporciona datos precisos y en tiempo real sobre la cantidad de agua disponible en el depósito, permitiéndonos gestionar eficientemente los recursos hídricos. El monitorear continuamente el nivel del agua, también nos proporcionará información sobre la cantidad de agua recolectada en un espacio de tiempo determinado.

- Sensor de caudal YF-S201



Figura 12: Imagen Comercial Sensor YF-S201

El **YF-S201** es un sensor de flujo de agua que mide la cantidad de agua que pasa a través de él mediante la generación de pulsos eléctricos. Es ampliamente utilizado en sistemas de riego, electrodomésticos y otras aplicaciones que requieren monitoreo preciso del flujo de agua.

Características Principales

- **Medición de Flujo:** Capaz de medir flujos de agua en el rango de 1 a 30 litros por minuto (L/min).
- **Salida de Pulsos:** Proporciona una señal de salida en forma de pulsos digitales que pueden ser contados por un microcontrolador para calcular el flujo de agua.
- **Alta Precisión:** Ofrece una precisión de $\pm 10\%$ en la medición del flujo.
- **Material Duradero:** Fabricado con materiales resistentes a la corrosión, adecuado para uso en interiores y exteriores.
- **Fácil Integración:** Compatible con microcontroladores como Arduino.

Aplicación

El sensor de flujo YF-S201 se utilizará para medir el caudal de agua que pasa a través del sistema de riego. Este sensor proporcionará datos en tiempo real sobre el volumen de agua utilizado, permitiéndonos calcular el consumo de agua de manera precisa en cada ciclo de riego. Además, el YF-S201 nos ayudará a identificar posibles fugas o irregularidades en el flujo.

- Sensor capacitivo de Humedad del sustrato



Figura 13: Sensor Humedad Suelo (Imagen Comercial)

Es un dispositivo diseñado para medir la humedad del suelo mediante la detección de cambios en la capacitancia del suelo. A diferencia de los sensores resistivos, los sensores capacitivos no se corroen y ofrecen una mayor durabilidad y precisión en las mediciones.

Características Principales

- **Medición Capacitiva:** Utiliza un método de detección capacitiva que no se ve afectado por la corrosión **galvánica o electroquímica**), lo que aumenta la vida útil del sensor.
- **Alta sensibilidad:** Capaz de detectar cambios en la humedad del suelo con alta precisión.
- **Salida Analógica:** Proporciona una señal de salida analógica que varía en función de la humedad del suelo, fácil de leer con microcontroladores como Arduino.
- **Bajo Consumo de Energía:** Para aplicaciones de bajo consumo energético, adecuado para sistemas alimentados por baterías.

Aplicación

El sensor se utilizará para medir la cantidad de humedad presente en el suelo del huerto. Este sensor proporciona datos precisos sobre la humedad del suelo, permitiéndonos tomar decisiones en el ajuste de tiempos y ciclos de riego según las necesidades reales de las plantas. Al monitorizar continuamente la humedad del sustrato, podremos visualizar en un gráfico a qué velocidad está sucediendo la evaporación del agua en el sustrato anticipándonos a que esto suceda.

27

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez
UNIVERSIDAD DE LA LAGUNA

Fecha: 20/08/2024 23:04:24

Santiago Torres Álvarez
UNIVERSIDAD DE LA LAGUNA

20/08/2024 23:05:12

Juan Albino Méndez Pérez
UNIVERSIDAD DE LA LAGUNA

26/08/2024 12:55:49

Válvula Solenoide Normalmente cerrada 12VDC



Figura 14: Válvula Solenoide (Imagen Comercial)

Es un dispositivo electromecánico utilizado para controlar el flujo de líquidos. Al ser "normalmente cerrada", la válvula permanece cerrada cuando no se aplica corriente eléctrica, impidiendo el paso del fluido. Solo se abre cuando se energiza con una corriente de 12 voltios de corriente continua (VDC), permitiendo el paso del fluido.

Características Principales

- **Estado Normal:** Cerrada (NC - Normally Closed).
- **Voltaje de Operación:** 12 VDC.
- **Material de Construcción:** Plástico resistente a la corrosión, adecuado para su uso exterior.
- **Conexiones:** Roscas estándar 1/2", 3/4", 1" que facilitan su integración en sistemas de tuberías y riego.
- **Consumo de Energía:** Diseñada para un funcionamiento prolongado con bajo consumo.

Aplicación

En este proyecto las electroválvulas desempeñarán un papel crucial del sistema de riego. Las principales aplicaciones de las electroválvulas son las siguientes:

- Activación del riego.
- Rellenado del depósito de agua con abastecimiento de la red.
- Purga y recirculación del agua sucia.
- Segmentación de riego, activando distintas vías.

Bomba de superficie STERWINS 900w



Figura 15: Bomba STERWINS 900w (Imagen Comercial)

La **Bomba de Superficie Sterwins 900W** es una bomba eléctrica de alta eficiencia diseñada para el manejo de agua en aplicaciones de riego, suministro de agua doméstica y otras tareas relacionadas con el bombeo de líquidos. Con una potencia de 900 vatios, esta bomba ofrece un buen rendimiento para diversas necesidades de bombeo.

Características Principales

- **Potencia:** 900 vatios.
- **Caudal:** Alta capacidad de caudal, máximo 3800 L/hora.
- **Altura de Elevación:** Capaz de elevar el agua a alturas de hasta 43m a una presión de 4.2 bares, lo que la hace adecuada para riego y suministro de agua en terrenos variados.
- **Construcción:** Fabricada con materiales duraderos y resistentes a la corrosión, garantizando una larga vida útil.
- **Fuente de Alimentación:** Funciona con corriente monofásica, lo que facilita su instalación.
- **Portabilidad:** Diseño compacto y manejable que permite su fácil transporte y colocación en diferentes entornos.
- **Conexiones:** Equipado con conexiones estándar de 1" para una fácil integración con sistemas de riego y de uso comercial.

Aplicación

La bomba STERWINS 900W se utilizará para asegurar un suministro de agua constante y eficiente al sistema de riego por exudación.

Control automático para bomba STERWINS



Figura 16: Control automático bomba (Imagen Comercial)

El **Control Automático para Bomba STERWINS** es un dispositivo diseñado para gestionar el funcionamiento de la bomba de agua, proporcionando un control automatizado y seguro del bombeo. Este dispositivo mejora la operatividad y la eficiencia de las bombas de superficie, como la Sterwins 900W.

Características Principales

- **Automatización del Encendido y Apagado:** Controla automáticamente el encendido y apagado de la bomba en respuesta a la demanda de agua, eliminando la necesidad de intervención manual.
- **Protección contra Funcionamiento en Seco:** Incluye protección para evitar que la bomba funcione sin agua, lo que protege el motor y extiende la vida útil del equipo.
- **Presostato Integrado:** Regula la presión del agua en el sistema, asegurando un suministro constante de agua.
- **Indicadores LED:** Proporciona una visualización clara del estado operativo del sistema, facilitando el monitoreo y diagnóstico.
- **Compatibilidad:** Diseñado para ser compatible con una amplia gama de bombas de superficie, incluyendo las bombas STERWINS.
- **Instalación Sencilla:** Fácil de instalar y configurar.

Aplicación

El controlador nos permitirá la activación de la bomba cuando se detecte una llave o electroválvula abierta, indicando una demanda de riego. Al monitorear la presión, el sistema pondrá en marcha la bomba sólo cuando sea necesario. Además, protegerá la bomba contra el funcionamiento en seco, deteniendo automáticamente en caso de que no haya suficiente agua en las tuberías, evitando daños potenciales. Este control asegura un funcionamiento correcto del sistema de riego, respondiendo dinámicamente a las necesidades de agua del huerto.

30

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez
UNIVERSIDAD DE LA LAGUNA

Fecha: 20/08/2024 23:04:24

Santiago Torres Álvarez
UNIVERSIDAD DE LA LAGUNA

20/08/2024 23:05:12

Juan Albino Méndez Pérez
UNIVERSIDAD DE LA LAGUNA

26/08/2024 12:55:49

- Luminaria LED V-TAC VT-4934-W de Exterior 30W



Figura 17: Luminaria LED 30w (Imagen Comercial)

El **V-TAC VT-4934-W Proyector LED 30W SMD** es una luminaria para aplicaciones exteriores. Con una potencia de 30 vatios. Apta para iluminar áreas exteriores.

Características Principales

- **Potencia:** 30 vatios.
- **Eficiencia Lumínica:** 85 lúmenes por vatio (LM/W).
- **Ángulo de Iluminación:** Amplio ángulo de iluminación de 110°.
- **Temperatura de Color:** Luz fría de 6400K, adecuada para aplicaciones que requieren una iluminación clara y brillante.
- **Construcción:** Carcasa de aluminio con aletas para una mejor disipación del calor, asegurando la durabilidad.
- **Resistencia a la Intemperie:** Clasificación IP65, lo que lo hace resistente al polvo y al agua, adecuado para uso en exteriores en diversas condiciones climáticas.
- **Tecnología LED SMD:** Proporciona una iluminación eficiente y de alta calidad con una vida útil prolongada.

Aplicación

El uso que se le pretende dar a este dispositivo, es el de disponer de una buena iluminación sobre todo por las noches o en horarios de invierno donde la luz escasea.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

- Módulo Relé de 5V para Arduino de 4 Vías

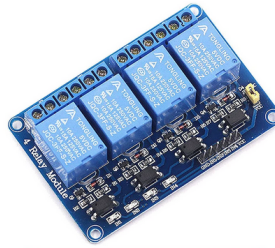


Figura 18: Módulo Relé 5v 4 vías (Imagen Comercial)

Es un componente esencial para proyectos de automatización y control, permitiendo a microcontroladores como Arduino manejar dispositivos de alto voltaje y corriente, como motores, luces y otros aparatos eléctricos. Este módulo está diseñado para ser fácil de integrar y utilizar en una variedad de aplicaciones.

Características Principales

- **Voltaje de Operación:** 5VDC, adecuado para la mayoría de microcontroladores, incluyendo Arduino.
- **Número de Relés:** 4 vías, lo que permite controlar hasta cuatro dispositivos independientes.
- **Capacidad de Contacto:** Cada relé puede manejar hasta 10A a 250V AC o 10A a 30V DC, proporcionando la capacidad de controlar dispositivos de alta potencia.
- **Aislamiento Óptico:** Protege el microcontrolador de posibles picos de voltaje y corriente parasitarias, asegurando un funcionamiento seguro.
- **Indicadores LED:** Cada relé tiene un LED indicador que muestra el estado de encendido/apagado, facilitando el monitoreo visual.
- **Interfaz de Control:** Se controla mediante señales de bajo voltaje (5V DC), que pueden ser proporcionadas directamente por los pines de salida digital de un Arduino u otro microcontrolador.

Aplicación

En este proyecto, el módulo de relés de 4 salidas se utilizará para controlar diversos componentes del sistema de riego.. Las principales utilidades del módulo de relés serán:

- Activación y desactivación de las válvulas de riego, recirculación y llenado del bidón con abastecimiento de agua.
- Activación de la luminaria de exterior LED.

Componentes de Riego (Miscelánea)

Para un sistema de riego eficiente, se utilizan diversos componentes no eléctricos ni informáticos que aseguran el suministro adecuado y controlado de agua.



Figura 19: Artículos para montaje de sistema de riego

Estos incluyen:

- **Depósito de Agua:** Contenedor grande utilizado para almacenar agua de lluvia.
- **Manguera de Riego por Exudación:** Tubo perforado que permite el riego lento y uniforme directamente a las raíces de las plantas.
- **Codos de Manguera:** Conectores angulares que permiten cambiar la dirección de la manguera de riego.
- **T de manguera:** Conectores en forma de T que permiten bifurcar el agua de riego por diferentes mangueras.
- **Llaves de Paso:** Válvulas manuales que controlan el flujo de agua en diferentes secciones del sistema de riego.
- **Adaptadores de Rosca:** Conectores que permiten unir mangueras a otros componentes con rosca, como grifos y válvulas.
- **Filtro de Agua:** Dispositivo que elimina partículas y sedimentos del agua, protegiendo los componentes del sistema de obstrucciones.

Estos componentes son esenciales para construir un sistema de riego funcional y duradero, facilitando la distribución eficiente del agua en el huerto.

5.2 SOFTWARE

A continuación se definirá una lista de las tecnologías de software utilizadas en este proyecto, junto con una breve descripción de cada una:

- **Programación en Arduino IDE**

Arduino IDE es un entorno de desarrollo integrado (IDE) utilizado para escribir, compilar y cargar programas en microcontroladores de la plataforma Arduino. Utiliza una variante del lenguaje de programación C/C++.

Usos en el proyecto:

- ❖ Control de sensores y actuadores.
- ❖ Comunicación Serial con la Raspberry Pi.
- ❖ Lectura de datos de sensores (DHT, BMP085, anemómetro, sensor UV).
- ❖ Control de actuadores (riego, purgado, llenado, foco).
- ❖ Interacción con un display LCD I2C.

- **Programación en Python**

Python es un lenguaje de programación interpretado, de alto nivel y propósito general. Es conocido por su legibilidad, compatibilidad y facilidad de uso.

Usos en el proyecto:

- ❖ Interacción con el Arduino a través de la comunicación serial.
- ❖ Procesamiento de datos recibidos del Arduino.
- ❖ Creación de un servidor web local para visualizar y controlar el sistema IoT.

- **Protocolo de Comunicación Serial**

Comunicación Serial es un método de transmisión de datos en el que los bits se envían secuencialmente a través de un único canal.

Usos en el proyecto:

- ❖ Comunicación entre el Arduino y la Raspberry Pi.
- ❖ Envío y recepción de datos de sensores y comandos de control.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

- **Protocolo de Comunicación I2C**

I2C (Inter-Integrated Circuit) es un protocolo de comunicación serial que permite la conexión de múltiples dispositivos en un solo bus de datos.

Usos en el proyecto:

- ❖ Interacción con el display LCD I2C desde el Arduino.
- ❖ Comunicación con sensor barométrico.
- ❖ Posibilidad de conectar otros sensores y actuadores compatibles con I2C.

- **Creación de un Servidor Web Local con Dash y Bootstrap**

Dash es un framework de Python para construir aplicaciones web analíticas e interactivas. Está construido sobre Flask, Plotly.js y React.js.

Bootstrap es un framework de CSS de código abierto dirigido al desarrollo de sitios web y aplicaciones web responsive y móviles.

Usos en el proyecto:

- ❖ Creación de una interfaz gráfica de usuario (GUI) para visualizar datos en tiempo real y botones para controlar los actuadores del sistema.
- ❖ Implementación de gráficos interactivos para visualizar datos históricos de los sensores.
- ❖ Uso de componentes de Bootstrap para estilizar la aplicación y crear una experiencia de usuario amigable.

- **Dash y Plotly**

Plotly es una biblioteca de gráficos de código abierto que permite la creación de gráficos interactivos.

Usos en el proyecto:

- Visualización de datos históricos de temperatura, humedad, presión, velocidad y dirección del viento.
- Creación de gráficos interactivos para mejorar la comprensión de los datos.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

- **Hilos (Threading) en Python**

Threading es una técnica de programación en la que un programa puede tener múltiples "hilos" de ejecución concurrente.

Usos en el proyecto:

- ❖ Enviar la hora actual al Arduino cada minuto.
- ❖ Solicitar datos al Arduino cada cinco segundos.
- ❖ Asegurar que la interfaz web se actualice en tiempo real sin bloquear la ejecución del programa.

- **HTML y CSS (a través de Dash y Bootstrap)**

HTML (HyperText Markup Language) es el lenguaje estándar para crear páginas web.

CSS (Cascading Style Sheets) es un lenguaje de estilo utilizado para describir la presentación de un documento escrito en HTML.

Usos en el proyecto:

- ❖ Definición de la estructura y el diseño de la interfaz de usuario.
- ❖ Uso de componentes y estilos de Bootstrap para una apariencia y funcionalidad mejoradas.

Haciendo uso conjunto de todas éstas tecnologías podremos construir nuestro sistema IoT de una forma sólida.

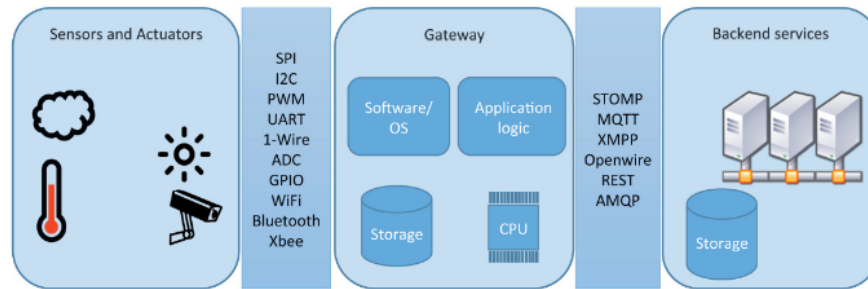


Figura 22: Arquitectura Sistema IoT (Libro Internet of Things Principles and Paradigms)

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015. <i>La autenticidad de este documento puede ser comprobada en la dirección: http://sede.ull.es/validacion</i>	
Identificador del documento: 6750970	Código de verificación: mZvJrzui
Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

6. IMPLEMENTACIÓN

En el siguiente esquema se puede observar el esquema de conexión, así como las entradas y salidas ocupadas en el Arduino Mega y la Raspberry Pi para el correcto funcionamiento del sistema central y los periféricos.

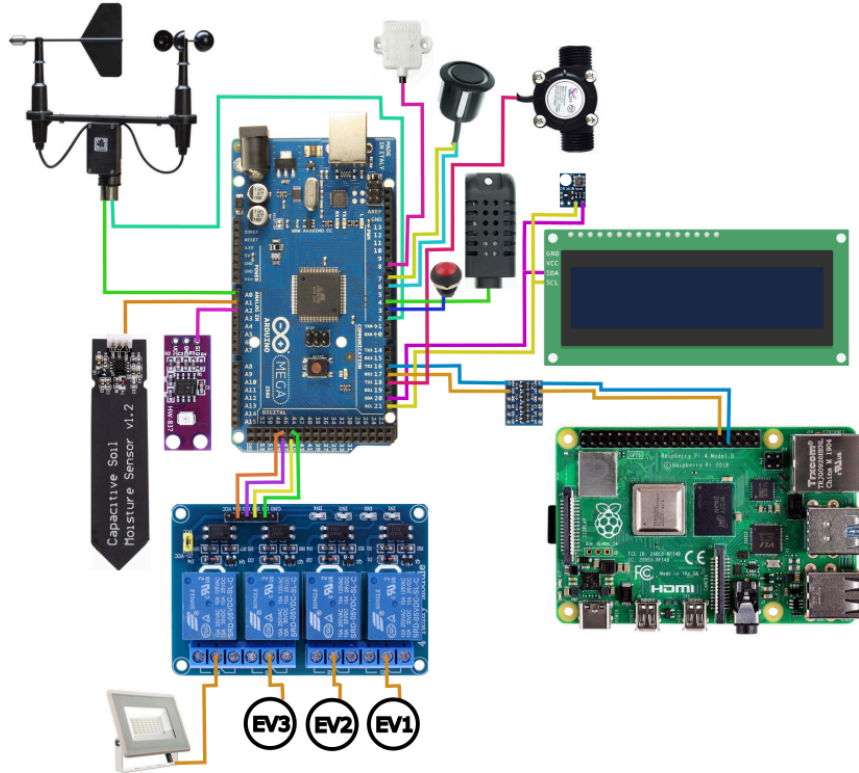


Figura 23: Esquema Conexión Sensores y Actuadores

Los pines de Arduino utilizados para la conexión son los siguientes:

Sensor Humedad Suelo → pin A1	Pulsador Menú → pin D3 INT1
Sensor UV GUVA-S12SD → pin A2	Sensor Hum./Temp. AM2302 → pin D4
Sensor Dirección Viento → pin A0	Pres.Barométrica, LCD → pin I2C (SDA, SCL)
Sensor Velocidad Viento → pin D2 INT0	Elect.Válvula 1 → pin D44
Sensor Presencia XKC-Y23A → pin D8	Elect.Válvula 2 → pin D45
Sensor US → pin D6 & D7 (Echo/Trig)	Elect.Válvula 3 → pin D46
Sensor Caudal XKC-Y23A → pin D5	Luminarias → pin D47
Raspberry Pi → pin D18 y D19 (Tx2, Rx2)	

A continuación se detallarán los pasos seguidos para la implementación del sistema. Tanto la parte de hardware como de software.

6.1 Implementación del Hardware

El sistema central del riego automatizado consta de dos componentes principales: una Raspberry Pi y un Arduino Mega. Estos dispositivos trabajan en conjunto para coordinar la recolección de datos, el procesamiento y la ejecución de las acciones de riego, como el resto de actuadores. Todo el sistema central estará instalado en el interior de una caja que centralizará todas las conexiones de los sensores y de más sistemas periféricos. Además se ha integrado en la parte frontal un display LCD con un pulsador para poder navegar por unos menús básicos donde se pueda realizar la lectura de los sensores de manera rápida.



Figura 22: Caja Sistema Central con display LCD

La caja que alberga el Display LCD y el pulsador ha sido diseñada en 3D haciendo uso del Software Fusion360 e impresa en una impresora 3D Prusa MK3+.

El Arduino Mega se utiliza para conectar todos los sensores y actuadores debido a su capacidad para gestionar múltiples entradas y salidas. Se encarga de la recolección de datos en tiempo real de diversos sensores (humedad del sustrato, radiación UV, velocidad y dirección del viento, temperatura y humedad ambiente, y presión atmosférica) y la activación de dispositivos como las electroválvulas y la bomba de riego.

Por otro lado, la Raspberry Pi, con su mayor capacidad de procesamiento y funcionalidades avanzadas, recibe y procesa la información del Arduino a través de comunicación Serial. La Raspberry Pi es responsable del almacenamiento de datos

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

en la base de datos, el análisis de los mismos, la ejecución de algoritmos de control del riego y la gestión del servidor web para el control remoto.

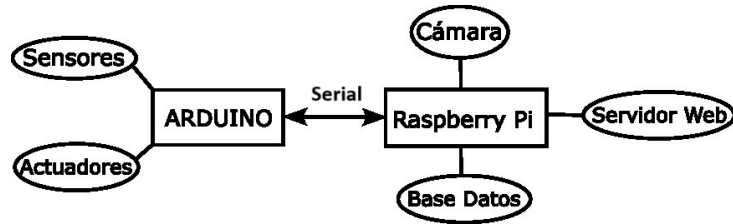


Figura 23: Esquema Sistema Central

La comunicación entre el Arduino y la Raspberry Pi mediante comunicación Serial es lo suficientemente rápida, permitiendo la transferencia de datos en tiempo real y minimizando la carga de procesamiento en la Raspberry Pi. El hecho de que el Arduino se encargue de la tarea de realizar las lecturas de los sensores y de activar y desactivar los actuadores, libera a la Raspberry Pi para realizar tareas más complejas y mantener activa la interfaz de usuario a través de la aplicación web.

Este enfoque modular ofrece flexibilidad y escalabilidad. El Arduino puede gestionar una variedad de sensores y actuadores, permitiendo la fácil adición de nuevos dispositivos sin reconfigurar todo el sistema. La centralización del procesamiento en Raspberry Pi facilita la actualización del software y la implementación de nuevas funcionalidades en el servidor web y la base de datos sin afectar la configuración de los sensores y actuadores.

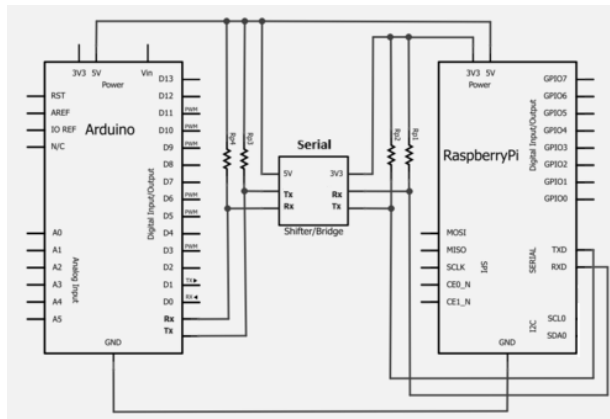


Figura 24: Comunicación Serial entre Arduino / Raspberry Pi

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

Es importante resaltar que, debido a que el Arduino opera a un voltaje lógico de 5V y la Raspberry Pi a 3.3V, es necesario utilizar un "Level Shifter" para adaptar los niveles de voltaje entre ambas placas.

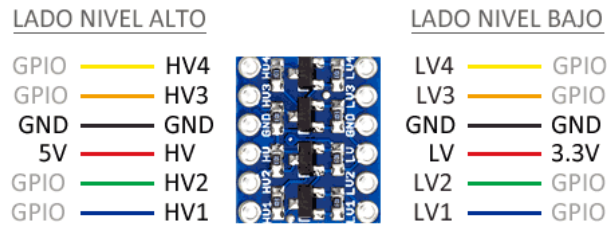


Figura 25: Level Shifter

En el interior de la caja del sistema central estarán instaladas las dos placas de control y procesamiento (Arduino y Raspberry Pi) así como las fuentes de alimentación que alimentarán los diferentes del circuito, como el módulo de relés que servirá para activar/desactivar las electroválvulas de riego y el sistema de alumbrado.

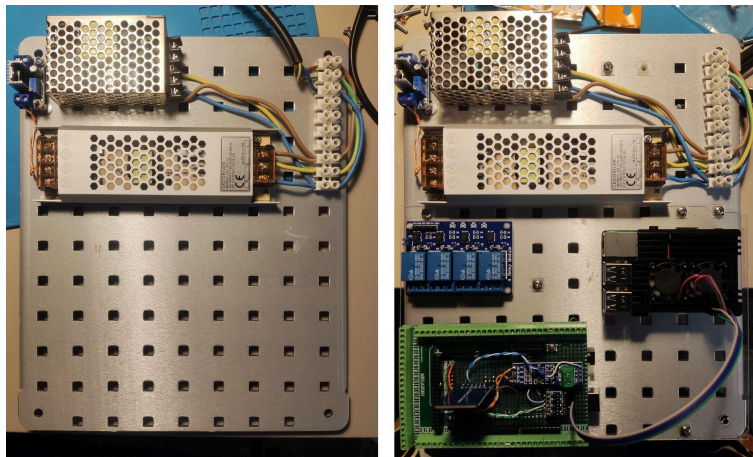


Figura 26: Interior caja sistema central

El sistema cuenta con dos fuentes de alimentación tipo switching de 12VDC para alimentar los principales circuitos y dispositivos.

Una de ellas de 5 Amperios se utiliza para alimentar el Arduino por la entrada Vin y a su vez un regulador de 5V 2A conectado a esta fuente de alimentación, alimentará los sensores que funcionan con estos niveles de voltaje.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015. La autenticidad de este documento puede ser comprobada en la dirección: http://sede.ull.es/validacion	
Identificador del documento: 6750970	Código de verificación: mZvJrzui
Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

La otra fuente de alimentación será de 3 Amperios y se utilizará exclusivamente para la excitación de las bobinas de las electroválvulas del sistema de riego.

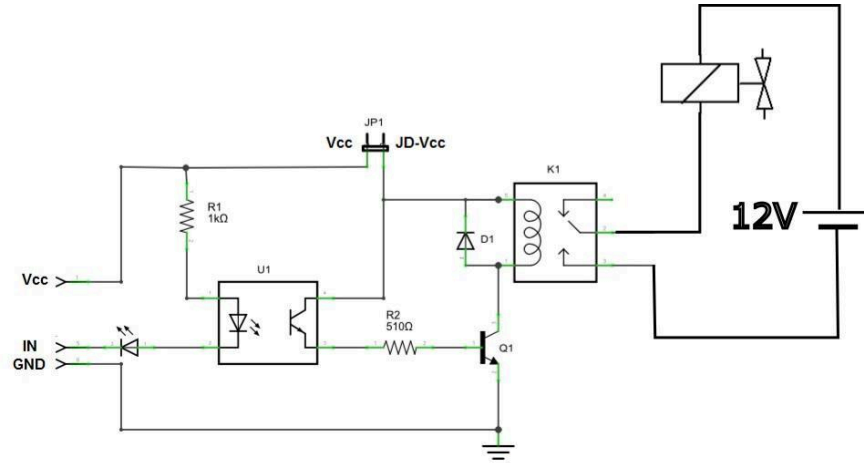


Figura 27: Circuito Activación Electroválvula

Dado que esta caja estará a la intemperie, se ha aprovechado para integrar los sensores de temperatura/humedad (AM2302) y presión barométrica (BMP180) por el exterior de la propia caja.

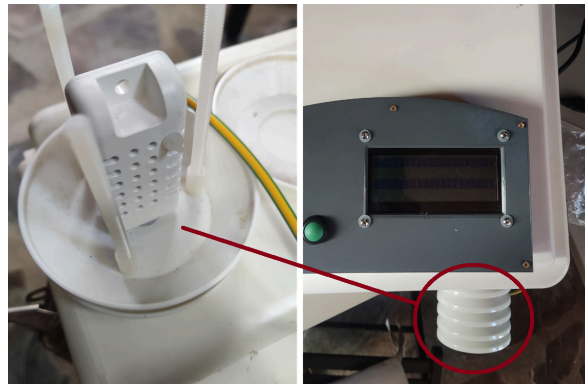


Figura 28: Sensor Temperatura y Humedad

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015. La autenticidad de este documento puede ser comprobada en la dirección: http://sede.ull.es/validacion	
Identificador del documento: 6750970	Código de verificación: mZvJrzui
Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

El sensor de Temperatura/Humedad utiliza un sensor capacitivo de humedad y un termistor para medir el ambiente. La información recopilada se convierte en una señal digital que es enviada al microcontrolador a través de un único pin de datos (pin D2 de Arduino).

El sensor de presión barométrica en cambio, utiliza un piezoeléctrico que detecta cambios en la presión del aire. Estos datos se transmiten digitalmente al microcontrolador a través del protocolo de comunicación I2C.



Figura 29: Piezoeléctrico Sensor Barométrico

El resto de sensores y actuadores irán instalados en el exterior y serán cableados hasta el sistema central para su conexión. A continuación se detalla la forma en la que cada uno de los sensores han sido conectados.

6.1.1 Circuito adaptador Velocidad del Viento

El sensor de velocidad del viento que utilizaremos cuenta con un imán giratorio de dos polos, el cual induce en una bobina transductora una señal AC cuando el anemómetro gira debido a la intensidad del viento. Por ello, la frecuencia de esa señal AC generada es directamente proporcional a la velocidad del viento.

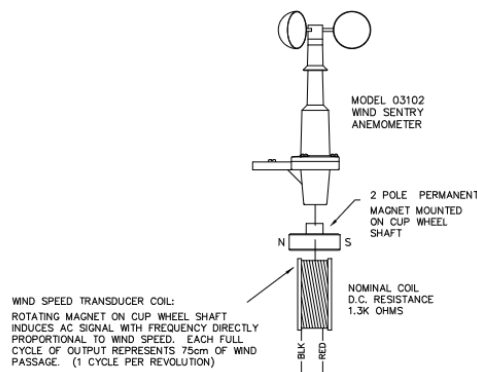


Figura 30: Anemómetro Young 03002

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015. La autenticidad de este documento puede ser comprobada en la dirección: http://sede.ull.es/validacion	
Identificador del documento: 6750970	Código de verificación: mZvJrzui
Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

Después de realizar mediciones con un osciloscopio en la bobina que genera la señal AC, podemos observar que se crea una señal sinusoidal de un nivel de voltaje relativamente bajo que deberá de ser acondicionado.

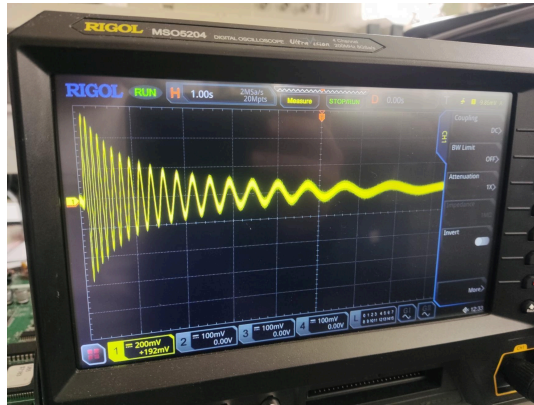


Figura 31: Señal AC bobina anemómetro

Para poder conectar la señal del anemómetro al Arduino, previamente deberemos de convertir la señal AC en una señal pulsada en la que aparecerá un pulso por cada periodo de la frecuencia de la señal generada. Para ello realizaremos una simulación del siguiente circuito (como la que se puede observar en la Figura 32).

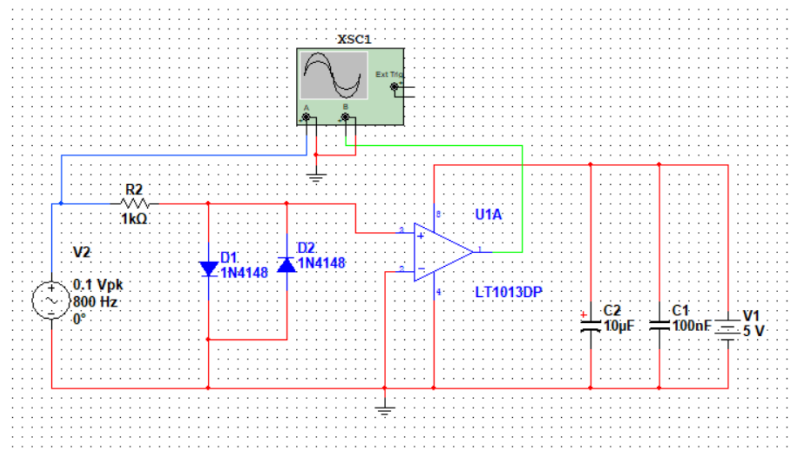


Figura 32: Circuito adaptador de señal

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez
UNIVERSIDAD DE LA LAGUNA

Fecha: 20/08/2024 23:04:24

Santiago Torres Álvarez
UNIVERSIDAD DE LA LAGUNA

20/08/2024 23:05:12

Juan Albino Méndez Pérez
UNIVERSIDAD DE LA LAGUNA

26/08/2024 12:55:49

De este modo, obtenemos una señal pulsada con un periodo equivalente a la frecuencia de la señal sinusoidal inducida.

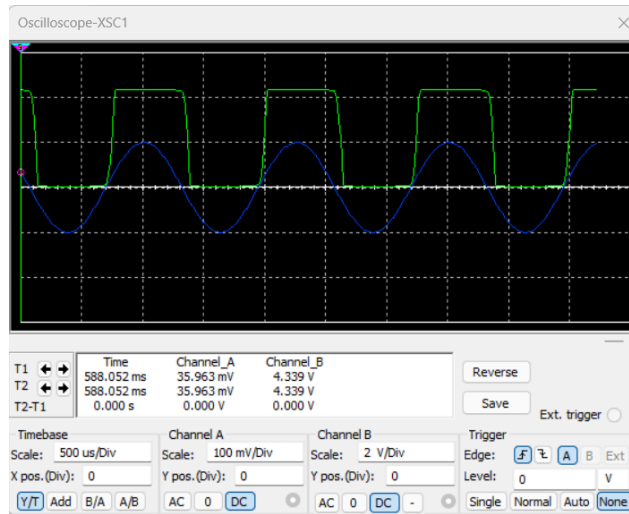


Figura 33: Señal AC vs Señal Pulsada

Ahora podríamos introducir la señal pulsada en una de las entradas digitales (con interrupción) del Arduino y realizar lecturas de tiempos entre pulsos para determinar la velocidad del viento según la relación facilitada por el fabricante.

CALIBRATION FORMULAS

Model 03102 Wind Sentry Anemometer

WIND SPEED vs CUP WHEEL RPM

m/s	=	(0.01250 x rpm) + 0.2
knots	=	(0.02427 x rpm) + 0.4
mph	=	(0.02795 x rpm) + 0.4
km/hr	=	(0.04499 x rpm) + 0.7

WIND SPEED vs OUTPUT FREQUENCY - Hz

m/s	=	(0.7500 x Hz) + 0.2
knots	=	(1.4562 x Hz) + 0.4
mph	=	(1.6770 x Hz) + 0.4
km/hr	=	(2.6994 x Hz) + 0.7

Figura 34: Relación Frecuencia Señal AC / velocidad Viento

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez
UNIVERSIDAD DE LA LAGUNA

Fecha: 20/08/2024 23:04:24

Santiago Torres Álvarez
UNIVERSIDAD DE LA LAGUNA

20/08/2024 23:05:12

Juan Albino Méndez Pérez
UNIVERSIDAD DE LA LAGUNA

26/08/2024 12:55:49

El resto de componentes no necesitan de ninguna adaptación especial para su correcto funcionamiento, ya que vienen con sus circuitos de adaptación de señal integrados en el mismo módulo. Gracias a eso, podrán ser conectados directamente al Arduino tal y como se describe en el diagrama de conexión (Figura 23).

6.1.2 Lecturas de sensores con señales Analógicas.

Utilizaremos las entradas de los conversores ADC de Arduino en los siguientes casos:

- ★ **Sensor de humedad del suelo:** Este sensor trae integrado un circuito preamplificador que adaptará la pequeña señal analógica generada por el sensor en una de mayor estabilidad y menos susceptible al ruido. Conectaremos la salida analógica que sale del previo directamente a una de las entradas analógicas de Arduino.



Figura 35: Preamplificador sensor humedad sustrato

- ★ **Sensor de sentido de dirección del viento:** Este sensor tiene tan solo un potenciómetro que gira libre sobre el eje de la veleta. De este modo, al contar con una resistencia variable, se producirá una caída de tensión en la señal de salida del sensor en función de la posición en la que se encuentre.

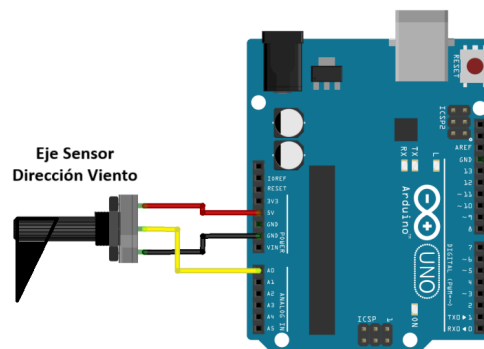


Figura 36: Conexión Sensor Dirección Viento

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

- ★ **Sensor de radiación UV:** Este sensor cuenta con un circuito preamplificador en el propio sensor que acondicionará la pequeña señal producida por el transductor que recibe la señal ultravioleta. De este modo, obtendremos una señal que podrá ser introducida en una de las entradas analógicas de Arduino.



Figura 37: Preamplificador sensor UV

6.1.3 Lecturas de sensores con señales digitales.

Para el conexionado de los sensores restantes utilizaremos las entradas digitales de Arduino.

- ★ **Sensor de caudal:** Este sensor produce pulsos a diferente frecuencia en función de la velocidad del flujo interno del paso del agua. Estos pulsos se introducirán en una de las entradas digitales que disponga de la posibilidad de adjuntar una interrupción de modo que nos permita medir la distancia de los tiempos entre pulsos de manera precisa.

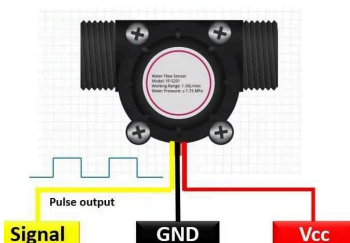


Figura 38: Señal Pulsada Sensor Caudal

- ★ **Sensor presencia de agua:** Este sensor simplemente cambia de estado lógico. Al detectar presencia de agua en la tubería, la salida digital se pone en estado HIGH y al no detectar presencia en LOW.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez
UNIVERSIDAD DE LA LAGUNA

Fecha: 20/08/2024 23:04:24

Santiago Torres Álvarez
UNIVERSIDAD DE LA LAGUNA

20/08/2024 23:05:12

Juan Albino Méndez Pérez
UNIVERSIDAD DE LA LAGUNA

26/08/2024 12:55:49

6.2 Implementación del Software

Comenzaremos programando diferentes *sketch* en el IDE de Arduino para poder testear y comprobar el funcionamiento de cada uno de los sensores que tendremos instalado en nuestro sistema. Más adelante se unificarán estos códigos para poder realizar las lecturas de todos los sensores al mismo tiempo.



Figura 39: Prueba lectura de sensores

- Sensor de Presión barométrica:

El BMP180 mide la presión barométrica utilizando un sensor piezoeléctrico que detecta cambios en la presión del aire. Estos datos se transmiten digitalmente (a través de I2C) al microcontrolador. Haciendo uso de la librería **Adafruit BMP085** tan solo deberemos de definir el sensor y ejecutar la lectura de la presión en Pascal, la cual se podrá expresar en milibares dividiendo el valor obtenido por 100:

```
Adafruit_BMP085 bmp;  
  
Pressure = bmp.readPressure();
```

- **Sensor de Temperatura & Humedad AM2302:**

Al realizar las lecturas de los sensores, el microcontrolador interno del dispositivo recopila estos datos y los envía a través de un pin digital de datos al Arduino, como se puede observar en la Figura 34.

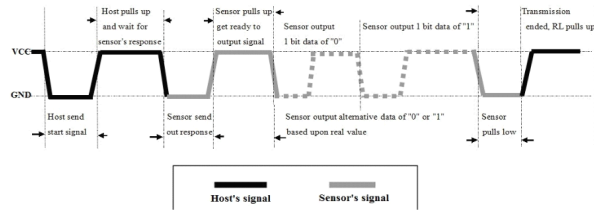


Figura 40: Señal de datos Sensor AM2302

Para programar la lectura de estos datos, haremos uso de la librería **DHT**, mediante la cual con unas pocas líneas de código podremos obtener las lecturas de interés.

```
#include "DHT.h"
#define DHTPIN 2 // Pin digital conectado al sensor
#define DHTTYPE DHT22 // AM2302B es equivalente a DHT22
DHT dht(DHTPIN, DHTTYPE);
// Leer la humedad
float h = dht.readHumidity();
// Leer la temperatura en Celsius
float t = dht.readTemperature();
```

Gracias a la lectura de temperatura y humedad, podremos obtener la sensación térmica.

```
// Calcular el índice de calor en Celsius
float hic = dht.computeHeatIndex(t, h, false);
```

- **Sensor UV GUA-S12SD:**

El sensor GUA-S12SD utiliza un fotodiodo para detectar la radiación UV. La intensidad de la luz UV es convertida en una señal de voltaje analógica que puede ser leída por un microcontrolador. Introduciendo la señal del sensor por una de las entradas Analógicas de Arduino realizaremos la lectura en mV.

```
sensorValue = analogRead(A2);
```

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015. La autenticidad de este documento puede ser comprobada en la dirección: http://sede.ull.es/validacion	
Identificador del documento: 6750970	Código de verificación: mZvJrzui
Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

En función del valor obtenido, gracias a la tabla del fabricante (Figura 35), podemos calcular el índice UV. El código para realizar esta lectura es el que se muestra a continuación.

UV Index	0	1	2	3	4	5
Vout(mV)	<50	227	318	408	503	606
Analog Value	<10	46	65	83	103	124
UV Index	6	7	8	9	10	11 ⁺
Vout(mV)	696	795	881	976	1079	1170+
Analog Value	142	162	180	200	221	240

Figura 41: Tabla de Índice UV para sensor GUVA S12SD

```
vout = vout * 4980.0 / 1024; //Ajuste a valor analógico
if (vout < 50)
{
    uv = 0;
}
else if (vout < 227)
{
    uv = 1;
}
.
.
.
else if (vout < 1079)
{
    uv = 10;
}
else
{
    uv = 11;
}
```

- **Sensor Velocidad del viento Young 03002:**

Este sensor, después de pasar por el circuito adaptador de señal AC a pulsada (véase figura 30), entra a uno de los pines digitales del arduino.

Para poder conocer la frecuencia de estos pulsos, deberemos calcular el tiempo que transcurre entre pulsos. Para ello asociaremos al pin digital una función de interrupción para que el contador aumente cada vez que entre un pulso ascendente por la entrada digital.

```
const int sensorPin = 2; // Pin con interrupción INT0
attachInterrupt(digitalPinToInterrupt(sensorPin), countPulse, RISING);
volatile unsigned long pulseCount = 0;
unsigned long startTime;
unsigned long currentTime = millis();
unsigned long elapsedTime = currentTime - startTime;

void countPulse() {
  pulseCount++;
}
```

Gracias al tiempo obtenido entre pulsos calcularemos la frecuencia de la señal pulsada, y con la fórmula proporcionada por el fabricante obtendremos la velocidad del viento en km/h.

```
// Calcula la frecuencia en Hz
float frequency = pulseCount / (elapsedTime / 1000.0);
// Calcula la velocidad del viento en Km/h
float windSpeed = (2.6994 * frequency) + 0.7;
```

- **Sensor Dirección del viento Young 03002:**

Éste sensor actúa como un potenciómetro. Estará alimentado a 5VDC y según la posición que se encuentre (en los 360°) nos dará un voltaje continuo en la salida que irá desde los 0 hasta los 5V. La salida del sensor la conectaremos a una de las entradas analógicas de Arduino y leeremos el voltaje de entrada para determinar la orientación en la que se encuentra la veleta.

```
const int directionPin = A0; // Pin analógico donde está conectado el anemómetro
int directionValue = analogRead(directionPin); // Leer el valor del anemómetro
float direction = map(directionValue, 0, 1023, 0, 360); // Mapear el valor grados
```

- **Sensor presencia líquido XKC-Y23A:**

El XKC-Y23A emplea un principio capacitivo para detectar la presencia de líquido. Cuando se instala en el exterior de un contenedor no metálico, el sensor mide los cambios en la capacitancia causados por la presencia o ausencia de líquido en el interior. La salida digital del sensor cambia de estado cuando se detecta la presencia de líquido, por ello, conectaremos la salida del sensor a una entrada digital del Arduino. Y para determinar si hay o no presencia de líquido en el conducto únicamente tendremos que realizar una lectura del pin digital asociado.

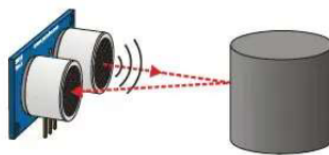
```
if (digitalRead(detection_Pin)) {  
    Serial.println("Líquido Detectado");  
}  
else {  
    Serial.println("Líquido NO Detectado");  
}
```

- **Sensor de Ultrasonidos JSN-SR04T**

El JSN-SR04T utiliza el principio de eco ultrasónico para medir distancias, mediante el cual se emite un pulso de sonido ultrasónico y se procede a medir el tiempo que tarda el eco en regresar después de reflejarse en un objeto.

Éste sensor está compuesto por un emisor y un receptor. Haciendo uso de un cristal de cuarzo, el emisor produce una señal ultrasónica que al transmitirse por el aire y rebotar contra un objeto será capturada de nuevo por el receptor. Gracias a este retardo producido entre la salida del emisor y la entrada del receptor convertiremos el tiempo en distancia de la siguiente manera.

```
// Configura los pines del sensor  
pinMode(trigPin, OUTPUT);  
pinMode(echoPin, INPUT);
```



$$\text{Tiempo} = 2 * (\text{Distancia} / \text{Velocidad})$$
$$\text{Distancia} = \text{Tiempo} \cdot \text{Velocidad} / 2$$

Figura 38: Principio funcionamiento sensor US

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

Conociendo que la velocidad del sonido por el aire con una presión atmosférica a nivel del mar, temperatura de 25°C, y 50% de humedad es de 343 m/s, sabemos que la señal tarda 29,2 microsegundos en recorrer 1 centímetro. Y como la señal tiene que ir, rebotar en el objeto y volver, la fórmula queda de la siguiente manera.

$$343 \frac{m}{s} \cdot 100 \frac{cm}{m} \cdot \frac{1}{1000000} \frac{s}{\mu s} = \frac{1}{29.2} \frac{cm}{\mu s}$$

Para obtener el tiempo que tarda en rebotar la señal, haremos uso de la función `pulseIn(EchoPin,HIGH)`, que contará el tiempo hasta que el pin que tenemos configurado como *echo* se eleve a nivel lógico alto.

$$Distancia(cm) = \frac{Tiempo(\mu s)}{29.2 \cdot 2}$$

```
// Envío de un pulso de 10 microsegundos al pin de disparo
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);

// Lee el tiempo que tarda en recibir el eco
duration = pulseIn(echoPin, HIGH);

// Calcula la distancia en centímetros
distance = duration * 0.034 / 2;
```

Estando el sensor colocado en la parte alta del depósito (orientado hacia abajo) conoceremos la distancia a la que está rebotando con la superficie del agua pudiendo así determinar el porcentaje de llenado del depósito una vez ajustados los márgenes máximos y mínimos.

- Sensor capacitivo Humedad del Sustrato:

El sensor de humedad del suelo capacitivo funciona mediante la detección de cambios en la capacitancia del suelo. A medida que el contenido de agua en el suelo cambia, la capacitancia también cambia. El sensor mide esta variación y genera una señal de salida analógica que es proporcional al nivel de humedad del suelo. Esta señal puede ser leída por un microcontrolador para determinar la humedad del suelo.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015. La autenticidad de este documento puede ser comprobada en la dirección: http://sede.ull.es/validacion	
Identificador del documento: 6750970	Código de verificación: mZvJrzui
Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

Sabiendo los valores máximos y mínimos de la salida del sensor, podemos estimar el porcentaje de humedad del suelo, si asumimos que esta es lineal con la medición del sensor.

Aproximadamente los valores obtenidos son 2.3V en el aire, y 1.3V en agua. Empleando un ADC de 10 bits del Arduino (0-1023) se traduce, respectivamente, en 550 y 250. Gracias a estos valores podremos obtener el valor de la humedad del sustrato en porcentaje.

```
const int sensorPin = A0;
const int humedadAire = 550;
const int humedadAgua = 250;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  int humedad = analogRead(sensorPin);
  Serial.println(humedad);

  porcentajeHumedad = map(humedad, humedadAire, humedadAgua, 0, 100);
  if(porcentajeHumedad > 100) porcentajeHumedad = 100;
```

- Conjunto de lecturas y display LCD

Una vez hayamos testeado todos los sensores que vayamos a utilizar, unificamos las lecturas en un mismo código e implementamos una visualización en el display LCD. Para ello se conecta un pulsador a uno de los pines digitales que podamos asociar una interrupción y lo utilizaremos para navegar por los diferentes menús en el display.

```
pinMode(buttonPin, INPUT_PULLUP); // Configuración del pin del
pulsador
attachInterrupt(digitalPinToInterrupt(buttonPin), buttonISR,
FALLING); // Configuración de la interrupción para el botón

void buttonISR() {
  buttonPressed = true;
}

if (buttonPressed) {
  buttonPressed = false;
  menu = (menu + 1) % 3; // Cambiar entre los tres menús (0, 1, 2)
  Serial.print("Menu changed to: ");
  Serial.println(menu); // Mostrar el cambio de menú en el monitor
serial
  displayMenu();
```

53

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

```
}
```

De éste modo navegaremos por 3 distintos menús y, en función del menú en el que nos encontremos, el LCD imprimirá una información u otra.

```
void displayMenu() {  
  lcd.clear();  
  switch (menu) {  
    case 0:  
    lcd.setCursor(0, 0);  
    lcd.print("ACTUADORES: ");  
    . . .  
    break;  
    case 1:  
    lcd.setCursor(0, 0);  
    lcd.print("ESTACIÓN: ");  
    . . .  
    break;  
    case 2:  
    lcd.setCursor(0, 0);  
    lcd.print("RIEGO: ");  
    . . .  
    break;  
  }  
}
```

Para imprimir la información por los diferentes segmentos del display utilizaremos las siguientes librerías.

```
#include <Wire.h>  
#include <LiquidCrystal_I2C.h>
```

6.2.1 Comunicación

Una vez tenemos las lecturas de los sensores y se visualiza la información en el display, es el momento de realizar la comunicación con la Raspberry Pi para la transferencia de datos. En este caso el Arduino se conectará con la Raspberry Pi a través del puerto serial 2.

En primer lugar, estableceremos la comunicación Serial con la Raspberry a una velocidad de 9600 baudios con la siguiente función.

```
void initializeCommunication() {  
  Serial2.begin(9600);  
}
```

- Una vez establecida la comunicación, el Arduino quedará a la espera de recibir un comando por parte de la Raspberry solicitando los datos de los sensores.

```
void handleCommunication() {
  if (Serial2.available()) {
    String command = Serial2.readStringUntil('\n');
    if (command == "REQUEST_DATA") {
      sendDataToPi();
    }
  }
}
```

Si el comando recibido es “REQUEST_DATA”, entonces el arduino llamará a la función `sendDataToPi()`;

```
void sendDataToPi() {
  Serial2.print("Temperature: ");
  Serial2.print(temperatura);
  Serial2.print("°C, Humidity: ");
  Serial2.print(humedad);
  Serial2.print("%, Pressure: ");
  Serial2.print(presionatm);
  Serial2.print(" mb, Index Temp: ");
  Serial2.print(indicetemperatura);
  Serial2.print("°C, Wind Speed: ");
  Serial2.print(velviento);
  Serial2.print(" km/h, Wind Direction: ");
  Serial2.print(dirviento);
  Serial2.print(" deg, UV Index: ");
  Serial2.print(radiacion);
  Serial2.print(", Water Present: ");
  Serial2.print(waterpresent ? "Yes" : "No");
  Serial2.print(", Riego: ");
  Serial2.print(digitalRead(riegoPin) == LOW ? "ON" : "OFF");
  Serial2.print(", Purgado: ");
  Serial2.print(digitalRead(purgadoPin) == LOW ? "ON" : "OFF");
  Serial2.print(", Llenado: ");
  Serial2.print(digitalRead(llenadoPin) == LOW ? "ON" : "OFF");
  Serial2.print(", Foco: ");
  Serial2.print(digitalRead(focoPin) == LOW ? "ON" : "OFF");
  Serial2.print(", Tank Level: ");
  Serial2.print(niveldeposito);
  Serial2.print(", Caudal: ");
  Serial2.print(caudal);
  Serial2.print(", Humedad del Sustrato: ");
  Serial2.print(humsustrato);
  Serial2.print(", Hora: ");
  Serial2.println(hora);
}
```

- En esta función `sendDataToPi()`; el Arduino envía la información de las lecturas de todos los sensores de manera ordenada, para que la Raspberry sea capaz de recibirla.

- Si los comandos recibidos por la Raspberry Pi en cambio fueran para conmutar el estado de los diferentes actuadores, el arduino actuaría de la siguiente manera.

```

else if (command.startsWith("SET_TIME:")) {

    } else if (command.startsWith("SET_RIEGO:")) {
        EVriego = command.substring(10) == "ON";
        digitalWrite(riegoPin, EVriego ? LOW : HIGH);
        Serial.print("Riego set to: ");
        Serial.println(EVriego ? "ON" : "OFF");
    } else if (command.startsWith("SET_PURGADO:")) {
        EVpurgado = command.substring(12) == "ON";
        digitalWrite(purgadoPin, EVpurgado ? LOW : HIGH);
        Serial.print("Purgado set to: ");
        Serial.println(EVpurgado ? "ON" : "OFF");
    } else if (command.startsWith("SET_LLENADO:")) {
        EVllenado = command.substring(11) == "ON";
        digitalWrite(llenadoPin, EVllenado ? LOW : HIGH);
        Serial.print("Llenado set to: ");
        Serial.println(EVllenado ? "ON" : "OFF");
    } else if (command.startsWith("SET_FOCO:")) {
        foco = command.substring(9) == "ON";
        digitalWrite(focoPin, foco ? LOW : HIGH);
        Serial.print("Foco set to: ");
        Serial.println(foco ? "ON" : "OFF");
    }
}

```

Como se puede observar en la sección de código anterior, el Arduino recibirá el comando de la Raspberry y conmutará el actuador correspondiente al comando recibido.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
 La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

6.2.2 Integración del Servidor con Dash y Bootstrap

A continuación, se procederá a explicar cómo se ha integrado el servidor local con Dash y Bootstrap, y cómo se ha creado la interfaz con los diferentes componentes y ventanas con los datos, las gráficas y los botones del sistema.

Importación de las Bibliotecas Necesarias:

- Se importan las bibliotecas necesarias para crear la aplicación web utilizando Dash y Bootstrap, y para la comunicación con el Arduino a través de la Serial.
- También importamos la librería threading para poder hacer uso de múltiples hilos en la aplicación.

```
import dash
from dash import dcc, html
import dash_bootstrap_components as dbc
import threading
import logging
import time
import re
import serial
```

Variables Globales:

- Se definen variables globales para almacenar los datos recibidos del Arduino y para los datos históricos utilizados en las gráficas.

```
arduino_data = {
    "temperature": "0",
    "humidity": "0",
    "pressure": "0",
    "index_temp": "0",
    "wind_speed": "0",
    "wind_direction": "0",
    "uv_index": "0",
    "water_present": "No",
    "riego": "OFF",
    "purgado": "OFF",
    "llenado": "OFF",
    "foco": "OFF",
    "tank_level": "0",
    "caudal": "0",
    "humsustrato": "0",
    "hora": "00:00"
}
```

57

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

```

historical_data = {
    "temperature": [],
    "humidity": [],
    "pressure": [],
    "wind_speed": [],
    "wind_direction": [],
    "timestamps": []
}

```

Función para Parsear los Datos Recibidos del Arduino:

- Esta función convierte los datos recibidos en un formato más manejable y actualiza las variables globales.

```

def parse_data(data):
    global arduino_data, historical_data
    try:
        parts = data.split(',')
        for part in parts:
            if ':' in part:
                key, value = part.split(':', 1)
                key = key.strip().lower().replace(" ", "_")
                arduino_data[key] = value.strip()

        timestamp = time.strftime("%H:%M:%S")
        historical_data["timestamps"].append(timestamp)

        try:
            historical_data["temperature"].append(float(re.sub(r'^\d.+',
            ', ', arduino_data["temperature"])))
        except ValueError:
            historical_data["temperature"].append(0)

        try:
            historical_data["humidity"].append(float(re.sub(r'^\d.+', ' ',
            arduino_data["humidity"])))
        except ValueError:
            historical_data["humidity"].append(0)

        try:
            historical_data["pressure"].append(float(re.sub(r'^\d.+', ' ',
            arduino_data["pressure"])))
        except ValueError:
            historical_data["pressure"].append(0)

        try:
            historical_data["wind_speed"].append(float(re.sub(r'^\d.+',
            ', ', arduino_data["wind_speed"])))
        except ValueError:
            historical_data["wind_speed"].append(0)

        try:
            historical_data["wind_direction"].append(float(re.sub(r'^\d.+', ' ',
            arduino_data["wind_direction"])))
        except ValueError:
            historical_data["wind_direction"].append(0)

```

58

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
 La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

```

if len(historical_data["timestamps"]) > 100:
    for key in historical_data:
        historical_data[key].pop(0)
except Exception as e:
    logging.error(f"Error parsing data: {e}")

```

Creación de las Pestañas y Componentes

Definición del Layout de la Aplicación:

- Se utiliza `dcc.Tabs` para crear las diferentes pestañas de la aplicación. Cada pestaña contiene un `dbc.Container` con las filas y columnas necesarias para mostrar los datos y los controles.

```

layout = html.Div([
    dcc.Tabs([
        dcc.Tab(label='Status', children=[
            dcc.Interval(id='interval-component', interval=1*1000,
n_intervals=0),
            dbc.Container([
                dbc.Row([dbc.Col(html.H1("Estado Huerto IoT",
className="text-center mb-4"))]),
                dbc.Row([
                    dbc.Col(html.Div([html.H4("Temperatura:"),
html.P(id='temperature')])),
                    dbc.Col(html.Div([html.H4("Humedad:"),
html.P(id='humidity')])),
                    dbc.Col(html.Div([html.H4("Presión:"),
html.P(id='pressure')])),
                    dbc.Col(html.Div([html.H4("Índice de calor:"),
html.P(id='index_temp')]))
                ]),
                dbc.Row([
                    dbc.Col(html.Div([html.H4("Velocidad del
viento:"), html.P(id='wind_speed')])),
                    dbc.Col(html.Div([html.H4("Dirección del
viento:"), html.P(id='wind_direction')])),
                    dbc.Col(html.Div([html.H4("Índice UV:"),
html.P(id='uv_index')])),
                    dbc.Col(html.Div([html.H4("Agua presente:"),
html.P(id='water_present')]))
                ]),
                dbc.Row([
                    dbc.Col(html.Div([html.H4("Riego:"),
html.P(id='riego')])),
                    dbc.Col(html.Div([html.H4("Purgado:"),
html.P(id='purgado')])),
                    dbc.Col(html.Div([html.H4("Llenado:"),
html.P(id='llenado')])),
                    dbc.Col(html.Div([html.H4("Foco:"),
html.P(id='foco')]))
                ]),
                dbc.Row([

```

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrZui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49


```

        dbc.Col(html.Div([html.H4("Nivel del tanque:"),
html.P(id='tank_level')])),
        dbc.Col(html.Div([html.H4("Caudal:"),
html.P(id='caudal')])),
        dbc.Col(html.Div([html.H4("Humedad del
sustrato:"), html.P(id='humsustrato')])),
        dbc.Col(html.Div([html.H4("Hora:"),
html.P(id='hora')]))
    ])
])
]),

```

Callbacks para Actualizar los Datos en Tiempo Real:

- Se definen callbacks para actualizar los datos en tiempo real en la pestaña [Status](#), las gráficas en la pestaña [Monitor](#), y para manejar los botones de los actuadores en la pestaña [Actuadores](#).

```

def register_callbacks(app):

    # Callbacks para la pestaña de Status
    @app.callback(
        [Output('temperature', 'children'),
        Output('humidity', 'children'),
        Output('pressure', 'children'),
        Output('index_temp', 'children'),
        Output('wind_speed', 'children'),
        Output('wind_direction', 'children'),
        Output('uv_index', 'children'),
        Output('water_present', 'children'),
        Output('riego', 'children'),
        Output('purgado', 'children'),
        Output('llenado', 'children'),
        Output('foco', 'children'),
        Output('tank_level', 'children'),
        Output('caudal', 'children'),
        Output('humsustrato', 'children'),
        Output('hora', 'children')],
        [Input('interval-component', 'n_intervals')]
    )
    def update_status(n):
        return (f"{arduino_data['temperature']}",
                f"{arduino_data['humidity']}",
                f"{arduino_data['pressure']}",
                f"{arduino_data['index_temp']}",
                f"{arduino_data['wind_speed']}",
                f"{arduino_data['wind_direction']}°",
                f"UV: {arduino_data['uv_index']}",
                f"Agua: {arduino_data['water_present']}",
                f"Riego: {arduino_data['riego']}",
                f"Purgado: {arduino_data['purgado']}",
                f"Llenado: {arduino_data['llenado']}",
                f"Foco: {arduino_data['foco']}",
                f"Nivel: {arduino_data['tank_level']}",
                f"Caudal: {arduino_data['caudal']}",
                f"H. Sustrato: {arduino_data['humsustrato']}",
                arduino_data["hora"])

```

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
 La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

Creación de las Gráficas

Las gráficas se implementan en la pestaña **Monitor** usando Dash y Plotly.

Definición del Layout de la Pestaña **Monitor**:

Se usa `dcc.Graph` para cada una de las gráficas y `dbc.Container` para organizar las gráficas en filas.

```
dcc.Tab(label='Monitor', children=[
    dcc.Interval(id='interval-graph', interval=1*1000,
n_intervals=0),
    dbc.Container([
        dbc.Row([dbc.Col(dcc.Graph(id='temperature-graph'))]),
            dbc.Row([dbc.Col(dcc.Graph(id='humidity-graph'))]),
            dbc.Row([dbc.Col(dcc.Graph(id='pressure-graph'))]),
        dbc.Row([dbc.Col(dcc.Graph(id='wind_speed-graph'))]),
        dbc.Row([dbc.Col(dcc.Graph(id='wind_direction-graph'))]),
    ])
])
```

Creación de los Pulsadores

Los pulsadores se implementan en la pestaña **Actuadores** usando Dash y Bootstrap.

Definición del Layout de la Pestaña **Actuadores**:

- Se usa `dbc.Button` para crear botones de Bootstrap y `html.P` para mostrar el estado de los actuadores.

```
dcc.Tab(label='Actuadores', children=[
    dbc.Container([
        dbc.Row([dbc.Col(html.H1("Control de Actuadores",
className="text-center mb-4"))]),
        dbc.Row([
            dbc.Col(html.Div([html.H4("Riego:"),
html.P(id='riego_status'), dbc.Button("REGAR", id='REGAR',
color="primary")])),
            dbc.Col(html.Div([html.H4("Purgado:"),
html.P(id='purgado_status'), dbc.Button("PURGAR", id='PURGAR',
color="primary")])),
            dbc.Col(html.Div([html.H4("Llenado:"),
html.P(id='llenado_status'), dbc.Button("LLENAR", id='LLENAR',
color="primary")])),
            dbc.Col(html.Div([html.H4("Foco:"),
html.P(id='foco_status'), dbc.Button("FOCO", id='FOCO', color="primary")]))
        ])
    ])
])
```

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

Callback para los Pulsadores:

- Se define un callback que se ejecuta cuando se presiona uno de los botones. Este callback envía un comando al Arduino para cambiar el estado del actuador correspondiente y actualiza el estado mostrado en la interfaz.

```
@app.callback(
    [Output('riego_status', 'children'),
     Output('purgado_status', 'children'),
     Output('llenado_status', 'children'),
     Output('foco_status', 'children')],
    [Input('REGAR', 'n_clicks'),
     Input('PURGAR', 'n_clicks'),
     Input('LLENAR', 'n_clicks'),
     Input('FOCO', 'n_clicks')]
)
```

Función `toggle_actuator` para Enviar Comandos al Arduino

Esta función se usa para enviar un comando al Arduino para cambiar el estado de un actuador.

```
def update_actuators(riego_clicks, purgado_clicks,
llenado_clicks, foco_clicks):
    ctx = callback_context

    if not ctx.triggered:
        button_id = 'No clicks yet'
    else:
        button_id = ctx.triggered[0]['prop_id'].split('.')[0]

    if button_id == 'REGAR':
        toggle_actuator('riego')
    elif button_id == 'PURGAR':
        toggle_actuator('purgado')
    elif button_id == 'LLENAR':
        toggle_actuator('llenado')
    elif button_id == 'FOCO':
        toggle_actuator('foco')

    return (f"Riego: {arduino_data['riego']}",
            f"Purgado: {arduino_data['purgado']}",
            f"Llenado: {arduino_data['llenado']}",
            f"Foco: {arduino_data['foco']}")
```

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

7. PRUEBAS DE FUNCIONAMIENTO

Una vez implementado el sistema, se procede a realizar las pruebas de funcionamiento en su ubicación definitiva.



Figura 43: Sistema central instalado en pared

Se verifica que todos los sensores y actuadores se encuentren correctamente conectados al Arduino según el esquema de conexiones previsto. Se revisan las conexiones eléctricas para evitar posibles cortocircuitos y se carga el código en el microcontrolador. Posteriormente, se inicia el servidor Python en Raspberry Pi para establecer la comunicación con el Arduino.

Podemos comprobar que la lectura de los sensores se están realizando correctamente y se muestran por pantalla. El estado de los actuadores es correcto y se visualiza toda la información por el display LCD.

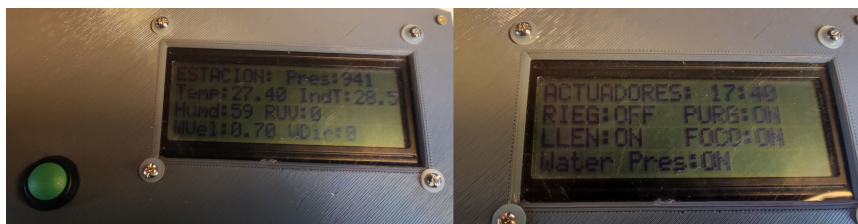


Figura 44: Navegación Menús LCD

63

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

Para confirmar la efectividad de la comunicación, se enviaron comandos desde Raspberry Pi al Arduino. Primero, se envía el comando `REQUEST_DATA` para solicitar los datos de los sensores, y se verifica que el Arduino responde con los datos en el formato esperado. Luego, se actualiza la hora del sistema en la Raspberry Pi y se envía el comando `SET_TIME:HH:MM`, observando que la hora se actualiza correctamente en el display del Arduino. Ambas pruebas confirman que hay una comunicación bidireccional.

Pruebas de Sensores

Sensor de Temperatura y Humedad (DHT22)

Se monitorizaron las lecturas de temperatura y humedad y se comparan con un termómetro calibrado. Las lecturas del sensor DHT22 son consistentes y precisas, reflejando correctamente las condiciones ambientales.

Sensor de Presión (BMP085)

Las lecturas del sensor BMP085 fueron precisas y se ajustaron correctamente al nivel del mar, considerando la altitud de la ubicación de la prueba.

Sensor de Radiación UV

Se realizaron pruebas de exposición del sensor UV a diferentes niveles de luz ultravioleta. Las lecturas del índice UV se monitorizaron desde el display LCD mostrando variaciones en función de la intensidad de la luz UV. Esto confirma el correcto funcionamiento del sensor.

Sensor de Viento

Para probar la velocidad y dirección del viento, se instala sobre una barra provisional a la intemperie en una zona expuesta a todos los sentidos. Las lecturas de velocidad y dirección del viento se actualizan en tiempo real, reflejando correctamente las condiciones climáticas.



Figura 45: Anemómetro Instalado

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

Pruebas de Interfaz Web

Monitorización en Tiempo Real

Se comprobó que los datos de los sensores se actualizan en tiempo real en la pestaña "Status" del servidor web. Las lecturas de temperatura, humedad, presión, velocidad del viento, y dirección del viento se actualizan cada segundo sin retrasos perceptibles.



Figura 46: Pantalla Status del Servidor Web

Gráficas de Datos

Las gráficas en la pestaña "Monitor" muestran datos históricos precisos de las variables monitorizadas. Las gráficas de temperatura, humedad, presión, velocidad del viento y dirección del viento se actualizan correctamente, proporcionando una visualización clara de las tendencias de los datos.

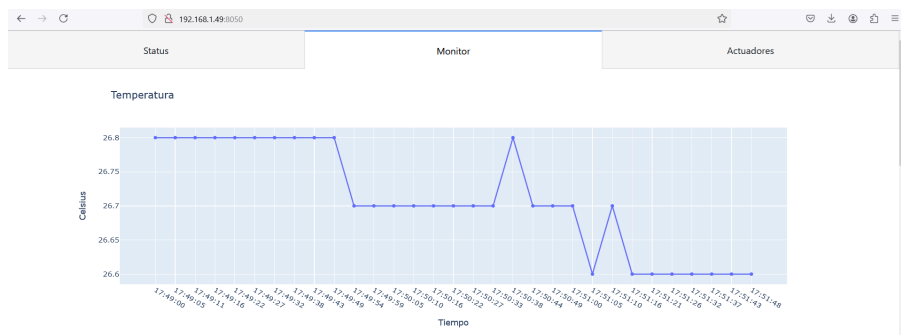


Figura 47: Pantalla Monitor - Gráfico Temperatura

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez
UNIVERSIDAD DE LA LAGUNA

Fecha: 20/08/2024 23:04:24

Santiago Torres Álvarez
UNIVERSIDAD DE LA LAGUNA

20/08/2024 23:05:12

Juan Albino Méndez Pérez
UNIVERSIDAD DE LA LAGUNA

26/08/2024 12:55:49

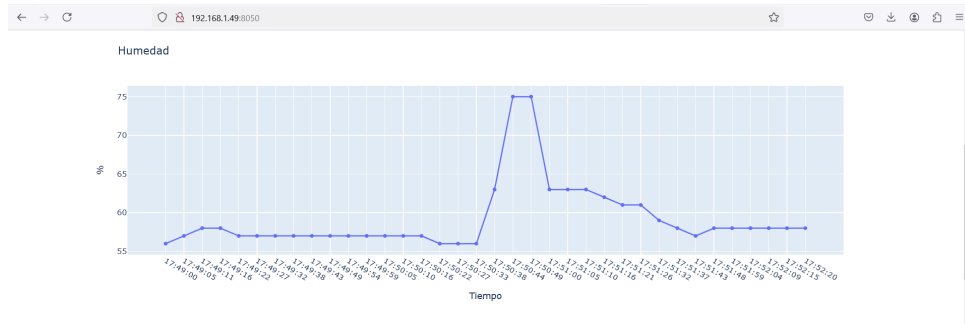


Figura 48: Pantalla Monitor - Gráfico Humedad

Control de Actuadores

Los interruptores en la pestaña "Actuadores" permiten un control preciso de los actuadores. Cada interruptor muestra el estado ON/OFF del actuador correspondiente, y los actuadores responden inmediatamente (escasos segundos) a los cambios, confirmando su correcto funcionamiento.

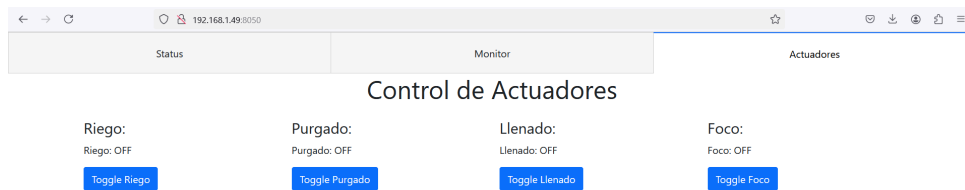


Figura 49: Pantalla Actuadores - EV Riego, purga, llenado y Foco

Al activar cualquiera de estos botones, se produce un cambio en la variable booleana correspondiente, llegando a activar la salida digital del arduino que a su vez provocará la activación del relé de control.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

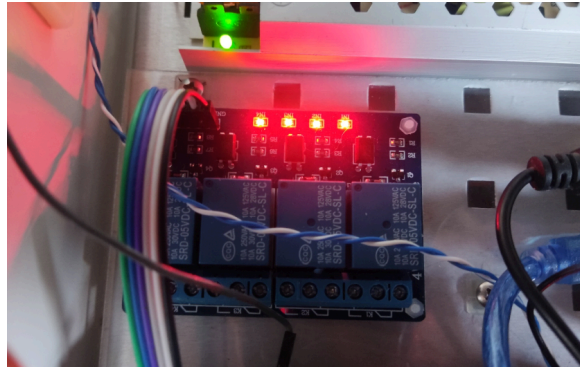


Figura 50: Activación de Relés

Al activar el pulsador Riego, haremos que la electroválvula de riego se abra haciendo que el controlador de la bomba detecte la apertura y active la bomba de riego para mantener una presión adecuada.



Figura 51: Activación EV Riego

En este momento podremos monitorizar las diferentes variables del sistema de riego como el caudal, o el nivel del depósito que aunque el controlador de la bomba tiene un sistema de protección para apagar la bomba en caso de que el depósito se vacíe, deberemos preocuparnos de mantener un buen nivel de agua en las reservas en todo momento.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrZui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49



Figura 52: Manguera de riego por exudación

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

8. Resultados Obtenidos

La integración de múltiples sensores y actuadores en el sistema de monitoreo y control de huerto IoT proporciona una visión completa y detallada de las condiciones ambientales y de suelo, lo cual es crucial para implementar prácticas de riego eficientes y sostenibles. A continuación, se describen los resultados obtenidos y su impacto en la gestión del riego:

Monitorización en Tiempo Real

El sistema recopila y muestra en tiempo real datos de temperatura, humedad, presión atmosférica, radiación UV, velocidad y dirección del viento, humedad del sustrato, nivel del tanque de agua y caudal.

La monitorización en tiempo real de estos parámetros nos permitirá ajustar el riego en función de las condiciones actuales del ambiente y del suelo. Por ejemplo, en días soleados con alta radiación UV y temperaturas elevadas, se puede aumentar la frecuencia del riego para compensar la mayor evaporación. En contraste, en días nublados y húmedos, el riego se puede reducir, evitando el desperdicio de agua y la saturación del suelo.

Control de Actuadores

Los actuadores para riego, purgado, llenado y foco pueden ser controlados de manera remota a través de la interfaz web, lo que permite activar o desactivar estos sistemas según sea necesario y de forma remota.

El control remoto de los actuadores permite un manejo cómodo del sistema. Por ejemplo, si el sensor de humedad del sustrato indica niveles bajos de humedad, podríamos activar el riego desde cualquier lugar en el que nos encontremos, asegurando que las plantas reciban el agua necesaria. Esto no solo optimiza el uso del agua sino que también mejora la salud de las plantas al proporcionarles condiciones óptimas de crecimiento en todo momento evitando el empeoramiento de las raíces producido por el estrés hídrico.

Análisis de Datos Históricos

La interfaz web permite visualizar gráficas de datos históricos de temperatura, humedad, presión, velocidad y dirección del viento.

El análisis de datos históricos nos permitirá identificar patrones y tendencias en las condiciones ambientales. Esta información es valiosa para planificar el riego de manera más eficiente en un futuro. Por ejemplo, si se observa que ciertas horas del día tienen consistentemente alta demanda de agua, se puede ajustar el horario de riego para adaptarse a estas necesidades, minimizando el desperdicio de agua por evaporación y garantizando que las plantas reciban riego en los momentos más críticos.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

Optimización del Uso del Agua

El sistema proporciona información detallada y precisa sobre las condiciones del ambiente y del suelo, permitiéndonos hacer riegos más precisos y dirigidos.

La optimización del uso del agua es crucial en la agricultura sostenible. Al proporcionar agua solo cuando y donde es necesario, se reduce el consumo de agua, se previene la erosión del suelo. Además, un riego preciso mejora la salud de las plantas, lo que puede resultar en un aumento de la productividad y una reducción de los costos operativos.

Según la imagen del gráfico simulado (Figura 53), gracias al sistema implementado, podremos interpretar diferentes patrones de evaporación del sustrato en función de la climatología. En este caso un día de verano en el que según van avanzando las horas del día, la radiación UV, el aumento de las temperaturas y descenso de la humedad ambiente afecta drásticamente en la evaporación de agua del sustrato.

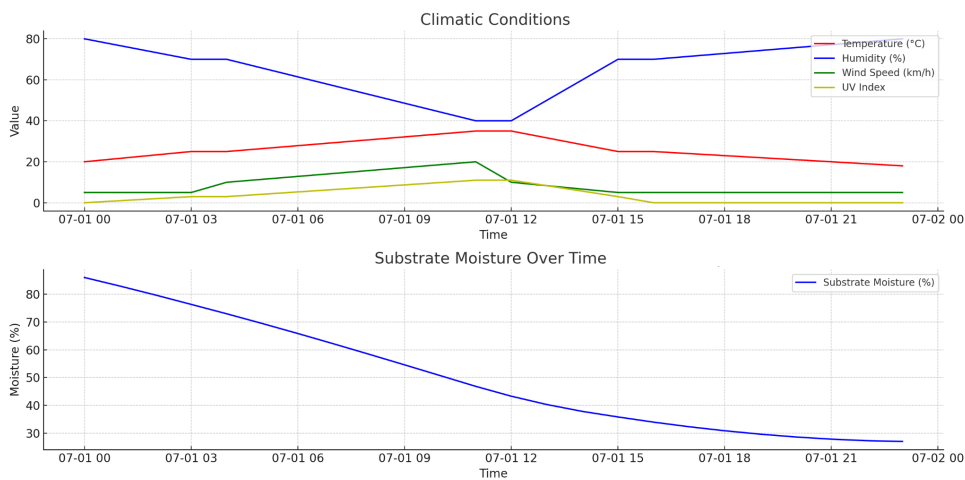


Figura 53: Evaporación según climatología (simulación)

El sistema de monitoreo y control de huerto IoT ha demostrado ser una herramienta eficaz para implementar prácticas de riego eficientes y sostenibles. La capacidad de monitorizar las condiciones ambientales en tiempo real, controlar remotamente los actuadores y analizar datos históricos proporciona a los agricultores la información y el control necesarios para optimizar el uso del agua, mejorar la salud de las plantas y contribuir a una mayor producción.

Estos resultados validan la efectividad del sistema y su potencial para ser implementado en diversas aplicaciones agrícolas.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

9. Conclusiones y Líneas Futuras

Conclusiones

El desarrollo e implementación del sistema de monitoreo y control de huerto IoT ha demostrado ser una solución eficaz para optimizar las labores en la agricultura. A través de la integración de múltiples sensores y actuadores, junto con una interfaz web intuitiva, el sistema proporciona una visión completa y en tiempo real de las condiciones ambientales y del suelo. Los resultados de las pruebas realizadas confirman que el sistema es capaz de:

- Proporcionar datos precisos y en tiempo real sobre temperatura, humedad, presión atmosférica, radiación UV, velocidad y dirección del viento, nivel de agua en el tanque y humedad del sustrato.
- Permitir el control remoto y eficiente de actuadores como el riego, purgado del agua, llenado del depósito y activación de luminarias.
- Gracias a los gráficos se facilita el análisis de datos históricos para identificar patrones y tendencias que ayudan en la planificación y optimización del riego.

Líneas Futuras

Para seguir mejorando y expandiendo las capacidades del sistema, aprovechando lo ya implementado, se proponen las siguientes líneas de trabajo futuras:

1. Integración de una Base de Datos

Almacenar los datos de los sensores durante un período más largo para contar con una gran cantidad de información para el análisis.

Implementar una base de datos que permita el almacenamiento y consulta de datos históricos. Esto facilita el análisis de tendencias a largo plazo y la generación de informes detallados sobre las condiciones climáticas del huerto.

2. Sistema de Predicción Meteorológica

Utilizar datos en tiempo real y datos históricos para realizar predicciones meteorológicas más precisas mediante modelos de inteligencia artificial.

Desarrollar un modelo predictivo que combine los datos de los sensores en tiempo real con los datos históricos almacenados en la base de datos. Este modelo podrá anticipar condiciones meteorológicas futuras y ajustar las estrategias de riego en consecuencia.

3. Sistema Autónomo de Toma de Decisiones

Automatizar las decisiones de riego basadas en la época del año y las predicciones meteorológicas, activando y desactivando las electroválvulas de riego de manera independiente.

Implementar un sistema autónomo que utilice los datos de los sensores y las predicciones meteorológicas para determinar automáticamente los tiempos y frecuencias de riego. Esto incluirá la capacidad de activar y desactivar las electroválvulas de riego de forma independiente para optimizar el uso del agua.

4. Integración de un Sistema de Alarmas

Detectar y notificar problemas como fugas, rebosamiento de agua, sequía en ciertas partes del terreno y obstrucciones en la canalización del agua.

Desarrollar un sistema de alarma que monitoree continuamente el estado del huerto y envíe notificaciones en caso de detectar condiciones anormales, como índices de radiación muy elevados o problemas en el sistema de riego.

6. Adición de Nuevos Actuadores

Ampliar el sistema con nuevos actuadores que permitan un control más detallado y preciso del entorno del huerto.

Incorporar electroválvulas adicionales para cada línea de cultivo, permitiendo el ajuste de tiempos de riego por línea de producción. Además, incluye el manejo de sistemas de ventilación en caso de tratarse de un invernadero cerrado, ajustando automáticamente las condiciones del ambiente según las necesidades específicas de las plantas.

La implementación de estas mejoras contribuirá significativamente a la agricultura sostenible, optimizando el uso de recursos y mejorando la productividad y salud de los cultivos.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

10. Presupuesto

CONCEPTO	PRECIO UNITARIO	UNIDADES	IMPORTE
COSTES MATERIALES¹			656,99 €
COMPONENTES ELECTRÓNICOS			279,13 €
Arduino Mega 2560	14,70 €	1	14,70 €
Shield Bornas Arduino Mega	11,67 €	1	11,67 €
Raspberry Pi 4 (4GB RAM)	55,00 €	1	55,00 €
Tarjeta micro SD (64GB)	8,50 €	1	8,50 €
Fuente de alimentación para Raspberry Pi	15,50 €	1	15,50 €
Fuente de alimentación 12VDC 5A	7,50 €	2	15,00 €
Módulo regulador de temperatura L7805 5VDC	2,38 €	1	2,38 €
Sensor de Temperatura y Humedad (DHT22)	5,50 €	1	5,50 €
Sensor de Presión Barométrica (BMP085)	4,20 €	1	4,20 €
Sensor de Radiación UV (GUVA-S12SD)	3,50 €	1	3,50 €
Sensor de Humedad del Suelo	1,72 €	4	6,88 €
Anemómetro (Sensor de Viento)	57,00 €	1	57,00 €
Módulo Relé de 4 Canales	6,20 €	1	6,20 €
Electroválvulas bobina 12vDC NC	19,00 €	3	57,00 €
Display LCD 4x20 con Módulo I2C	9,60 €	1	9,60 €
Pulsador	1,20 €	1	1,20 €
PCB por puntos y Jumpers	0,50 €	1	0,50 €
Miscelanea (Resistencias, Condensadores, Transistor)	4,80 €	1	4,80 €
Material Instalación Eléctrica			114,16 €
Caja de Sistema Central Aislamiento para intemperie	79,00 €	1	79,00 €
Caja Cuadro Eléctrico con carril DIN	12,00 €	1	12,00 €
Diferencial CHINT 2 polos 40A 2 módulos 30mA clase AC	14,97 €	1	14,97 €
Interruptor Magnetotérmico unipolar/neutro HAGER de 10A	4,59 €	1	4,59 €
Tubo PVC instalación eléctrica 2m	1,20 €	3	3,60 €
Material Sistema Riego			263,70 €
Bomba Riego Sterwings 900w Caudal Max 3,800 l/h	79,00 €	1	79,00 €
Control automático para bomba STERWINS	42,00 €	1	7,80 €
Filtro de agua Sterwings de 80mm	19,00 €	1	19,00 €

¹ Los precios mencionados son aproximados y pueden variar según el proveedor y la ubicación. Se recomienda verificar los precios actualizados antes de realizar la compra.

73

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970

Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez
UNIVERSIDAD DE LA LAGUNA

Fecha: 20/08/2024 23:04:24

Santiago Torres Álvarez
UNIVERSIDAD DE LA LAGUNA

20/08/2024 23:05:12

Juan Albino Méndez Pérez
UNIVERSIDAD DE LA LAGUNA

26/08/2024 12:55:49

Grifería	35,00 €	1	35,00 €
Manguera de riego por exudación 50m	49,00 €	2	98,00 €
Tubería de polietileno ø32mm 1m	0,50 €	15	7,50 €
Accesorios adaptadores tubería (Codos, T, adapt. Rosca)	17,40 €	1	17,40 €
COSTES DE EJECUCIÓN			5.407,00 €
TRABAJO DE INGENIERÍA			4.925,00 €
Trabajo de ingeniería - Diseño, cálculos, investigación, ejecución de soluciones y análisis de resultados	25,00 €	197	4.925,00 €
TRABAJO TÉCNICO			482,00 €
Instalación sistema informático, WIFI, servidor - Técnico informático	18,00 €	9	162,00 €
Instalación sistema de riego y sistema eléctrico - Técnico Especialista	18,00 €	8	144,00 €
Soldadura y montaje de componentes, cableado interno - Téc. Electrónica	16,00 €	11	176,00 €

CONCEPTO			TOTAL
COSTES MATERIALES			656,99 €
COSTES DE EJECUCIÓN			5.407,00 €
	TOTAL COSTES		6.063,99 €
	GASTOS GENERALES (G.G.) ²	13%	788,32 €
	BENEFICIO INDUSTRIAL (B.I.)	6%	363,84 €
	SUMA DE G.G. Y B.I.		1.152,16 €
	COSTE ESTIMADO		7.216,15 €
	I.G.I.C	7%	505,13 €
	COSTE TOTAL DEL PROYECTO		7.721,28 €

El coste total de este proyecto asciende a siete mil setecientos veintiún euros con veintiocho céntimos.

² Los Gastos generales (GG) y el beneficio industrial (BI) están regulados por el Real Decreto 1098/2001, de 12 de octubre, Reglamento general de la Ley de Contratos de las Administraciones Públicas. El porcentaje para gastos generales es del 13% al 17% y del 6% para el beneficio industrial del contratista.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

11. Bibliografía

Libros

1. **"Internet of Things: Principles and Paradigms"** - Edited by Rajkumar Buyya and Amir Vahid Dastjerdi.
 - o Un libro que cubre los principios y paradigmas del Internet de las Cosas (IoT), incluyendo aplicaciones en agricultura.
2. **"Smart Agriculture: An Approach Towards Better Farming"** - Edited by Ajay Rana, Satyajit Anand, and Vishal Jain.
 - o Este libro explora diferentes aspectos de la agricultura inteligente, incluyendo el uso de sensores y tecnología IoT.
3. **"Environmental Monitoring with Arduino: Building Simple Devices to Collect Data About the World Around Us"** - By Emily Gertz and Patrick Di Justo.
 - o Una guía práctica para construir dispositivos de monitoreo ambiental utilizando Arduino.

Artículos Científicos

1. **"A Review of Smart Agriculture IoT: Applications, Technology, and Opportunities"** - Published in IEEE Access.
 - o Un artículo que revisa las aplicaciones y tecnologías IoT en la agricultura inteligente, destacando oportunidades y desafíos.
2. **"IoT-Based Smart Agriculture: Towards Making the Fields Talk"** - Published in IEEE Internet of Things Journal.
 - o Este artículo discute cómo las tecnologías IoT pueden transformar la agricultura al proporcionar datos en tiempo real y tomar decisiones informadas.
3. **"Precision Agriculture and the Future of Farming in Europe"** - Report by the European Parliament.
 - o Un informe que examina el estado actual y el futuro de la agricultura de precisión en Europa, incluyendo tecnologías IoT.
4. **"Weather Prediction Using Machine Learning Techniques"** - Published in IEEE Transactions on Neural Networks and Learning Systems.
 - o Un artículo que explora el uso de técnicas de aprendizaje automático para la predicción meteorológica, relevante para la implementación de modelos predictivos en tu proyecto.

Documentación y Guías Técnicas

1. **Arduino Documentation** - Official Arduino Documentation (<https://www.arduino.cc/en/Tutorial/HomePage>)
 - o Documentación oficial de Arduino, incluyendo tutoriales y guías para el uso de sensores y actuadores.
2. **Raspberry Pi Documentation** - Official Raspberry Pi Documentation (<https://www.raspberrypi.org/documentation/>)
 - o Documentación oficial de Raspberry Pi, cubriendo configuraciones y proyectos IoT.

75

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

3. **Dash and Plotly Documentation** - Dash User Guide and Documentation (<https://dash.plotly.com/introduction>)
 - Guía y documentación de Dash y Plotly para la creación de aplicaciones web interactivas y visualización de datos.
4. **Manual Anemómetro Young 03002** - Manual de usuario y hoja de datos. (<https://s.campbellsci.com/documents/es/manuals/03002%20-%20830.pdf>)
 - Hoja de datos y manual de usuario con especificaciones y documentación técnica.

Artículos Técnicos y Blogs

1. **"Building a Smart Irrigation System with IoT"** - Article on Hackster.io.
 - Un artículo paso a paso sobre la construcción de un sistema de riego inteligente utilizando IoT.
2. **"Precision Agriculture with IoT and Machine Learning"** - Blog post on Medium.
 - Un post de blog que describe cómo integrar IoT y aprendizaje automático en la agricultura de precisión.
3. **"Real-time Environmental Monitoring with IoT"** - Article on Adafruit Learning System.
 - Un artículo sobre cómo configurar un sistema de monitoreo ambiental en tiempo real utilizando sensores IoT.
4. <https://www.geohistoarteducativa.net/>
 - Web educativa sobre nociones básicas del clima, geografía, presión atmosférica, temperatura y humedad.

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
 La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49

12. Anexos

Códigos Arduino y Python

<https://github.com/alu0101074306/TFM---Sistema-de-Riego-Automatizado-de-un-Huerto-con-IoT.git>

Este documento incorpora firma electrónica, y es copia auténtica de un documento electrónico archivado por la ULL según la Ley 39/2015.
La autenticidad de este documento puede ser comprobada en la dirección: <http://sede.ull.es/validacion>

Identificador del documento: 6750970 Código de verificación: mZvJrzui

Firmado por: Íñigo Martínez Sánchez UNIVERSIDAD DE LA LAGUNA	Fecha: 20/08/2024 23:04:24
Santiago Torres Álvarez UNIVERSIDAD DE LA LAGUNA	20/08/2024 23:05:12
Juan Albino Méndez Pérez UNIVERSIDAD DE LA LAGUNA	26/08/2024 12:55:49