



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería del Diseño

Diseño e implementación de un sistema electrónico de
riego remoto y automático

Trabajo Fin de Máster

Máster Universitario en Sensores para Aplicaciones Industriales

AUTOR/A: Rodriguez Rodriguez, Alvaro

Tutor/a: Masot Peris, Rafael

Cotutor/a: Alcañiz Fillol, Miguel

CURSO ACADÉMICO: 2022/2023



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escuela Técnica Superior de Ingeniería del Diseño

Universitat Politècnica de València

Escuela Técnica Superior de Ingeniería del Diseño
Máster Universitario en Sensores para Aplicaciones
Industriales

Trabajo de fin de máster

***Diseño e implementación de un sistema electrónico
de riego remoto y automático***

Curso académico: **2022/2023**

Autor: **Álvaro Rodríguez Rodríguez**

Tutor: **Rafael Masot Peris**

Cotutor: **Miguel Alcañiz Fillol**

Valencia, Julio 2023

Resumen

Diseño e implementación de un sistema electrónico de riego remoto y automático

El presente Trabajo de Fin de Máster (TFM) se centra en el diseño e implementación de un sistema electrónico de riego remoto y automático para plantas domésticas de interior. El objetivo principal es proporcionar una solución eficiente y fácil de usar para el cuidado y mantenimiento del riego en plantas de interior, mejorando su crecimiento y salud.

Para lograr esto, se llevarán a cabo las siguientes etapas:

- Investigación y análisis de tecnologías existentes en sistemas de riego para plantas de interior.
- Investigación y análisis de sensores y actuadores compatibles con tarjetas programables con WiFi, para la medición de parámetros físicos como la humedad del suelo, la temperatura ambiente, la humedad ambiente y la activación del riego.
- Implementación de algoritmos de control y optimización en la tarjeta programable para la gestión automática del riego, basados en datos recopilados de los sensores.
- Diseño y programación de un servidor web embebido en la tarjeta Arduino, que permita la configuración y monitorización del sistema de riego a través de una interfaz gráfica de usuario accesible desde un navegador web.
- Integración de tecnologías de comunicación inalámbrica, como WiFi, para permitir el acceso y control remoto del sistema de riego.
- Diseño e implementación del circuito en placas de prototipos perforadas, considerando las restricciones de tamaño y consumo energético en el ámbito doméstico.
- Pruebas y validación del sistema, incluyendo su eficiencia en el uso de agua y la capacidad de adaptación a diferentes condiciones ambientales y tipos de plantas de interior.

El resultado final del TFM será un sistema de riego remoto y automático para plantas domésticas de interior, fácil de implementar y usar, permitiendo a los usuarios optimizar el cuidado de sus plantas y mejorar su entorno doméstico.

Palabras clave: IoT, ESP8266, domótica, riego, planta, temperatura, humedad, sensores, servidor, web.

Autor: Álvaro Rodríguez Rodríguez

Tutor: Rafael Masot Peris

Cotutor: Miguel Alcañiz Fillol

Abstract

Design and implementation of a remote and automatic electronic irrigation system

This Master's Final Project (MFP) focuses on the design and implementation of a remote and automatic electronic irrigation system for domestic indoor plants. The main goal is to provide an efficient and user-friendly solution for the care and maintenance of indoor plant watering, thereby enhancing their growth and health.

To achieve this, the following stages will be carried out:

- Research and analysis of existing technologies in indoor plant watering systems.
- Research and analysis of sensors and actuators compatible with WiFi programmable cards, for the measurement of physical parameters such as soil moisture, ambient temperature, ambient humidity, and activation of watering.
- Implementation of control and optimization algorithms on the programmable card for automatic irrigation management, based on data collected from the sensors.
- Design and programming of a web server embedded in the Arduino card, which allows configuration and monitoring of the irrigation system through a graphical user interface accessible from a web browser.
- Integration of wireless communication technologies, such as WiFi, to allow remote access and control of the irrigation system.
- Design and implementation of the circuit on perforated prototype boards, considering size and energy consumption restrictions in the domestic field.
- Testing and validation of the system, including its water usage efficiency and the ability to adapt to different environmental conditions and types of indoor plants.

The final result of the MFP will be a remote and automatic irrigation system for domestic indoor plants, easy to implement and use, allowing users to optimize the care of their plants and improve their domestic environment.

Keywords: IoT, ESP8266, home automation, irrigation, plant, temperature, humidity, sensors, server, web.

Author: Álvaro Rodríguez Rodríguez

Advisor: Rafael Masot Peris

Co-advisor: Miguel Alcañiz Fillol

Índice general

Resumen.....	II
Abstract	III
Índice general.....	IV
Índice de figuras	V
Índice de tablas	VI
1 Introducción	1
1.1 Contexto y motivación.....	1
1.2 Problemática y desafíos.....	2
2 Objetivos	4
3 Material y métodos	5
3.1 Material	5
3.1.1 Sensores	5
3.1.2 Actuadores	8
3.1.3 Otros elementos.....	10
3.1.4 Resumen.....	13
3.2 Métodos	13
3.2.1 Ideación.....	14
3.2.2 Creación de requisitos.....	14
3.2.3 Elección de materiales.....	14
3.2.4 Diseño y elaboración	14
3.2.5 Programación back-end.....	19
3.2.6 Programación front-end	24
3.2.7 Mejoras.....	28
4 Resultados	29
4.1 Montaje	29
4.2 Interfaz web.....	34
5 Conclusión	35
6 Bibliografía.....	35
7 Anexos	37
7.1 Código de la programación del proyecto	37
7.1.1 Front-end.....	37
7.1.2 Back-end	63

Índice de figuras

Figura 1. Temp-Hum 3 Click - MIKROE-2937.	5
Figura 2. Diagrama de bloques funcionales del HDC2010.	6
Figura 3. Esquema de comunicación I ² C.	6
Figura 4. AZDelivery Sensor de Humedad del Suelo V1.2.	7
Figura 5. Sensor de nivel de agua.	8
Figura 6. Mini bomba de agua sumergible.	9
Figura 7. Tubo de goma.	9
Figura 8. Transistor 2N2222 NPN.	10
Figura 9. Esquema de conexiones del transistor 2N2222.	10
Figura 10. AZDelivery Modulo D1 Mini V3.	11
Figura 11. Fuente de alimentación.	12
Figura 12. Protoboard.	12
Figura 13. Placa de prototipos perforada.	13
Figura 14. Diagrama de bloques del sistema de riego.	15
Figura 15. Botella de agua mineral utilizada como depósito.	16
Figura 16. Esquema de la protoboard con todas sus conexiones y elementos.	17
Figura 17. Imagen del diseño en placa perforada. Imagen del anverso de la placa.	18
Figura 18. Imagen del diseño en placa perforada. Imagen del reverso de la placa.	19
Figura 19. Página principal de la interfaz web.	25
Figura 20. Página de calibrado de la bomba.	26
Figura 21. Código HTML y JavaScript correspondiente al encabezado de la web.	27
Figura 22. Elemento visual al que corresponde el código anterior.	27
Figura 23. Código HTML y JavaScript correspondiente a la sección de "riego manual".	28
Figura 24. Elemento visual al que corresponde el código anterior.	28
Figura 25. Montaje del circuito eléctrico en protoboard.	30
Figura 26. Depósito de agua con bomba y sensor de nivel instalados.	31
Figura 27. Instalación de la manguera alrededor del tallo de la planta.	32
Figura 28. Montaje completo del sistema utilizando la protoboard.	33
Figura 29. Montaje completo del sistema utilizando una placa perforada.	34
Figura 30. Página principal de la interfaz web en funcionamiento.	34
Figura 31. Página de calibrado de la bomba en funcionamiento.	35

Índice de tablas

Tabla 1. <i>Escribir un byte del sensor HDC2010.</i>	6
Tabla 2. <i>Escribir múltiples bytes del sensor HDC2010.</i>	7
Tabla 3. <i>Leer un byte del sensor HDC2010.</i>	7
Tabla 4. <i>Leer múltiples bytes del sensor HDC2010.</i>	7
Tabla 5. <i>Mapa de registros del sensor HDC2010.</i>	7

1 Introducción

1.1 Contexto y motivación

En el mundo actual, la preocupación por la conservación del medio ambiente, la sostenibilidad y la eficiencia en el uso de recursos naturales ha cobrado una importancia creciente. En particular, el agua es un recurso esencial y valioso que enfrenta desafíos de escasez y mal uso en diversas regiones del mundo. Aunque las plantas de interior no representan un consumo de agua comparable al de la agricultura a gran escala, su cuidado adecuado es importante para mantener un entorno doméstico saludable, estéticamente agradable y sostenible.

Las plantas de interior aportan numerosos beneficios a los hogares, como la purificación del aire, la mejora del bienestar emocional, la reducción del estrés y la absorción del ruido. Además, sirven como elementos decorativos que proporcionan vida y color al entorno doméstico. Sin embargo, el cuidado y mantenimiento de las plantas de interior puede ser una tarea desafiante y que consume tiempo, especialmente en lo que respecta al riego. Proporcionar la cantidad adecuada de agua a cada planta es fundamental para su salud y crecimiento, pero esto puede variar según factores como el tipo de planta, las condiciones ambientales, las estaciones del año y las necesidades individuales de cada especie.

En muchos hogares, la falta de tiempo, el desconocimiento de las necesidades específicas de cada planta y los olvidos pueden resultar en un riego inadecuado, lo que lleva a plantas marchitas o enraizadas. Además, en situaciones en las que los propietarios deben ausentarse de su hogar por períodos prolongados, como vacaciones o viajes de trabajo, mantener un riego adecuado y consistente para las plantas de interior puede ser aún más difícil.

La implementación de un sistema electrónico de riego remoto y automático para plantas domésticas de interior puede contribuir significativamente a optimizar el uso del agua, mejorar la salud y el crecimiento de las plantas, y facilitar su cuidado y mantenimiento, incluso en ausencia de los propietarios. Un sistema de riego automático y controlado a distancia permite a los usuarios programar y ajustar el riego según las necesidades de sus plantas, proporcionando una solución flexible y conveniente que se adapta a diferentes situaciones y estilos de vida.

Con el avance de la tecnología y la creciente adopción de dispositivos conectados en el hogar, la automatización y el control remoto de sistemas domésticos se han vuelto más accesibles y asequibles. La implementación de un sistema de riego electrónico que permita el monitoreo y control remoto de las plantas a través de dispositivos como smartphones, tablets y computadoras proporcionaría una solución conveniente y fácil de usar que se adapte a las necesidades de los propietarios y mejore la calidad del entorno doméstico.

Además, la integración de sensores y algoritmos de control inteligente en el sistema de riego puede permitir la adaptación automática del riego a las condiciones ambientales y a las necesidades específicas de cada planta, lo que resulta en un uso más eficiente del agua y un mejor cuidado de las plantas.

Por lo tanto, el diseño e implementación de un sistema electrónico de riego remoto y automático para plantas domésticas de interior es una propuesta relevante y oportuna en el contexto actual de creciente preocupación por la sostenibilidad, la eficiencia en el uso de recursos y la búsqueda de soluciones innovadoras para mejorar la calidad de vida y el entorno doméstico. La posibilidad de mantener el riego adecuado de las plantas de interior durante ausencias prolongadas, como

viajes y vacaciones, representa un valor agregado para este tipo de sistema, ya que proporciona tranquilidad a los propietarios y garantiza el cuidado continuo de sus plantas.

El desarrollo de este tipo de sistema también puede tener implicaciones más amplias en el ámbito de la agricultura urbana y la horticultura doméstica. Un sistema de riego automático y controlado a distancia podría ser aplicado en jardines verticales, huertos urbanos y otros espacios verdes, promoviendo la adopción de prácticas de cultivo sostenibles y eficientes en el consumo de agua en entornos urbanos y residenciales.

En este sentido, también puede contribuir a la concienciación y educación entorno a la importancia del uso responsable del agua y la promoción de prácticas sostenibles. Al proporcionar una herramienta fácil de usar y efectiva para el cuidado de las plantas de interior, este sistema puede inspirar a más personas a incorporar plantas en su entorno doméstico y, en consecuencia, a contribuir a la creación de espacios más verdes, saludables y sostenibles.

Por último, el diseño e implementación de un sistema de riego electrónico también puede representar una oportunidad para explorar la integración de tecnologías emergentes, como la inteligencia artificial y el Internet de las cosas, en la gestión y monitoreo de recursos naturales y sistemas domésticos. Estas tecnologías pueden permitir una mayor precisión, adaptabilidad y eficiencia en la gestión del riego y el cuidado de las plantas, así como ofrecer nuevas perspectivas y enfoques para abordar desafíos relacionados con la sostenibilidad y la conservación del medio ambiente.

En conclusión, el contexto y la motivación detrás del diseño e implementación de un sistema electrónico de riego remoto y automático para plantas domésticas de interior radican en la necesidad de optimizar el uso de recursos naturales, mejorar el cuidado y mantenimiento de las plantas en el hogar y facilitar la adopción de prácticas sostenibles y eficientes en la vida cotidiana. Este proyecto no solo representa una solución innovadora y oportuna para el cuidado de las plantas de interior, sino que también puede tener implicaciones más amplias en la promoción de la sostenibilidad y el uso responsable del agua en entornos urbanos y residenciales.

1.2 Problemática y desafíos

El cuidado de las plantas de interior, aunque a menudo se considera una actividad recreativa y placentera, también implica enfrentar una serie de problemáticas y desafíos que pueden afectar su éxito y el bienestar de las plantas en el hogar. Estos desafíos pueden abordarse mediante el diseño e implementación de un sistema electrónico de riego remoto y automático, pero también es necesario tener en cuenta ciertos obstáculos y limitaciones específicos a lo largo del proceso de desarrollo.

1. Diversidad de necesidades de riego: Existe una amplia variedad de plantas de interior, cada una con sus propias necesidades y requerimientos de riego. La cantidad y frecuencia de riego pueden variar según el tipo de planta, el tamaño, el tipo de suelo y las condiciones ambientales. Diseñar un sistema de riego que sea capaz de adaptarse a estas diferencias y proporcionar la cantidad adecuada de agua a cada planta representa un desafío importante.

2. Condiciones ambientales variables: Las condiciones ambientales, como la temperatura, la humedad y la luz, pueden afectar significativamente las necesidades de riego de las plantas de interior. Un sistema de riego efectivo debe ser capaz de monitorear y adaptarse a estas condiciones, ajustando el riego según sea necesario para mantener la salud y el crecimiento óptimos de las plantas.
3. Limitaciones de espacio y diseño: Los sistemas de riego para plantas de interior deben ser compactos y estéticamente agradables para adaptarse a los espacios limitados y la decoración de los hogares. El diseño del sistema debe tener en cuenta la colocación de los sensores, actuadores y componentes electrónicos, así como la disposición y el tamaño de las tuberías de riego y otros elementos del sistema.
4. Facilidad de uso y accesibilidad: Un sistema de riego electrónico debe ser fácil de instalar, configurar y utilizar, incluso para usuarios sin conocimientos técnicos avanzados. La interfaz de usuario y el proceso de configuración deben ser intuitivos y accesibles, permitiendo a los usuarios monitorear y controlar el riego de sus plantas de interior de manera sencilla y efectiva.
5. Eficiencia energética y consumo de agua: El sistema de riego debe ser eficiente tanto en términos de energía como de consumo de agua, minimizando el desperdicio y optimizando la distribución del agua en función de las necesidades de cada planta.
6. Conectividad y comunicación inalámbrica: La integración de tecnologías de comunicación inalámbrica, como WiFi, en el sistema de riego puede presentar desafíos en términos de alcance, fiabilidad y seguridad. Es necesario garantizar que el sistema pueda comunicarse de manera efectiva con dispositivos de control remoto, como smartphones y tablets.
7. Costo y disponibilidad de componentes: El desarrollo de un sistema de riego electrónico debe tener en cuenta el costo y la disponibilidad de los componentes necesarios, como sensores, actuadores, tarjetas programables y otros elementos electrónicos. El sistema debe ser asequible y accesible para la mayoría de los usuarios, lo que implica buscar soluciones rentables y eficientes sin comprometer la calidad y el rendimiento del sistema.
8. Seguridad y protección de datos: Al tratarse de un sistema conectado a Internet y controlado de forma remota, es fundamental garantizar la seguridad y la protección de los datos de los usuarios. El sistema debe contar con mecanismos de autenticación para prevenir accesos no autorizados y posibles riesgos asociados.

9. Integración con otros sistemas y dispositivos del hogar: En el contexto de hogares inteligentes y la creciente adopción de dispositivos conectados, el sistema de riego electrónico podría ser capaz de integrarse con otros sistemas y dispositivos domésticos. Esto puede incluir la compatibilidad con plataformas de automatización del hogar, como Google Home o Amazon Alexa, lo que permitiría a los usuarios controlar y monitorear el riego de sus plantas de interior a través de un ecosistema más amplio de dispositivos y aplicaciones.

En resumen, el diseño e implementación de un sistema electrónico de riego remoto y automático para plantas domésticas de interior presenta una serie de problemáticas y desafíos que deben abordarse para lograr un sistema efectivo, eficiente y fácil de usar. Estos desafíos incluyen la diversidad de necesidades de riego, las condiciones ambientales variables, las limitaciones de espacio y diseño, la facilidad de uso y accesibilidad, la eficiencia energética y el consumo de agua, la conectividad y comunicación inalámbrica, el costo y disponibilidad de componentes, la seguridad y protección de datos, y la integración con otros sistemas y dispositivos del hogar. A lo largo del desarrollo del proyecto, es esencial tener en cuenta estos aspectos para garantizar un sistema de riego electrónico de alta calidad que satisfaga las necesidades y expectativas de los usuarios.

2 Objetivos

El objetivo principal del TFM es diseñar e implementar un sistema electrónico de riego remoto y automático para plantas domésticas de interior que sea eficiente, fácil de usar y adaptable a diferentes condiciones ambientales y tipos de plantas. Para lograr este objetivo, se llevarán a cabo las siguientes etapas:

- Investigación y análisis de tecnologías existentes en sistemas de riego para plantas de interior.
- Investigación y análisis de sensores y actuadores compatibles con tarjetas programables con WiFi, para la medición de parámetros físicos como la humedad del suelo, la temperatura ambiente, la humedad ambiente y la activación del riego.
- Implementación de algoritmos de control y optimización en la tarjeta programable para la gestión automática del riego, basados en datos recopilados de los sensores.
- Diseño y programación de un servidor web embebido en la tarjeta programable, que permita la configuración y monitorización del sistema de riego a través de una interfaz gráfica de usuario accesible desde un navegador web.
- Integración de tecnologías de comunicación inalámbrica, como WiFi, para permitir el acceso y control remoto del sistema de riego.
- Diseño e implementación del circuito en placas de prototipos perforadas, considerando las restricciones de tamaño y consumo energético en el ámbito doméstico.
- Pruebas y validación del sistema, incluyendo su eficiencia en el uso de agua y la capacidad de adaptación a diferentes condiciones ambientales y tipos de plantas de interior.

3 Material y métodos

En este capítulo, se describirán los materiales utilizados en el desarrollo del sistema electrónico de riego remoto y automático, así como los métodos empleados para su implementación y ensamblaje. Se prestará especial atención a los sensores y actuadores, incluyendo su funcionamiento y los tratamientos previos necesarios para su uso en el sistema. Además, se detallará el diseño y construcción de la placa de circuito y la programación del microcontrolador AZDelivery Módulo D1 Mini V3 en el entorno IDE de Arduino.

3.1 Material

3.1.1 Sensores

En este proyecto, se ha considerado fundamental medir tres condiciones ambientales específicas para asegurar el adecuado riego de las plantas de interior: la temperatura ambiente, la humedad ambiente y la humedad del suelo. La medición de la temperatura y la humedad ambiente permite ajustar la cantidad de agua necesaria para el riego, mientras que la humedad del suelo determina cuándo es el momento adecuado para regar la planta. Además se ha necesitado medir el nivel de agua del depósito. Para realizar estas mediciones, se han seleccionado tres sensores:

1. Temp-Hum 3 Click - MIKROE-2937.
2. AZDelivery Sensor de Humedad del Suelo V1.2.
3. Sensor de nivel de agua.

3.1.1.1 Temp-Hum 3 Click - MIKROE-2937



Figura 1. Temp-Hum 3 Click - MIKROE-2937.

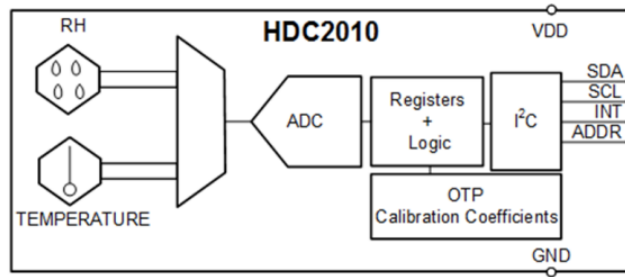


Figura 2. Diagrama de bloques funcionales del HDC2010.

Esta placa incorpora el sensor HDC2010 junto a la electrónica necesaria para hacerlo funcionar. Con su calibrado de fábrica, este sensor permite medir la temperatura y la humedad ambiente con alta precisión y confiabilidad. Los datos recopilados por este sensor se utilizan para ajustar el volumen de agua con la que se riega, teniendo en cuenta las condiciones ambientales específicas.

Entre las características de este sensor, se encuentran:

- Rango de medida de humedad: 0% - 100%.
- Precisión de medida de humedad: $\pm 2\%$.
- Rango de medida de temperatura: -40°C - 125°C .
- Precisión de medida de temperatura: $\pm 0.2^{\circ}\text{C}$.
- Tensión de alimentación: 1.62VDC - 3.6VDC.
- Corriente en modo "Sleep": 50nA.
- Interfaz de comunicación: I²C.

Estas características hacen de este un sensor ideal para la aplicación. Se trata de un sensor con buena precisión de medida, buen rango de medida, tensión de alimentación compatible con la placa programable, un bajo consumo y una interfaz de comunicación I²C.

La interfaz de comunicación I²C permite comunicar el sensor con la placa programable sin ocupar ninguna entrada analógica.

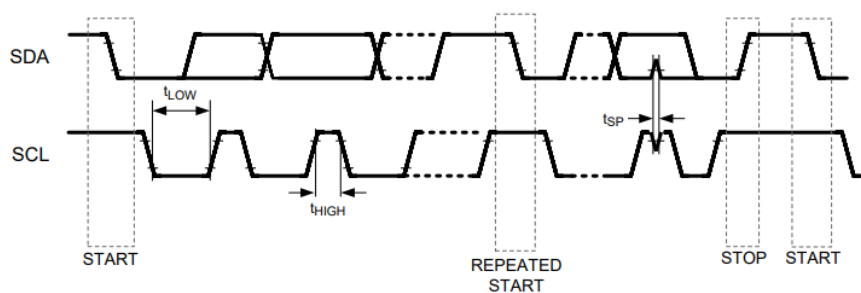


Figura 3. Esquema de comunicación I²C.

La comunicación I²C nos permite leer y escribir registros de memoria del sensor HDC2010. Estos registros se pueden leer y escribir de forma individual o en grupos. En las próximas tablas se muestran las secuencias de comunicación a seguir para leer y escribir registros de memoria del sensor HDC2010.

Tabla 1. Escribir un byte del sensor HDC2010.

Master	START	Slave address (W)		Address		DATA		STOP
Slave			ACK		ACK		ACK	

Tabla 2. Escribir múltiples bytes del sensor HDC2010.

Master	START	Slave address (W)		Address		DATA		DATA		STOP
Slave			ACK		ACK		ACK		ACK		

Tabla 3. Leer un byte del sensor HDC2010.

Master	START	Slave address (W)		Address		Start	Slave address (W)			NACK	STOP
Slave			ACK		ACK			ACK	DATA		

Tabla 4. Leer múltiples bytes del sensor HDC2010.

Master	START	Slave address (W)		Address		Start	Slave address (W)			ACK		ACK	NACK	STOP
Slave			ACK		ACK			ACK	DATA		DATA				

Tabla 5. Mapa de registros del sensor HDC2010.

ADDRESS (HEX)	NAME	RESET VALUE	DESCRIPTION
0x00	TEMPERATURE LOW	0	Temperature [7:0]
0x01	TEMPERATURE HIGH	0	Temperature [15:8]
0x02	HUMIDITY LOW	0	Humidity [7:0]
0x03	HUMIDITY HIGH	0	Humidity [15:8]
0x04	INTERRUPT/DRDY	0	DataReady and interrupt configuration
0x05	TEMPERATURE MAX	0	Max temperature value measured (peak detector)
0x06	HUMIDITY MAX	0	Max humidity value measured (peak detector)
0x07	INTERRUPT MASK	0	Interrupt Mask
0x08	TEMP_OFFSET_ADJUST	0	Temperature offset adjustment
0x09	HUM_OFFSET_ADJUST	0	Humidity offset adjustment
0x0A	TEMP_THR_L	11111111	Temperature Threshold Low
0x0B	TEMP_THR_H	0	Temperature Threshold High
0x0C	RH_THR_L	11111111	Humidity threshold Low
0x0D	RH_THR_H	0	Humidity threshold High
0x0E	RESET&DRDY/INT CONF	0	Soft Reset and Interrupt Configuration
0x0F	MEASUREMENT CONF	0	Measurement configuration
0xFC	MANUFACTURER ID LOW	01001001	Manufacturer ID Low
0xFD	MANUFACTURER ID HIGH	01010100	Manufacturer ID High
0xFE	DEVICE ID LOW	11010000	Device ID Low
0xFF	DEVICE ID HIGH	00000111	Device ID High

3.1.1.2 AZDelivery Sensor de Humedad del Suelo V1.2



Figura 4. AZDelivery Sensor de Humedad del Suelo V1.2.

Este sensor se encarga de medir la humedad del suelo en el que se encuentran las plantas. Los datos proporcionados por este sensor son esenciales para determinar el momento óptimo para

activar el riego, garantizando que las plantas reciban la cantidad adecuada de agua en función de sus necesidades.

Entre las características de este sensor, se encuentran:

- Tensión de alimentación: 5VDC.
- Señal de salida: analógica.
- Tipo de sensor: capacitivo.

El sensor de humedad del suelo opera mediante la medición de la capacitancia del suelo, una propiedad que fluctúa en función de su nivel de humedad. Este sensor transmite los datos obtenidos a través de una señal analógica, lo que requiere que nuestra tarjeta programable esté equipada con al menos un puerto de entrada analógica para interpretar la información. Una de las ventajas significativas de la medición de la humedad del suelo a través de la capacitancia es que este método no implica la contaminación del suelo, además, el sensor no sufre desgaste ni corrosión, proporcionando así mediciones precisas y confiables a largo plazo.

3.1.1.3 Sensor de nivel de agua.



Figura 5. Sensor de nivel de agua.

Este sensor es necesario para detectar cuándo se vacía el depósito de agua. En concreto, el sensor de nivel utilizado es el interruptor ZP4510. Contiene una boya que flota con el agua de forma que esta cierra o abre el interruptor dependiendo del nivel del agua.

3.1.2 Actuadores

Además de los sensores para medir las condiciones ambientales, el sistema de riego requiere actuadores que permitan llevar a cabo el proceso de riego de manera eficiente y precisa. Para ello, se han incluido los siguientes elementos:

1. Mini bomba de agua sumergible.
2. Tubo de goma.
3. Depósito de agua.
4. Transistor 2N2222 NPN.

3.1.2.1 Mini bomba de agua sumergible



Figura 6. Mini bomba de agua sumergible.

Esta bomba es responsable de suministrar el agua necesaria para el riego de las plantas. Su tamaño compacto y bajo consumo de energía la hacen ideal para su uso en sistemas de riego domésticos.

Entre las características de este sensor, se encuentran:

- Caudal de agua: hasta 100L/h.
- Vida útil: 200h de funcionamiento continuo.
- Tensión de alimentación: 5VDC.
- Sumergible

3.1.2.2 Tubo de goma



Figura 7. Tubo de goma.

El tubo de goma se utiliza para transportar el agua desde la bomba de agua sumergible hasta las plantas, asegurando una distribución uniforme y precisa del agua. Se ha utilizado un tubo de 5mm de diámetro.

3.1.2.3 Depósito de agua

Este componente almacena el agua que será utilizada para el riego de las plantas. El depósito debe ser lo suficientemente grande para almacenar una cantidad adecuada de agua y garantizar un suministro constante durante el proceso de riego.

En este caso se ha decidido utilizar una botella de agua mineral de litro y medio modificada para adaptarla a las necesidades del proyecto.

3.1.2.4 Transistor 2N2222 NPN



Figura 8. Transistor 2N2222 NPN.

Este transistor actúa como un interruptor electrónico controlado por la placa programable para activar y desactivar la bomba de agua sumergible según las necesidades de riego determinadas por los sensores.

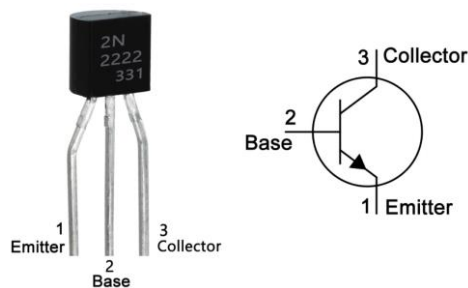


Figura 9. Esquema de conexiones del transistor 2N2222.

Este transistor, cuando se le aplica una pequeña corriente en la base (2), permite pasar corriente del colector (3) al emisor (1). Por otro lado, cuando no hay corriente en la base (2), se corta toda corriente del colector (3) al emisor (1). De esta forma, nuestra placa programable, conectada a la base (2) a través de una resistencia, es capaz de controlar la alimentación de la bomba a través de una fuente de alimentación conectada al colector (3) y el emisor (1) que actuarán como interruptor.

3.1.3 Otros elementos

Aparte de los sensores y actuadores, el proyecto también requiere de otros elementos para llevar a cabo su implementación y funcionamiento:

1. AZDelivery Modulo D1 Mini V3.
2. Fuente de alimentación.
3. Protoboard (para desarrollo).
4. Placa de prototipos perforada (para diseño final).
5. Leds.
6. Cables y resistencias.

3.1.3.1 AZDelivery Modulo D1 Mini V3

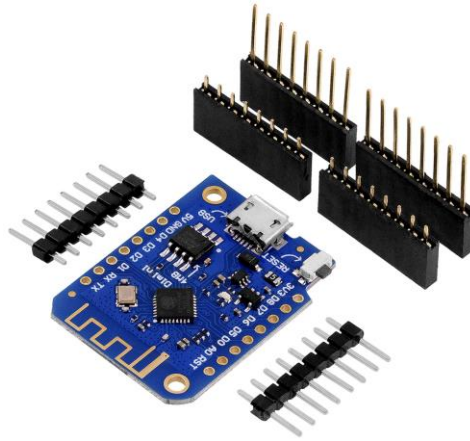


Figura 10. AZDelivery Modulo D1 Mini V3.

El AZDelivery Modulo D1 Mini V3 desempeña un papel central en el sistema de riego automatizado y remoto, ya que se encarga de la recopilación de datos de los sensores, el control de los actuadores y la comunicación con el usuario a través de una interfaz web. Además, es el encargado de procesar los algoritmos de control y optimización para la gestión del riego.

Este módulo es una placa compacta basada en el potente microcontrolador ESP8266, que cuenta con conectividad WiFi integrada. Este chip es ampliamente utilizado en proyectos de IoT (Internet de las cosas) debido a su flexibilidad, bajo consumo de energía y capacidad de conectividad inalámbrica.

Especificaciones técnicas del AZDelivery Modulo D1 Mini V3:

- Microcontrolador: ESP8266-12F.
- Frecuencia de CPU: 80 MHz / 160 MHz.
- Memoria Flash: 4MB.
- Conectividad: WiFi 802.11 b/g/n.
- Entradas y salidas digitales: 11.
- Entradas analógicas: 1 (Rango de 0V a 3.3V).
- Tensión de alimentación: 3.3V.
- Puerto micro USB para programación y alimentación.

En el contexto de este proyecto, la capacidad de conexión WiFi integrada del módulo D1 Mini V3 es crucial ya que permite el acceso y control remoto del sistema de riego. El usuario podrá monitorear y ajustar los parámetros de riego a través de una interfaz gráfica de usuario accesible desde cualquier navegador web, gracias a un servidor web embebido en la tarjeta.

Las entradas y salidas digitales del módulo se utilizarán para la lectura de datos de los sensores y el control de los actuadores. Por ejemplo, el sensor de humedad del suelo proporciona una señal analógica que se lee en la entrada analógica del módulo. En base a estos datos y a los algoritmos de control implementados, el módulo podrá activar la bomba de agua mediante una de sus salidas digitales.

En resumen, el AZDelivery Modulo D1 Mini V3 es el corazón del sistema de riego. Su poder de procesamiento, junto con su conectividad inalámbrica y capacidad de control sobre los sensores y actuadores, permiten el funcionamiento eficiente y autónomo del sistema de riego.

3.1.3.2 Fuente de alimentación



Figura 11. Fuente de alimentación.

La fuente de alimentación suministra la energía necesaria para el funcionamiento de todos los componentes del sistema, incluidos el microcontrolador, sensores, bomba de agua y otros elementos electrónicos.

Con la idea de una alimentación continua que no requiera de estar pendiente del nivel de carga de baterías, se ha decidido utilizar una fuente de alimentación conectada a la corriente de la casa.

En este caso se trata de una fuente de alimentación con un output de 5 V y 2.4 A y un cable micro USB con interruptor.

3.1.3.3 Protoboard (para desarrollo)

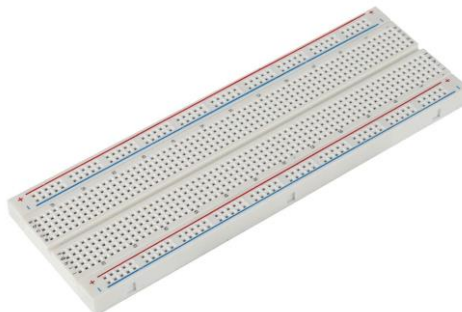


Figura 12. Protoboard.

Durante la fase de desarrollo y prueba del diseño del circuito, se utiliza una protoboard para validar el funcionamiento y compatibilidad de los componentes antes de pasar a la construcción del diseño final en la placa de prototipos perforada.

3.1.3.4 Placa de prototipos perforada (para diseño final)

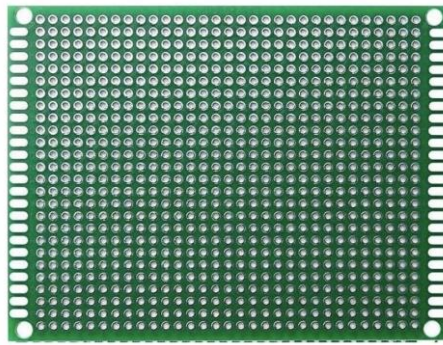


Figura 13. Placa de prototipos perforada.

Una vez validado el diseño en la protoboard, se procede a la construcción del circuito final en la placa de prototipos perforada. Esta etapa implica la colocación y soldadura de los componentes, así como la realización de las conexiones eléctricas necesarias.

3.1.3.5 Leds

Los leds serán utilizados en la fase de desarrollo y prueba del proyecto para comprobar que las salidas digitales de la placa programable se están activando correctamente. También formarán parte del diseño final del proyecto haciendo de indicador visual de la falta de agua en el depósito.

3.1.3.6 Cables y resistencias

Los cables y resistencias son componentes esenciales en el diseño y construcción del circuito electrónico. Los cables permiten la conexión entre los diferentes componentes del sistema, mientras que las resistencias ayudan a controlar el flujo de corriente y proteger los componentes sensibles de posibles daños por sobrecorriente.

3.1.4 Resumen

En resumen, el proyecto de riego automático y remoto requiere una variedad de sensores, actuadores y otros componentes electrónicos para medir las condiciones ambientales, controlar el proceso de riego y garantizar el correcto funcionamiento del sistema. La selección de estos elementos y su correcta integración en el diseño del circuito son cruciales para lograr un sistema de riego eficiente, confiable y fácil de usar, adaptado a las necesidades específicas de las plantas de interior en entornos domésticos.

3.2 Métodos

En este apartado, se detallará el proceso y los métodos utilizados para diseñar e implementar el sistema de riego remoto y automático. Abarcará desde la concepción inicial del diseño del sistema hasta su implementación final, describiendo paso a paso las decisiones técnicas tomadas, las herramientas y tecnologías utilizadas y los desafíos encontrados durante el desarrollo.

Además, se discutirán los métodos empleados para la programación de los algoritmos de control y la creación del servidor web embebido, así como la integración y el funcionamiento de los sensores y actuadores seleccionados.

Finalmente, se discutirán las posibles mejoras y extensiones del sistema que podrían considerarse para trabajos futuros.

El objetivo es proporcionar una visión clara y detallada de cómo se ha llevado a cabo este proyecto, para facilitar su replicación, mejora y adaptación a otras aplicaciones o contextos.

3.2.1 Ideación

La inspiración para este proyecto surgió de una necesidad personal de un familiar cercano, quien requería una solución para mantener sus plantas adecuadamente regadas durante sus frecuentes viajes.

El concepto original contemplaba la creación de un dispositivo de fácil acceso remoto a través de internet, el cual permitiera activar el riego a demanda, suministrando una cantidad específica de agua. Se buscaba un dispositivo sencillo, cuyo desarrollo fuera directo y expedito.

Sin embargo, la idea evolucionó y me propuse añadir una mayor complejidad al proyecto. Surgió la propuesta de desarrollar una interfaz web y la capacidad de personalizar los parámetros de riego, permitiendo al usuario establecer los días específicos para el riego, la hora exacta y el volumen de agua requerido.

Finalmente, cuando se presentó la oportunidad de convertir este proyecto en mi Trabajo de Fin de Máster, decidí ampliar aún más su alcance e incorporar sensores para monitorizar las condiciones ambientales. Además, se implementó la funcionalidad de riego automático, que ajusta el suministro de agua en función de las condiciones detectadas, optimizando así el cuidado de las plantas de interior.

3.2.2 Creación de requisitos

Estos son los requisitos que se establecieron para el proyecto:

- Conexión remota (internet).
- Interfaz web para configuración.
- Capacidad de regar a demanda, de forma programada y de forma automática en función de las condiciones ambientales.
- Capacidad del sistema para detectar la falta de agua en el depósito y avisar al usuario.
- Capacidad del sistema de reportar vía e-mail los eventos importantes.

3.2.3 Elección de materiales

La elección de materiales se realizó en base a una serie de criterios que permitieran el correcto funcionamiento del proyecto y el correcto cumplimiento de los requisitos establecidos. Dichos requisitos son listados a continuación y reflejados en el apartado “Materiales” en este capítulo.

- Usabilidad
- Consumo
- Conectividad
- Durabilidad
- Eficiencia

3.2.4 Diseño y elaboración

En esta parte se va a detallar el diseño y elaboración del sistema de riego y todos sus elementos.

3.2.4.1 Diagrama de bloques

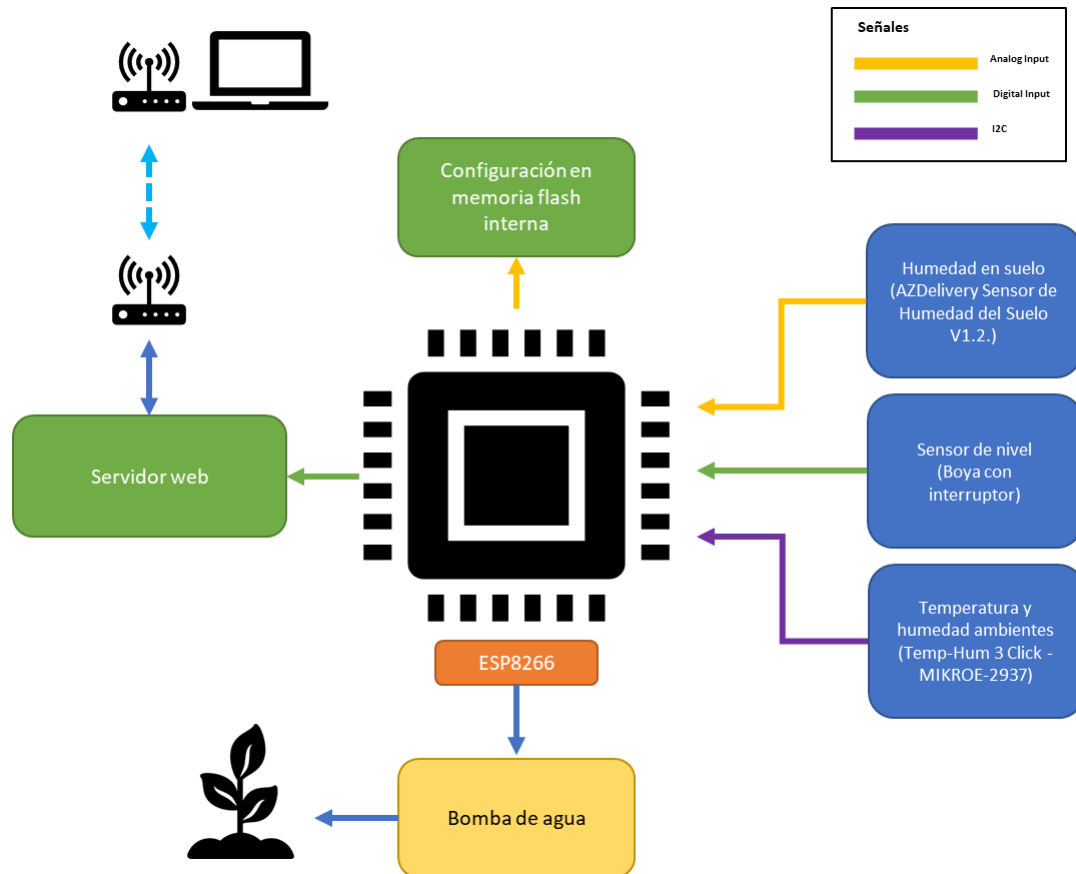


Figura 14. Diagrama de bloques del sistema de riego.

3.2.4.2 Depósito y sistema de riego

La elaboración de un sistema de riego con una mini bomba de bajo consumo requiere, sin lugar a duda, de un depósito de agua desde el cual la bomba pueda bombear agua para regar la planta. En principio podría utilizarse cualquier depósito de un tamaño aceptable desde el cual bombear el agua, pero, como se pretende tener un sistema de detección de falta de agua en el depósito, se necesitan tener algunas características más concretas:

- Forma alargada vertical: este requisito es importante para que la altura del agua en el depósito varíe rápidamente con el volumen de agua en el depósito y poder así detectar de forma más precisa el nivel de agua en el cual queremos considerar que falta agua. Además, de esta forma podemos tener completamente sumergida la bomba con menos cantidad de agua.
- Material de las paredes maleable y fácil de trabajar: como se pretende instalar un sensor de nivel mecánico (una boya) en el depósito, es necesario que las paredes del depósito sean fáciles de trabajar para poder cortar, hacer agujeros, pegar de nuevo o cualquier trabajo que precise la instalación del sensor de nivel.
- Capacidad de dejar el depósito cerrado: puesto que el depósito puede tener agua durante mucho tiempo antes de tener que volver a ser rellenado, es importante poder cerrarlo para evitar la evaporación del agua. Es necesario dejar una pequeña apertura para que no haga vacío cuando la bomba saque agua del depósito.

Tras analizar los requisitos para el depósito de agua, la decisión final fue utilizar una botella de agua mineral. Esta tiene forma alargada vertical, su material es fácil de trabajar y puedes dejarla cerrada utilizando su tapón.

La botella tiene una capacidad de 1,5 L que, a pesar de no ser un depósito demasiado grande, es un volumen razonable sobre todo para el desarrollo del proyecto.



Figura 15. Botella de agua mineral utilizada como depósito.

En la figura anterior puede verse una imagen del modelo de botella que fue utilizada para construir el depósito. También pueden verse unas marcas dibujadas sobre la botella para explicar a continuación las modificaciones que se han realizado a la botella.

1. A esta altura se ha cortado todo el diámetro de la botella para tener la apertura para poder meter la bomba e instalar el sensor de nivel. Una vez instalados el sensor de nivel e introducida la bomba (la cual queda “suelta” apoyada al fondo de la botella), se volvió a pegar la botella dejando un hueco para sacar la manguera y los cables de la bomba.
2. En las marcas dibujadas en la imagen se realizaron dos perforaciones para introducir dos palillos que sujeten el sensor de nivel. Los dos palillos van adheridos a los orificios de la botella y al sensor de nivel con silicona. Además, uno de los orificios es también utilizado para sacar los cables del sensor. La silicona además sella los orificios para que no se produzcan pérdidas de agua.

Haciendo uso de un tubo de goma como el descrito en el apartado “Materiales” de este capítulo, se ha construido una manguera que se bifurca para poder rodear el tallo de la planta a regar. Las uniones entre tubos se han hecho con pegamento. Los dos trozos de manguera que deberían quedar rodeando el tallo de la planta han sido sellados con silicona en sus extremos y perforados varias veces para que el agua salga por las perforaciones. La otra parte de la manguera ha sido unida a la salida de la bomba con pegamento.

3.2.4.3 Protoboard

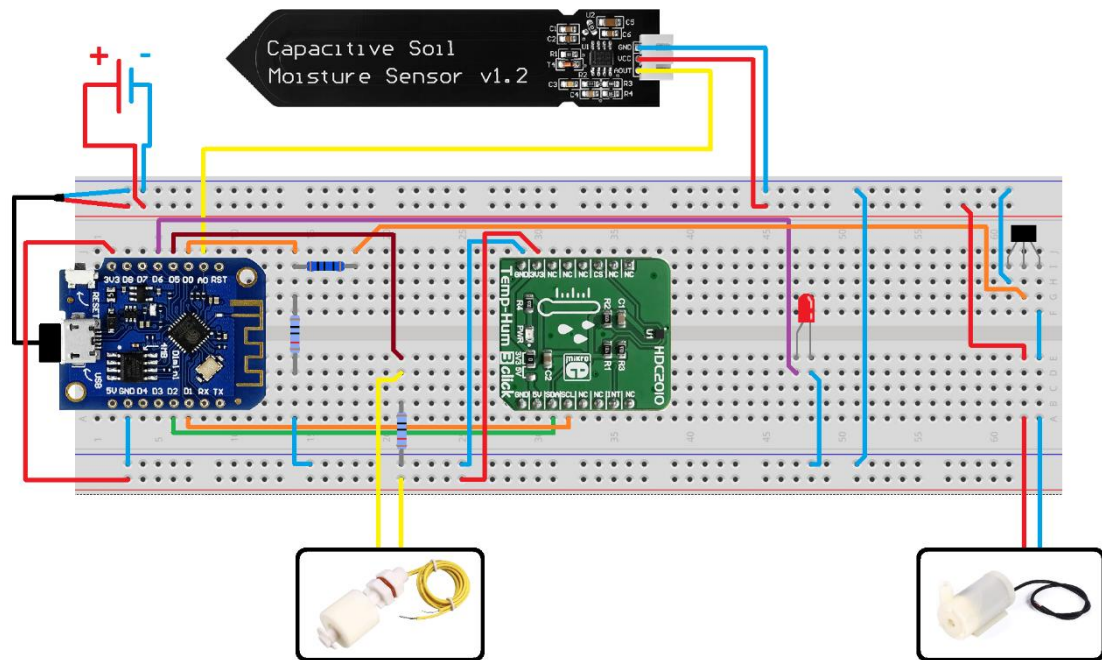


Figura 16. Esquema de la protoboard con todas sus conexiones y elementos.

Para el diseño y desarrollo del proyecto se utilizó una protoboard.

En la figura anterior se muestra el esquema del resultado final en esta fase de desarrollo sobre la protoboard. A continuación, se irá explicando dicho esquema y el trabajo realizado.

En primer lugar, se ha colocado la placa programable de forma que todos los pines de la misma queden utilizables y, además, su puerto micro USB quede accesible. Puesto que esta tarjeta únicamente puede alimentarse a través del puerto micro USB, se utilizó una fuente de alimentación de 5V con cable micro USB con interruptor, y se cortó el cable en dos partes. Una parte de la fuente de alimentación a las dos líneas superiores de alimentación de la protoboard, y la otra parte va de las líneas de alimentación de la protoboard al puerto micro USB de la placa programable. A continuación, se conectó el output de alimentación de 3,3V de la placa y su tierra a la línea de alimentación inferior de la protoboard. Después, las tierras de ambas líneas de alimentación fueron unidas. De esta forma ya tenemos dos líneas de alimentación preparadas para alimentar todos los componentes del circuito.

El siguiente paso ha sido conectar los diferentes elementos:

Sensor de temperatura y humedad (MIKROE-2937): este sensor fue colocado en la protoboard de forma que todos sus pines fueran utilizables. A continuación, se alimentó con la línea de alimentación de 3,3V y se conectaron sus pines "SDA" y "SCL" a los pines "D2" y "D1" de la placa programable respectivamente para realizar la comunicación I²C. Estos pines de la placa programable han sido configurados para dicha comunicación I²C.

Sensor de humedad en suelo (AZDelivery Sensor de Humedad del Suelo V1.2): este sensor ha sido alimentado utilizando la línea de alimentación de 5V de la protoboard. Por otro lado, este sensor transmite su medida a través de una señal analógica, por ello se ha conectado su salida analógica a la entrada analógica "A0" de la placa programable.

Sensor de nivel (boya con interruptor): Este sensor con interruptor ha sido colocado entre el pin digital "D5" de la placa programable configurado como "input" y los 3,3V de la línea de alimentación correspondiente. Además se colocó una resistencia de 10 k Ω entre la entrada digital "D5" y tierra para conseguir que el input lógico de "D5" caiga a "0" cuando se abra el interruptor de la boya.

Mini bomba: Como la placa programable no es capaz de dar corriente suficiente para hacer funcionar la bomba, se ha utilizado un transistor 2n2222 para hacerla funcionar. Para ello se ha conectado la bomba entre el colector y los 5V de la línea de alimentación correspondiente. Después se ha conectado el emisor a tierra y la base al pin "D0" de la placa programable configurado como "output" a través de una resistencia de 100 Ω para evitar una sobre corriente que estropee el transistor.

Led rojo: Este led se ha situado entre el pin "D6" de la placa programable configurado como "output" y tierra. Este led se utiliza para indicar que el depósito se está quedando sin agua.

Con esto, queda definido el diseño final del proyecto. El siguiente paso fue pasar este diseño a una placa perforada donde se obtuviera un producto final visualmente mucho más ordenado, y mucho más robusto al estar todos los elementos soldados a la placa.

3.2.4.4 Placa perforada

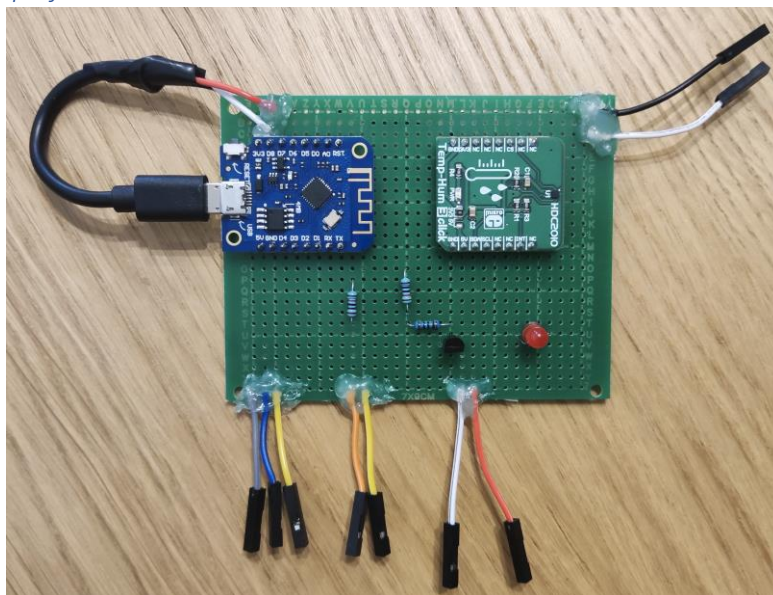


Figura 17. Imagen del diseño en placa perforada. Imagen del anverso de la placa.

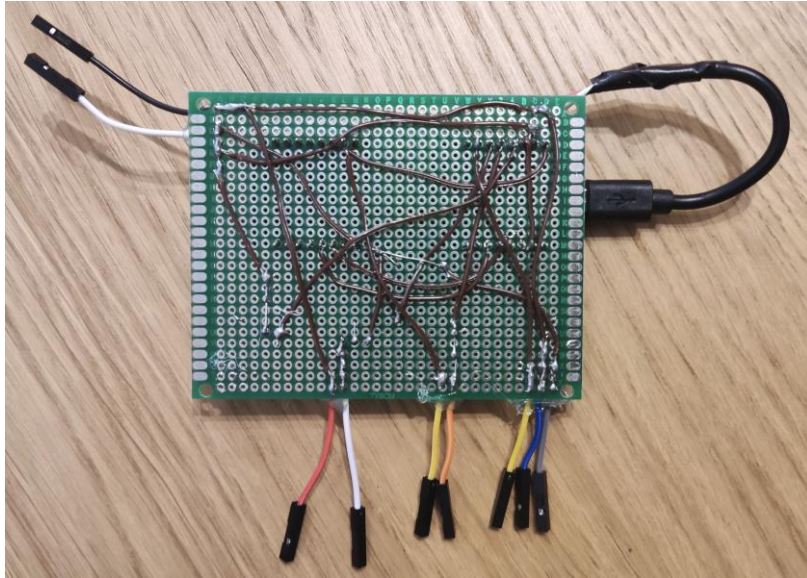


Figura 18. Imagen del diseño en placa perforada. Imagen del reverso de la placa.

Este es el diseño final del proyecto. Para este se ha utilizado una placa perforada sobre la que se han soldado con estaño los diferentes componentes del circuito.

El esquema de conexiones del circuito es exactamente idéntico al explicado para la protoboard. No obstante, el resultado es un producto mucho más compacto, con menos ruido visual y mucho más robusto.

Se ha optado por crear una interfaz entre la placa y los diferentes elementos del circuito utilizando cables hembra de colores a lo que se conectan los cables macho de los mismos colores procedentes de los diferentes elementos del proyecto. La pareja de cables negro y blanco se conectan con la fuente de alimentación. El grupo de cables gris, azul y amarillo se conectan al sensor de humedad en suelo. La pareja de cables naranja y amarillo se conectan al sensor de nivel de agua en el depósito. La pareja de cables blanco y rojo se conectan a la bomba de agua.

Para añadir robustez al diseño, se ha aplicado silicona caliente a los puntos de unión de los cables de la interfaz con la placa perforada. Sin esta silicona, con el movimiento de estos cables al conectar y desconectar los elementos, se acabarían partiendo los cables.

Una vez mostrada y explicada la parte física del proyecto, se va a pasar a hablar de la programación.

3.2.5 Programación back-end

Se ha elaborado un código para que el sistema de riego cumpla las siguientes funciones:

- Capacidad del sistema de calibrar el caudal de la bomba desde la interfaz gráfica web de forma sencilla.
- Capacidad de realizar un riego manual (a demanda) de un volumen determinado de agua. A esto se le ha llamado “riego manual”.
- Capacidad de programar el riego para que realice de forma automática a la hora deseada los días de la semana deseados con el volumen deseado. A esto se le ha llamado “riego programado”.

- Capacidad del sistema de realizar riegos automáticos de forma que, el momento de riego sea decidido en base a la humedad en suelo de la planta, y el volumen de agua con el que se riega sea configurado por la interfaz web pero ajustado en función de la temperatura y humedad ambientes. A esto se le ha llamado “riego automático”.
- Capacidad del sistema de reportar los riegos realizados vía email, así como notificar de la falta de agua en el depósito.

En este apartado se va a explicar programa desarrollado para el proyecto.

El programa completo del proyecto se ha estructurado en 12 archivos:

- RiegoRemoto.ino
- config.h
- handleServer.h
- irrigation.h
- schedule.h
- HDC2010.h
- moisture.h
- automatic.h
- tankLevel.h
- mail.h
- secrets.h
- builtinfiles.h

3.2.5.1 *RiegoRemoto.ino*

Este archivo contiene el código que la placa programable va a ejecutar. Al inicio de este documento se añaden utilizando el comando “#include” el resto de archivos del programa.

En este fichero se definen algunas constantes y se ejecutan dos funciones principales del programa: “setup(void)” y “loop(void)”.

La función “setup(void)” se ejecuta una sola vez al iniciar la placa programable. En ella se ejecutan todas las funciones necesarias para inicializar los diferentes componentes del código que necesitan ser inicializadas y poner en marcha algunos servicios.

A continuación se muestran las diferentes funciones realizadas dentro de “setup(void)”:

1. Se inicializan los diferentes pines digitales como entradas, salidas o comunicación I²C.
2. Inicialización del puerto serie para poder visualizar mensajes en la consola del IDE de Arduino para poder hacer depuración de errores.
3. Inicialización del sistema de ficheros para poder crear un fichero de configuración.
4. Creación del objeto de configuración y, en caso de no existir el fichero de configuración en el sistema de ficheros, se realiza su creación con valores de configuración por defecto. Si ya existe el fichero de configuración, su configuración es cargada en el objeto de configuración del programa.
5. Inicialización del módulo WiFi.
6. Configuración de la IP de la placa programable y conexión al router.
7. Inicialización del cliente NTP para asignar correctamente la hora de la placa programable.

8. Se crean los endpoints del servidor con las páginas que se visualizarán y con las funciones del servidor (handles).
9. Se inicia el servidor.
10. Si el “riego programado” está activado, se pone en marcha la alarma que se encargará de llevar a cabo dicho riego.

La función “loop(void)” se ejecuta después de la función “setup(void)”, pero con la diferencia de que “loop(void)” se ejecutará en bucle mientras la placa programable siga encendida.

A continuación se muestran las diferentes funciones realizadas dentro de “loop(void)”:

1. Se ejecuta la gestión de las solicitudes de los clientes.
2. Se comprueba que el nivel de agua en el depósito sea correcto.
3. Se comprueba si se cumplen las condiciones para realizar un “riego automático”.
4. Se llama a las alarmas programadas y se espera 10 milisegundos antes de volver a ejecutar “loop(void)”.

3.2.5.2 *config.h*

En este archivo se define la clase “Configuration”. Esta clase contiene todos los parámetros de configuración necesarios:

1. “filename_”: nombre del archivo en el que se guarda la configuración.
2. “calibrationVolume_”: volumen utilizado para el cálculo del calibrado.
3. “calibrationTime_”: tiempo utilizado para el cálculo del calibrado.
4. “scheduledIrrigationHour_[3]”: array que contiene horas minutos y segundos a los que se tiene que ejecutar un “riego programado”.
5. “scheduledIrrigationDays_[7]”: array que contiene la información de cuales de los 7 días de la semana se debe realizar un “riego programado”.
6. “scheduledIrrigationVolume_”: volumen de agua que debe utilizarse en los “riegos programados”.
7. “scheduledIrrigationEnabled_”: indica si los “riegos programados” están habilitados o no.
8. “automaticIrrigationVolume_” volumen de agua que debe utilizarse en los “riegos automáticos”.
9. “automaticIrrigationMoistureLimit_”: porcentaje de humedad límite que, si la humedad del suelo baja de dicho valor, se realizará un “riego automático”.
10. “automaticIrrigationEnabled_”: indica si los “riegos automáticos” están habilitados o no.
11. “notificationsEnabled_”: indica si las notificaciones vía email están habilitadas o no.

En esta clase se ha definido un constructor que obliga a darle un nombre al archivo de configuración que se utilizará para guardar y leer la configuración.

Además, se han definido diferentes métodos para interactuar con la configuración:

1. “bool fileExists()”: este método te devuelve “true” si el fichero de configuración existe.
2. “void setDefaultConfig()”: este método se utiliza para darle un valor “default” a todos los parámetros de configuración.
3. “bool load()”: este método carga desde el fichero de configuración los valores guardados y los escribe en los parámetros en RAM. Devuelve “true” si la carga de la configuración se ha realizado con éxito.

4. “bool save()”: escribe en el fichero de configuración los valores de todos los parámetros que están en RAM. Devuelve “true” si los parámetros han sido guardados en el fichero de configuración con éxito.
5. Métodos para establecer configuraciones: se ha creado un método para escribir cada parámetro de configuración.
6. Métodos para leer configuraciones: se ha creado un método para leer cada parámetro de configuración.

3.2.5.3 *handleServer.h*

En este archivo se han creado las funciones que se asociarán a cada endpoint para interactuar con el front-end de la web.

3.2.5.4 *irrigation.h*

En este archivo se han creado dos funciones.

Una de las funciones, “void irrigation(float volume)”, es la función que se llama siempre que quiere realizarse un riego de cualquier tipo. A esta función se le pasa el volumen de agua que quiere utilizarse para regar, y ella se encarga de calcular el tiempo que tiene que encender la bomba en función de los valores de calibración de la misma, y ejecutar la acción de riego.

La otra función “void finishIrrigation()” se encarga de parar la bomba. La utiliza la función anterior cuando transcurre el tiempo de riego calculado.

3.2.5.5 *schedule.h*

En este archivo se han creado las funciones necesarias para gestionar todo lo relacionado con el “riego programado”. Se han definido tres funciones.

La primera de ellas es “void scheduledIrrigation()”. Esta función se llama cuando la alarma con la hora configurada salta, y comprueba si este día de la semana tocaba riego. En función del resultado de la comprobación, realizará el riego o no.

La segunda función es “void updateScheduledIrrigationAlarm()”. Esta función elimina la alarma que exista (si existe), y crea una nueva con la hora existente en la configuración. A esta alarma se le asigna la ejecución de la función anterior.

Por último tenemos la función “void removeScheduledIrrigationAlarm()”. Esta función elimina la alarma que exista (si existe).

3.2.5.6 *HDC2010.h*

Este archivo contiene las funciones necesarias para realizar medidas de temperatura y humedad ambientes utilizando el sensor HDC2010.

Son cuatro las funciones creadas.

En primer lugar, la función “void writeRegister(uint8_t reg, uint8_t value)” se encarga de escribir registros del sensor vía I²C.

Después, la función “uint16_t readRegister(uint8_t reg)” se encarga de leer registros del sensor vía I²C.

Por último las funciones “float readTemperature()” y “float readHumidity()” se encargan de, utilizando las dos funciones anteriores, leer los valores de temperatura y humedad del sensor respectivamente.

3.2.5.7 *moisture.h*

En este archivo se ha creado la función necesaria para leer valores de humedad en suelo con el sensor capacitivo.

Esta función se llama “int readSoilMoisture()” y utiliza la entrada analógica “A0” de la placa programable para realizar la medida.

3.2.5.8 *automatic.h*

En este archivo se han creado las funciones necesarias para gestionar todo lo relacionado con el “riego automático”.

En primer lugar está la función “bool needAutomaticIrrigation()”, que realiza una lectura de la humedad en suelo y devuelve “true” si es necesario realizar un “riego automático”.

Después tenemos las funciones “float getVolumeModification()” y “int getModifiedVolume(int volume)” que se encargan de devolver el volumen de agua utilizado para hacer el “riego automático”. Este volumen se calcula partiendo del volumen de riego configurado para el “riego automático” y ajustándolo en $\pm 20\%$ en función de la temperatura y humedad ambiente.

Por último, la función “void automaticIrrigation()” es la encargada de ejecutar el “riego automático”.

Se ha implementado el código de tal forma que no se puedan realizar “riegos automáticos” con un intervalo menor a seis horas.

3.2.5.9 *tankLevel.h*

Este archivo contiene las funciones necesarias para gestionar el estado de bajo nivel de agua en el depósito. Este estado detendrá cualquier riego, bloqueará la posibilidad de cualquier riego y encenderá el led rojo destinado a este estado.

Las funciones “void setLowTankLevel()” y “void removeLowTankLevel()” se encargan de establecer y quitar el estado de bajo nivel de agua en el depósito y de encender y apagar el led correspondiente.

Por otro lado, la función “void processTankLevel()” se encarga de leer el estado del sensor de nivel para decidir si alguna de las dos funciones anteriores deben ser llamadas. Es necesario que la lectura del sensor de nivel sea estable durante al menos 5 segundos para considerar que ha habido un cambio en su valor.

3.2.5.10 *mail.h*

Este archivo contiene la configuración y las funciones necesarias para enviar email de notificación de los diferentes eventos de la aplicación.

La función encargada de realizar el envío de los emails es “void sendMailReporting(String subject, String body)”.

Esta función se utiliza para notificar (en caso de que las notificaciones estén habilitadas) diferentes eventos de la aplicación:

1. Riego manual realizado: se envía un email al usuario notificando que un “riego manual” ha sido realizado y el volumen utilizado.
2. Riego programado realizado: se envía un email al usuario notificando que un “riego programado” ha sido realizado y el volumen utilizado.

3. Riego automático realizado: se envía un email al usuario notificando que un “riego automático” ha sido realizado, el volumen utilizado, la temperatura y humedad ambientes en el momento de realizar el riego, la humedad en suelo en el momento de realizar el riego, la humedad en suelo 2 minutos después de realizar el riego y un aviso de que no se realizará ningún “riego automático” hasta pasadas seis horas desde el último “riego automático”.

3.2.5.11 *secrets.h*

Este archivo contiene el usuario y la contraseña del WiFi.

3.2.5.12 *builtinfiles.h*

Este archivo contiene los html utilizados para el front-end del servidor. Estos html se han codificado en arrays de bytes para hacerlos más ligeros y mejorar el rendimiento del servidor web.

3.2.6 Programación front-end

La programación del front-end del servidor web se ha realizado mediante archivos html. Se ha escrito un archivo html para cada página que tenga que mostrar el servidor.

Cada uno de estos archivos html contiene toda la información tanto de la parte gráfica como de las funciones a realizar. A continuación, se resume el contenido de los documentos html:

1. Definición de estilos de toda la parte gráfica para obtener un aspecto moderno y dinámico.
2. Definición de todos los textos y campos de entrada de datos.
3. Definición de las diferentes secciones de la visualización.
4. Definición, utilizando JavaScript, de todas las funciones a realizar por el front-end. Estas funciones incluyen desde limitaciones a la hora de introducir información en los campos, hasta todas las peticiones del front-end al back-end, pasando por transiciones de elementos gráficos o la aparición de mensajes “pop-up” mostrando información del back-end como un correcto guardado de la configuración o una posible pérdida de la comunicación con el servidor.

Estos archivos html, una vez terminado, fueron convertidos a un array de bytes e introducidos en el código en el archivo “builtinfiles.h”. Estos arrays de bytes son los que el servidor envía al navegador cuando este se lo pide para poder mostrar el front-end.

A continuación, se va a mostrar la interfaz gráfica desarrollada y se va a explicar su diseño y funcionamiento.

En primer lugar, en la siguiente figura se muestra una imagen de la página principal de la interfaz web. Desde aquí se controlan y configuran todas las funciones del sistema.



Figura 19. Página principal de la interfaz web.

Podemos dividir esta página en cuatro secciones: encabezado, riego manual, riego programado y riego automático. A continuación, se van a explicar todas las funciones y cómo funciona cada sección.

- **Encabezado:** Esta sección contiene el título de la aplicación, un botón que se encarga de habilitar y deshabilitar las notificaciones vía email, y dos botones de navegación. Uno de los botones de navegación, el que tiene el icono de una tuerca, lleva a la pantalla de calibración de la bomba, y el otro botón debería de llevar a una página desde donde gestionar la historización de datos, pero puesto que esto es sólo una mejora planteada y no existe, este botón no lleva a ningún sitio.
- **Riego manual:** En esta sección se gestiona el “riego manual”, en ella hay un campo donde introducir el volumen con el que se desea regar, y un botón que ejecuta el riego.
- **Riego programado:** En esta sección se gestiona el “riego programado”, en ella hay un campo para indicar el volumen con el que se quiere regar, una línea de botones para indicar qué días de la semana queremos que se realicen los “riegos programados”, y tres campos para indicar la hora, minutos y segundos del día a los que queremos que se realice nuestro riego programado. Además, hay un botón con el texto “Programar” que es el que hay que pulsar para que se aplique la configuración introducida, y un botón que se utiliza para habilitar el “riego programado”.
- **Riego automático:** En esta sección se gestiona el “riego automático”, en ella tenemos un campo en el que introducir el volumen de agua con el que queremos que se riegue, un campo donde introducir el porcentaje umbral de humedad en suelo para realizar un “riego automático”, y los valores en tiempo real de temperatura y humedad ambientes y humedad en el suelo. Además, hay un botón con el texto “Configurar” que es el que hay que pulsar para que se guarde la configuración introducida, y un botón que se utiliza para habilitar el “riego automático”.

A continuación, en la siguiente figura se va a mostrar una imagen de la página de calibrado de la bomba.



Figura 20. Página de calibrado de la bomba.

Podemos dividir esta página en dos secciones: encabezado y calibrado.

- **Encabezado:** Esta sección contiene el título de la página y un botón de navegación que te devuelve a la página principal.
- **Calibrado:** Esta sección es la que permite realizar un calibrado del caudal de la bomba de forma cómoda. En ella se encuentra un campo para añadir un volumen de agua y un campo para añadir un tiempo. Con esto indicaremos cuanto tiempo tarda la bomba en bombear cierto volumen de agua. Cuando tengamos introducidos los datos correctos, hay que darle al botón de Guardar calibrado para que estos datos sean guardados. También se pueden encontrar en esta sección dos botones, “START” y “STOP”. Estos botones facilitan la medida de tiempo de la calibración. Cuando le damos a “START”, la bomba empieza a bombear agua y el sistema empieza a contar el tiempo transcurrido. Cuando le damos a “STOP”, la bomba deja de bombear agua y el sistema pone el tiempo transcurrido en el campo de tiempo. Ahora sólo queda anotar el volumen bombeado en el campo de volumen y darle a “Guardar calibrado”.

A continuación, se van a mostrar algunos ejemplos de código empleados en la implementación.

```

<body>
<div id="header">
  <div class="nav">
    <h1 id='t1'>RIEGO REMOTO</h1>
    <p id='t2'>Configura tu riego, tu plantita te lo agradecerá</p>
    <div class="slidernotifspace">
      <span id="t3" style="...">NOTIFICACIONES</span>
      <label class="switch" style="...">
        <input type="checkbox" id="notificationsEnable">
        <span class="slider"></span>
      </label>
    </div>
    <a href="/calibrarbomba">
      <div class="iconspace">
        </div>
      
    </a>
    <a href="/historicaldata">
      <div class="iconspace" style="...">
        </div>
      
</div>

```

Figura 21. Código HTML y JavaScript correspondiente al encabezado de la web.

Este código corresponde al encabezado de la web. Contiene una parte de código en JavaScript que se utiliza para comunicar las acciones realizadas sobre los botones de dicho encabezado al back-end y para establecer un tiempo máximo de espera del front-end para recibir respuesta del back-end. Una vez pasado el tiempo de espera, el navegador mostrará una ventana emergente notificando que se ha perdido la conexión con el servidor.



Figura 22. Elemento visual al que corresponde el código anterior.

```

<div class="control">
  <span class="controltitlespace" style="..."><controltitle>RIEGO MANUAL</controltitle></span><br />
  <form method="post" action="/startManualIrrigation" onsubmit="startManualIrrigation(event); return false;">
    <label id='t3' for="manualIrrigationVolume">Volumen</label>
    <div class="input-container" style="...">
      <input type="text" name="manualIrrigationVolume" id="manualIrrigationVolume" value="100">
      <label id='t3' type="unit" style="...>manualIrrigationVolume">mL</label>
    </div>
  </form>
  <hr class="myline"><br />
  <button id='cmd' type="submit" onclick="startManualIrrigation(event)">Regar</button>
  <script>
    function startManualIrrigation(event) {
      event.preventDefault();
      var xhr = new XMLHttpRequest();
      var timeout = false;
      var timeoutDuration = 3000; // 3 segundos
      var timeoutMessage = "No se ha recibido respuesta del servidor. Por favor, inténtalo de nuevo.";

      xhr.open("POST", "/startManualIrrigation", true);
      xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");

      // Establecer el tiempo de espera
      var timer = setTimeout(function () {
        timeout = true;
        xhr.abort(); // Abortar la solicitud
        alert(timeoutMessage);
      }, timeoutDuration);

      xhr.onreadystatechange = function () {
        if (xhr.readyState === XMLHttpRequest.DONE) {
          clearTimeout(timer); // Limpiar el temporizador si se recibe la respuesta
          if (!timeout && xhr.status === 200) {
            // Handle response from backend here
            var responseText = xhr.responseText;
            alert(responseText);
          }
        }
      };

      var params = "manualIrrigationVolume=" + encodeURIComponent(document.getElementById("manualIrrigationVolume").value);
      xhr.send(params);
    }
  </script>
</div>

```

Figura 23. Código HTML y JavaScript correspondiente a la sección de “riego manual”.

Este código corresponde a la sección de riego manual. En este caso, el código JavaScript se encarga de comunicarle al back-end la decisión de realizar un riego automático cuando se pulse el botón “Regar”, establecer un tiempo máximo de espera de la misma manera que en el ejemplo anterior, y de recoger y mostrar la respuesta del back-end indicando que el “riego manual” ha sido realizado.



Figura 24. Elemento visual al que corresponde el código anterior.

3.2.7 Mejoras

En este apartado se va a hablar de futuras mejoras y desarrollo para perfeccionar la aplicación.

1. Calibración vía web del sensor de humedad en suelo. Los sensores de humedad en suelo requieren de una calibración dándole un valor del sensor al aire y un valor del sensor

- sumergido en agua. Estos valores actualmente se han medido e introducido en el código como constantes. Sería interesante poder introducirlos vía web.
2. Hacer que los botones “Programar” y “Configurar” de los riegos “programado” y “automático” aparezcan grises (inactivos) hasta que algún campo sea modificado y se vuelvan azules (activos). Esto haría la aplicación más intuitiva.
 3. Evitar que se puedan tener activos los riegos “programado” y “automático” a la vez. Actualmente se pueden tener ambos activos a la vez, pero se ha considerado que no tiene sentido poder hacer esto y que, en caso de que algún usuario tuviera ambos riegos activos a la vez, probablemente se trataría de un error.
 4. Configurar un servidor DNS para acceder al servidor web utilizando el hostname. Actualmente hay que utilizar la dirección IP para acceder al servidor web.
 5. Añadir un proceso de autenticación para acceder al servidor web. Actualmente se puede acceder al servidor web sin ningún control ni necesidad de introducir ninguna contraseña.
 6. Desarrollo de la página de datos históricos. Poder registrar internamente en el servidor web los riegos realizados y los datos ambientales y de humedad en suelo para poder mostrarlos en una página donde se generen gráficas y diferentes visualizaciones de interés.
 7. Existe la idea de hacer este proyecto más grande creando un servidor web con capacidad de gestionar el riego de múltiples plantas. Con esta idea, la placa programable de cada planta sería un cliente que se conectaría al servidor central desde el cual se gestionaría su riego y configuración.

4 Resultados

En esta sección se presentará el sistema de riego construido en su totalidad, tanto desde una perspectiva física como desde la interfaz web desarrollada para su control y monitorización. Asimismo, se expondrán las propuestas de mejora y ampliación consideradas para continuar con el desarrollo del sistema, con el fin de mejorar su rendimiento y aumentar las funcionalidades ofrecidas al usuario.

4.1 Montaje

En esta parte se mostrará el resultado obtenido del montaje físico del sistema de riego remoto.

Como se ha descrito hasta el momento, hubo una primera fase de desarrollo en la cual se realizó el montaje electrónico en una protoboard. A continuación se muestra una imagen del resultado de dicho montaje:

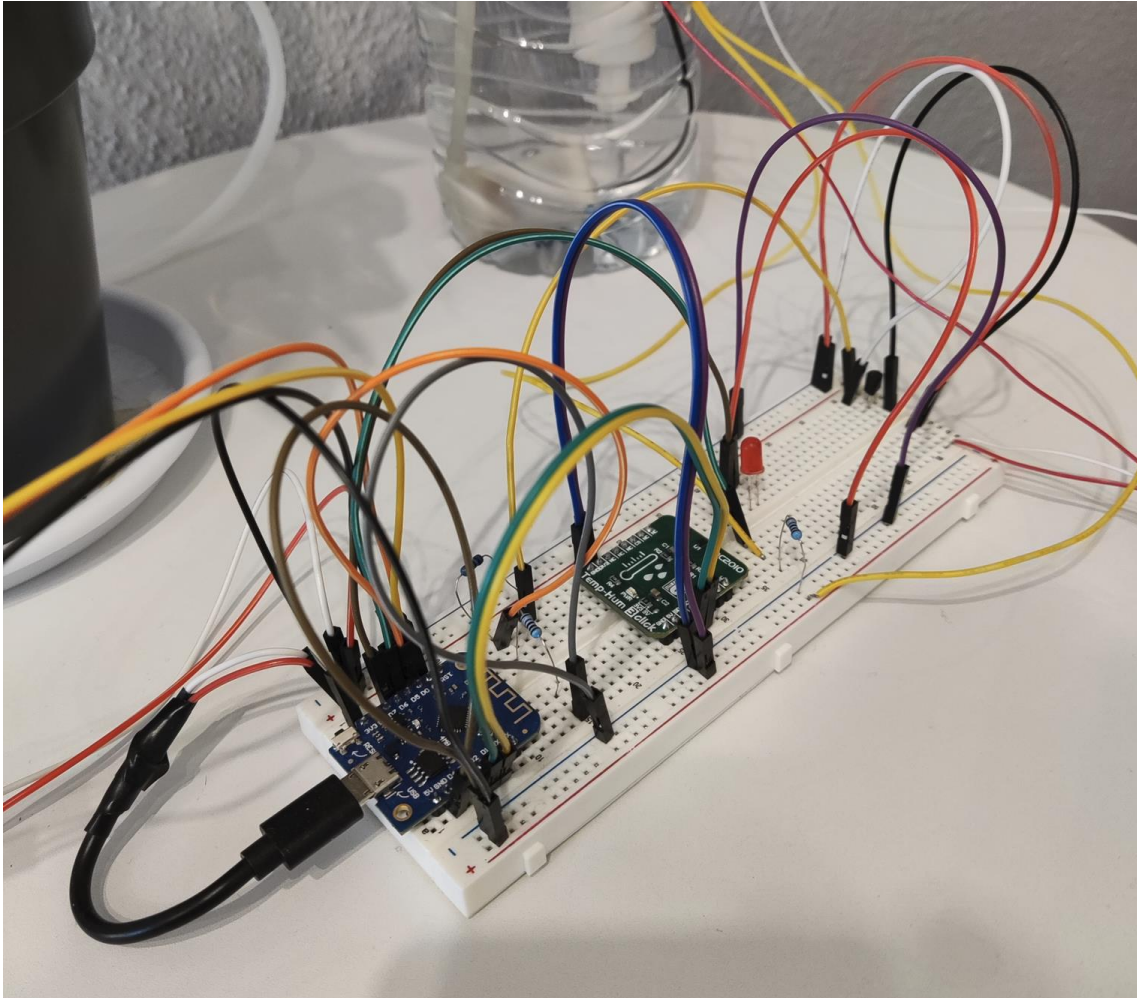


Figura 25. Montaje del circuito eléctrico en protoboard.

A continuación, se muestra una imagen del resultado de construir el depósito de agua con la bomba y el sensor de nivel instalados:

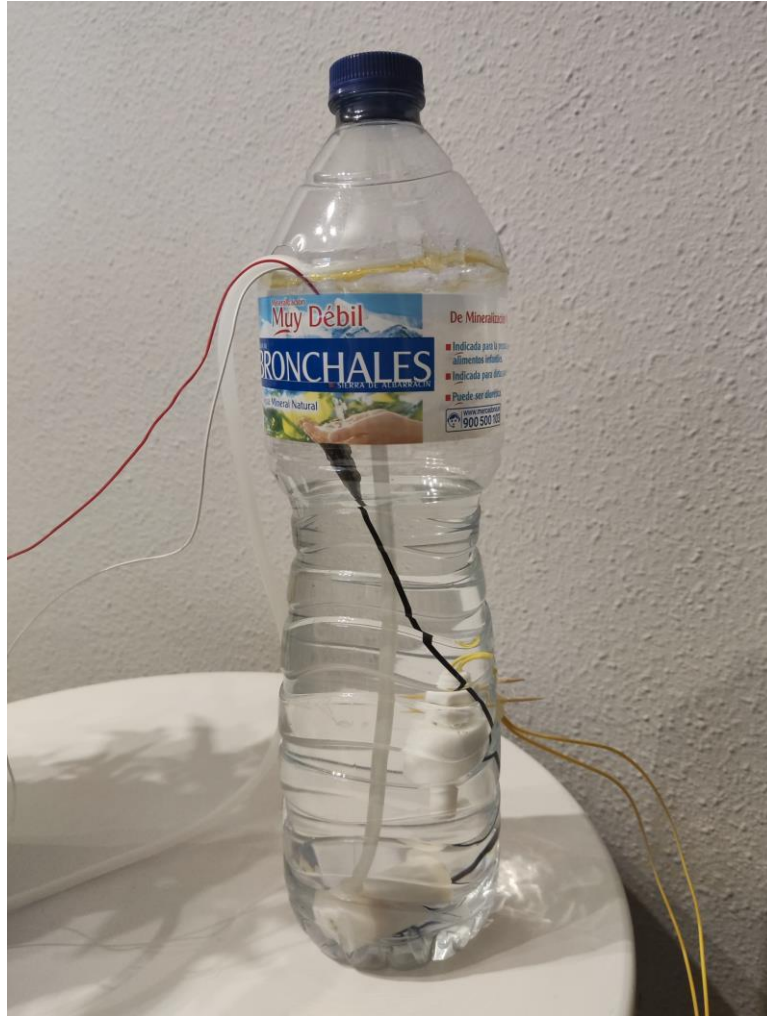


Figura 26. Depósito de agua con bomba y sensor de nivel instalados.

En la siguiente figura se muestra la instalación de la manguera alrededor del tallo de la planta. A esta manguera se le han realizado cuatro perforaciones por donde sale el agua.



Figura 27. *Instalación de la manguera alrededor del tallo de la planta.*

Por último, en la siguiente figura, puede observarse el montaje completo utilizando la protoboard. En esta imagen también puede observarse la instalación del sensor de humedad en suelo.

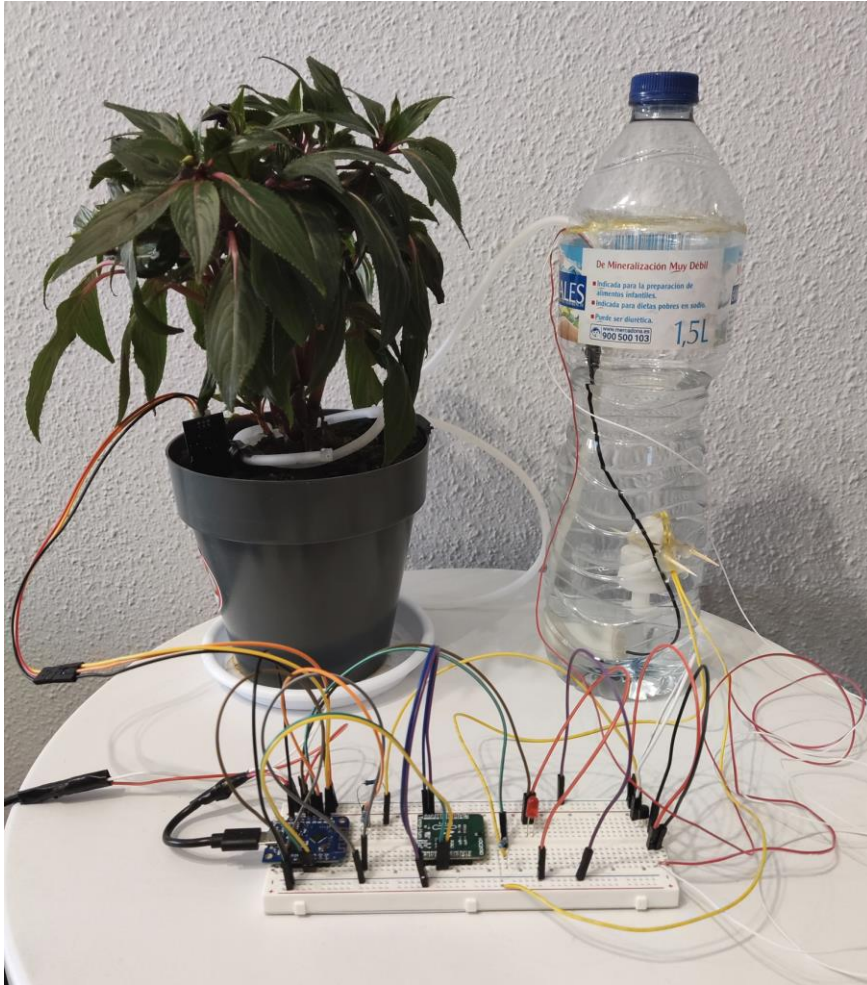


Figura 28. Montaje completo del sistema utilizando la protoboard.

El siguiente paso fue el de sustituir la protoboard por la placa perforada. Las imágenes del montaje del circuito en la placa perforada pueden verse en las figuras 17 y 18.

Y a continuación se muestra el resultado final del sistema completo utilizando la placa perforada:



Figura 29. Montaje completo del sistema utilizando una placa perforada.

4.2 Interfaz web

En este apartado se van a mostrar capturas de pantalla de la interfaz gráfica en funcionamiento:



Figura 30. Página principal de la interfaz web en funcionamiento.



Figura 31. Página de calibrado de la bomba en funcionamiento.

5 Conclusión

A través de este Trabajo de Fin de Máster, se ha logrado diseñar e implementar con éxito un sistema de riego remoto y automático para plantas de interior. El sistema proporciona una solución eficiente y fácil de usar para el cuidado y mantenimiento del riego en plantas de interior, lo que permite mejorar su crecimiento y salud.

El análisis exhaustivo de las tecnologías existentes y la elección cuidadosa de los sensores y actuadores compatibles con tarjetas programables con WiFi, permitió el desarrollo de un sistema de riego que se puede gestionar de forma automática, basado en datos recopilados de los sensores. La incorporación de un servidor web embebido en la tarjeta Arduino ha proporcionado una interfaz gráfica de usuario accesible desde un navegador web, facilitando la configuración y monitorización del sistema de riego.

El diseño final del sistema, implementado en placas de prototipos perforadas y considerando restricciones de tamaño y consumo energético, ha demostrado ser eficaz.

Este trabajo ha abierto nuevas vías para la investigación y el desarrollo de sistemas de riego inteligentes y eficientes. Aunque se han logrado resultados satisfactorios, existe un potencial considerable para mejoras futuras y ampliaciones del sistema, que podrían incluir la integración de más tipos de sensores, la optimización de los algoritmos de control y la mejora de la interfaz de usuario, como ya se ha mencionado en el apartado anterior.

En conclusión, el trabajo realizado en este Trabajo de Fin de Máster ha dado lugar a un sistema de riego remoto y automático eficiente y funcional, proporcionando una solución de riego efectiva y adaptable para plantas de interior. El desarrollo de este sistema ha sido un desafío gratificante, y espero que sirva de inspiración y base para futuros proyectos en este campo.

6 Bibliografía

- Placa programable WiFi (AzDelivery WeMos D1 mini):

https://www.wemos.cc/en/latest/d1/d1_mini.html#documentation

- Datasheet Temp Hum click 3:

<https://download.mikroe.com/documents/datasheets/hdc2010.pdf>

- Datasheet Sensor humedad en suelo:

https://cdn.shopify.com/s/files/1/1509/1638/files/Hygrometer_V1.2_Sensor_Modul_Datenblatt_AZ-Delivery_Vertriebs_GmbH.pdf?v=1608545956

- Datasheet transistor 2n2222:

<https://pdf1.alldatasheet.com/datasheet-pdf/view/15067/PHILIPS/2N2222.html>

7 Anexos

7.1 Código de la programación del proyecto

7.1.1 Front-end

7.1.1.1 *mainpage.html*

```
<!DOCTYPE html>
<html>
<head>
  <title>RIEGO REMOTO</title>
  <link rel="shortcut icon" href="https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcTxYG8RqjMR8bdewCu6ZVm-
MHRCWtftogDd2AJ1uw0LBM5NWBtcOhaAAamAlgsjLSF_p4g&usqp=CAU">
  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
  <meta name="viewport" content="width=device-width, initial-
scale=1">
  <style type="text/css">
    *{
padding: 0px; margin: 0px;
}
body {
font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}

form {
display: flex;
flex-direction: column;
max-width: 300px;
margin: auto;
}

.input-container {
display: flex;
justify-content: center;
align-items: center;
margin-bottom: 10px;
position: relative;
}

label[type="unit"] {
position: absolute;
}

input[type="text"] {
text-align: center;
border: none;
outline: none;
padding: 8px;
border-radius: 4px;
box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.1);
font-size: 16px;
transition: box-shadow 0.3s;
width: 100%;
max-width: 120px; /* Ajustar el tamaño máximo de los inputs */
}

input[type="text"]:focus {
box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.3);
}

input[type="submit"] {
```

```

        background-color: #063d6d;
        color: white;
        border: none;
        padding: 10px 20px;
        border-radius: 4px;
        font-size: 17px;
        cursor: pointer;
        margin: 20px auto auto auto;
        transition: background-color 0.3s;
    }

input[type="submit"]:hover {
    background-color: #06345c;
}

.toggle {
    width: 40px;
    height: 40px;
    border-radius: 8px;
    background-color: #adadad;
    border: none;
    color: #fff;
    font-size: 20px;
    font-weight: bold;
    text-align: center;
    cursor: pointer;
    transition: background-color 0.3s;
}

.active {
    background-color: #007bff;
}

controltitle{
    font-size: 24px;
    color: #fff;
    font-weight: bold;
}

.home{
    padding: 0px 0px 10px 0px; background-color: #ffffff;
}

#header {
    position: relative;
}

#header .nav{
    box-sizing: border-box; display: inline-block;
    padding: 10px 0px 10px 0px; background-color:#063d6d;
    width: 100%; height: auto;
    position: relative;
}

#header .icon {
    position: absolute;
    top: 50%;
    transform: translateY(-50%);
    right: 15px;
    background-size: contain;
    background-repeat: no-repeat;
    width: 50px; height: 50px;
}

```

```

}

#header .iconspace {
  position: absolute;
  top: 50%;
  transform: translateY(-50%);
  right: 10px;
  background-size: contain;
  background-repeat: no-repeat;
  width: 60px; height: 60px;
  border-radius: 15px;
  background-color: white;
}

#header .slidernotifspace {
  position: absolute;
  top: 50%;
  transform: translateY(-50%);
  right: 170px;
  background-size: contain;
  background-repeat: no-repeat;
  width: 290px; height: 60px;
  border-radius: 15px;
  background-color: white;
}

.control{
  box-sizing: border-box; display: inline-block;
  padding: 0px 0px 15px 0px; background-color: #cccccc;
  border: 4px solid #063d6d; border-radius: 20px;
  width: auto; height: auto; margin: 30px;
}

.controltitlespace{
  display: inline-block;
  padding: 10px 0px; background-color: #063d6d;
  border-radius: 15px 15px 0 0;
  width: 100%; height: auto;
}

.myline {
  border: 2px solid #063d6d;
}

#t1{
  font-weight: bold;
  text-align: center; font-size: 50px; color: white;
}

#t2{
  font-weight: bold;
  text-align: center; font-size: 20px; color: white;
}

#t3{
  font-weight: bold;
  font-size: 20px; color: #063d6d;
}

#imag{
  width: 200px; height: 200px; padding: 25px;
}

#cmd{
  background-color: #063d6d;
  color: white;
  border: none;
  font-weight: bold; font-size: 20px;
}

```

```

padding: 10px 20px;
border-radius: 20px;
width: 150px; height: 50px;
cursor: pointer;
transition: background-color 0.3s;
}
#cmd:hover{
background-color: #06345c;
}

.switch {
position: relative;
display: inline-block;
width: 100px;
height: 50px;
margin: 0px 20px 0px 30px;
}

.switch input {
opacity: 0;
width: 0;
height: 31px;
}

.slider {
position: absolute;
cursor: pointer;
top: 0;
left: 0;
right: 0;
bottom: 0;
background-color: #adadad;
border-radius: 25px;
transition: 0.4s;
}

.slider:before {
position: absolute;
content: "";
height: 38px;
width: 38px;
left: 6px;
bottom: 6px;
background-color: white;
border-radius: 50%;
transition: 0.4s;
}

input:checked + .slider {
background-color: #063d6d;
}

input:checked + .slider:before {
transform: translateX(50px);
}

.slider.round {
border-radius: 34px;
}

.slider.round:before {

```

```

        border-radius: 50%;
    }
</style>
</head>
<body>
<div id="header">
    <div class="nav">
        <h1 id='t1'>RIEGO REMOTO</h1>
        <p id='t2'>Configura tu riego, tu plantita te lo
    agradecerá</p>
        <div class="slidernotifspace">
            <span id="t3" style="position: absolute; font-size: 19px;
top: 16px; left: 15px;">NOTIFICACIONES</span>
            <label class="switch" style="position: absolute; margin:
0px; right: 7px; top: 5px;">
                <input type="checkbox" id="notificationsEnable">
                <span class="slider"></span>
            </label>
        </div>
        <a href="/calibrarbomba">
            <div class="iconspace">
            </div>
            
            </a>
            <a href="/historicaldata">
                <div class="iconspace" style="right: 90px;">
                </div>
                
            </a>
        </div>
<script>
    document.addEventListener('DOMContentLoaded', function () {
        const toggleSwitch =
document.getElementById('notificationsEnable');

        toggleSwitch.addEventListener('change', function () {
            var xhr = new XMLHttpRequest();
            var timeout = false;
            var timeoutDuration = 3000; // 3 segundos
            var timeoutMessage = "No se ha recibido respuesta del
servidor. Por favor, inténtalo de nuevo.";

            if (this.checked) {
                xhr.open("POST", "/enableNotifications", true);
                xhr.send();
            } else {
                xhr.open("POST", "/disableNotifications", true);
                xhr.send();
            }

            // Establecer el tiempo de espera
            var timer = setTimeout(function () {
                timeout = true;
                xhr.abort(); // Abortar la solicitud
                alert(timeoutMessage);
            }, timeoutDuration);

            xhr.onreadystatechange = function () {

```



```

    };

    var params = "manualIrrigationVolume=" +
encodeURIComponent(document.getElementById("manualIrrigationVolume").value);

    xhr.send(params);
}
</script>
</div>
<div class="control">
    <span class="controltitlespace" style="margin-bottom:
15px;"><controltitle>RIEGO PROGRAMADO</controltitle></span><br />
    <label id='t3'>Volumen</label><br />
    <div class="input-container" style="margin-top: 3px;
margin-bottom: 15px;">
        <input type="text" name="scheduledIrrigationVolume"
id="scheduledIrrigationVolume" value="">
        <label id='t3' type="unit" style="right:
63px;">mL</label><br /><br />
    </div>
    <label id='t3'>Dias</label><br />

    <div style="margin-top: 10px; margin-bottom: 15px;">
        <button class="toggle"
onclick="toggleButton(this)">L</button>
        <button class="toggle"
onclick="toggleButton(this)">M</button>
        <button class="toggle"
onclick="toggleButton(this)">X</button>
        <button class="toggle"
onclick="toggleButton(this)">J</button>
        <button class="toggle"
onclick="toggleButton(this)">V</button>
        <button class="toggle"
onclick="toggleButton(this)">S</button>
        <button class="toggle"
onclick="toggleButton(this)">D</button>
    </div>

    <label id='t3'>Hora</label><br />
    <input type="text" style="max-width: 30px; margin-top:
10px; margin-bottom: 30px;" name="scheduledIrrigationHours"
id="scheduledIrrigationHours" value="">&nbsp;
    <label id='t3'>:</label>&nbsp;
    <input type="text" style="max-width: 30px;"
name="scheduledIrrigationMinutes" id="scheduledIrrigationMinutes"
value="">&nbsp;
    <label id='t3'>:</label>&nbsp;
    <input type="text" style="max-width: 30px;"
name="scheduledIrrigationSeconds" id="scheduledIrrigationSeconds"
value=""><br />

    <hr class="myline"><br /> &nbsp;
    <label class="switch">
        <input type="checkbox" id="scheduledIrrigationEnable">
        <span class="slider"></span>
    </label> &nbsp;
    <button id='cmd' type="button"
onclick="configScheduledIrrigation(event)">Programar</button> &nbsp;
</script>
function toggleButton(button) {

```

```

        if (button.classList.contains("active")) {
            button.classList.remove("active");
        } else {
            button.classList.add("active");
        }
    }
}
function configScheduledIrrigation(event) {
    event.preventDefault();
    var xhr = new XMLHttpRequest();
    var timeout = false;
    var timeoutDuration = 3000; // 3 segundos
    var timeoutMessage = "No se ha recibido respuesta
del servidor. Por favor, inténtalo de nuevo.";

    xhr.open("POST", "/configScheduledIrrigation",
true);

    xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");

    // Establecer el tiempo de espera
    var timer = setTimeout(function () {
        timeout = true;
        xhr.abort(); // Abortar la solicitud
        alert(timeoutMessage);
    }, timeoutDuration);

    xhr.onreadystatechange = function () {
        if (xhr.readyState === XMLHttpRequest.DONE) {
            clearTimeout(timer); // Limpiar el
temporizador si se recibe la respuesta
            if (!timeout && xhr.status === 200) {
                // Handle response from backend here
                var responseText = xhr.responseText;
                alert(responseText);
            }
        }
    };

    // Recorre los botones y verifica cuáles están
activos
    var params = "";
    var botones =
document.querySelectorAll(".toggle");
    params += "&L=" +
botones[0].classList.contains("active");
    params += "&M=" +
botones[1].classList.contains("active");
    params += "&X=" +
botones[2].classList.contains("active");
    params += "&J=" +
botones[3].classList.contains("active");
    params += "&V=" +
botones[4].classList.contains("active");
    params += "&S=" +
botones[5].classList.contains("active");
    params += "&D=" +
botones[6].classList.contains("active");
    params += "&hours=" +
encodeURIComponent(document.getElementById("scheduledIrrigationHours")
.value);
    params += "&minutes=" +

```

```

encodeURIComponent(document.getElementById("scheduledIrrigationMinutes")
.value);
        params += "&seconds=" +
encodeURIComponent(document.getElementById("scheduledIrrigationSeconds")
.value);
        params += "&scheduledIrrigationVolume=" +
encodeURIComponent(document.getElementById("scheduledIrrigationVolume")
.value);

        xhr.send(params);
    }
    document.addEventListener('DOMContentLoaded', function
() {
        const toggleSwitch =
document.getElementById('scheduledIrrigationEnable');

        toggleSwitch.addEventListener('change', function
() {
            var xhr = new XMLHttpRequest();
            var timeout = false;
            var timeoutDuration = 3000; // 3 segundos
            var timeoutMessage = "No se ha recibido
respuesta del servidor. Por favor, inténtalo de nuevo.";

            if (this.checked) {
                xhr.open("POST",
"/enableScheduledIrrigation", true);
                xhr.send();
            } else {
                xhr.open("POST",
"/disableScheduledIrrigation", true);
                xhr.send();
            }

            // Establecer el tiempo de espera
            var timer = setTimeout(function () {
                timeout = true;
                xhr.abort(); // Abortar la solicitud
                alert(timeoutMessage);
            }, timeoutDuration);

            xhr.onreadystatechange = function () {
                if (xhr.readyState ===
XMLHttpRequest.DONE) {
                    clearTimeout(timer); // Limpiar el
temporizador si se recibe la respuesta
                }
            };
        });
    });
    window.onload = function() {
        var xhr = new XMLHttpRequest();
        xhr.open("GET", "/getIrrigationConfig", true);
        xhr.onreadystatechange = function() {
            if (xhr.readyState === XMLHttpRequest.DONE &&
xhr.status === 200) {
                var response =
JSON.parse(xhr.responseText);

                // Accediendo a los elementos del JSON
                var scheduledIrrigationHour =

```

```

response.scheduledIrrigationHour;
        var scheduledIrrigationDays =
response.scheduledIrrigationDays;
        var scheduledIrrigationVolume =
response.scheduledIrrigationVolume;
        var scheduledIrrigationEnabled =
response.scheduledIrrigationEnabled;
        var automaticIrrigationVolume =
response.automaticIrrigationVolume;
        var automaticIrrigationMoistureLimit =
response.automaticIrrigationMoistureLimit;
        var automaticIrrigationEnabled =
response.automaticIrrigationEnabled;
        var notificationsEnabled =
response.notificationsEnabled;

// Procesando los elementos del array
scheduledIrrigationHour

document.getElementById("scheduledIrrigationHours").value =
scheduledIrrigationHour[0].toString().padStart(2, '0');

document.getElementById("scheduledIrrigationMinutes").value =
scheduledIrrigationHour[1].toString().padStart(2, '0');

document.getElementById("scheduledIrrigationSeconds").value =
scheduledIrrigationHour[2].toString().padStart(2, '0');

// Procesando los elementos del array
scheduledIrrigationDays
        var botones =
document.querySelectorAll(".toggle");
        for (var i = 0; i <
scheduledIrrigationDays.length; i++) {
            if (scheduledIrrigationDays[i]) {
botones[i].classList.add("active");
            }
        }

// Procesando los elementos de
scheduledIrrigationVolume

document.getElementById("scheduledIrrigationVolume").value =
scheduledIrrigationVolume;

// Procesando los elementos de riego
automatico

document.getElementById("automaticIrrigationVolume").value =
automaticIrrigationVolume;

document.getElementById("automaticIrrigationMoistureLimit").value =
automaticIrrigationMoistureLimit;

// Procesando los elementos de
automaticIrrigationEnabled
        if (automaticIrrigationEnabled) {

```

```

document.getElementById("automaticIrrigationEnable").checked = true;
    }

    // Procesando los elementos de
scheduledIrrigationEnabled
    if (scheduledIrrigationEnabled) {
document.getElementById("scheduledIrrigationEnable").checked = true;
    }

    // Procesando los elementos de
notificationsEnabled
    if (notificationsEnabled) {
document.getElementById("notificationsEnable").checked = true;
    }
    }
};
xhr.send();
};
document.addEventListener('DOMContentLoaded', function
() {
    const numberInput =
document.getElementById('manualIrrigationVolume');
    const minValue = 0;
    const maxValue = 1000;

    numberInput.addEventListener('keypress', function
(event) {
        const keyCode = event.which;
        if (keyCode < 48 || keyCode > 57) { // 48-57
son los códigos de tecla para los números del 0 al 9
            event.preventDefault();
        }
    });
    numberInput.addEventListener('keyup', function
(event) {
        const keyCode = event.which;
        if ((keyCode < 48 || keyCode > 57) && (keyCode
< 96 || keyCode > 105) && keyCode !== 8 && keyCode !== 13 && keyCode
!== 17 && keyCode !== 20 && keyCode !== 9) {
            this.setCustomValidity(`Sólo se aceptan
caracteres numéricos`);
        } else {
            const value = parseInt(this.value, 10);
            if (value < minValue || value > maxValue)
{
                this.setCustomValidity(`Por favor,
ingrese un número entre ${minValue} y ${maxValue}`);
            } else {
                this.setCustomValidity('');
            }
        }
        if (!this.checkValidity()) {
            this.reportValidity();
        }
    });
    numberInput.addEventListener('blur', function () {
        const value = parseInt(this.value, 10);
        if (value < minValue || value > maxValue) {
            this.setCustomValidity(`Por favor, ingrese

```

```

un número entre ${minValue} y ${maxValue}`);
        } else {
            this.setCustomValidity('');
            if (isNaN(value) || value < minValue ||
value > maxValue) {
                this.value = "0";
            }
        }
        if (!this.checkValidity()) {
            this.reportValidity();
        }
    });
});
document.addEventListener('DOMContentLoaded', function
() {
    const numberInput =
document.getElementById('scheduledIrrigationVolume');
    const minValue = 0;
    const maxValue = 1000;

    numberInput.addEventListener('keypress', function
(event) {
        const keyCode = event.which;
        if (keyCode < 48 || keyCode > 57) { // 48-57
son los códigos de tecla para los números del 0 al 9
            event.preventDefault();
        }
    });
    numberInput.addEventListener('keyup', function
(event) {
        const keyCode = event.which;
        if ((keyCode < 48 || keyCode > 57) && (keyCode
< 96 || keyCode > 105) && keyCode !== 8 && keyCode !== 13 && keyCode
!== 17 && keyCode !== 20 && keyCode !== 9) {
            this.setCustomValidity(`Sólo se aceptan
caracteres numéricos`);
        } else {
            const value = parseInt(this.value, 10);
            if (value < minValue || value > maxValue)
{
                this.setCustomValidity(`Por favor,
ingrese un número entre ${minValue} y ${maxValue}`);
            } else {
                this.setCustomValidity('');
            }
        }
        if (!this.checkValidity()) {
            this.reportValidity();
        }
    });
    numberInput.addEventListener('blur', function () {
        const value = parseInt(this.value, 10);
        if (value < minValue || value > maxValue) {
            this.setCustomValidity(`Por favor, ingrese
un número entre ${minValue} y ${maxValue}`);
        } else {
            this.setCustomValidity('');
            if (isNaN(value) || value < minValue ||
value > maxValue) {
                this.value = "0";
            }
        }
    });
});

```

```

        }
        if (!this.checkValidity()) {
            this.reportValidity();
        }
    });
});
document.addEventListener('DOMContentLoaded', function
() {
    const numberInput =
document.getElementById('scheduledIrrigationHours');
    const minValue = 0;
    const maxValue = 23;

    numberInput.addEventListener('keypress', function
(event) {
        const keyCode = event.which;
        if (keyCode < 48 || keyCode > 57) { // 48-57
son los códigos de tecla para los números del 0 al 9
            event.preventDefault();
        }
    });
    numberInput.addEventListener('keyup', function
(event) {
        const keyCode = event.which;
        if ((keyCode < 48 || keyCode > 57) && (keyCode
< 96 || keyCode > 105) && keyCode !== 8 && keyCode !== 13 && keyCode
!== 17 && keyCode !== 20 && keyCode !== 9) {
            this.setCustomValidity(`Sólo se aceptan
caracteres numéricos`);
        } else {
            const value = parseInt(this.value, 10);
            if (value < minValue || value > maxValue)
{
                this.setCustomValidity(`Por favor,
ingrese un número entre ${minValue} y ${maxValue}`);
            } else {
                this.setCustomValidity('');
            }
        }
        if (!this.checkValidity()) {
            this.reportValidity();
        }
    });
    numberInput.addEventListener('blur', function () {
        const value = parseInt(this.value, 10);
        if (value < minValue || value > maxValue) {
            this.setCustomValidity(`Por favor, ingrese
un número entre ${minValue} y ${maxValue}`);
        } else {
            this.setCustomValidity('');
            if (isNaN(value) || value < minValue ||
value > maxValue) {
                this.value = "00";
            } else {
                this.value =
value.toString().padStart(2, '0'); // Añade un cero a la izquierda si
hay solo un dígito
            }
        }
        if (!this.checkValidity()) {
            this.reportValidity();
        }
    });
});

```



```

    }
    });
});
document.addEventListener('DOMContentLoaded', function
() {
    const numberInput =
document.getElementById('scheduledIrrigationMinutes');
    const minValue = 0;
    const maxValue = 59;

    numberInput.addEventListener('keypress', function
(event) {
        const keyCode = event.which;
        if (keyCode < 48 || keyCode > 57) { // 48-57
son los códigos de tecla para los números del 0 al 9
            event.preventDefault();
        }
    });
    numberInput.addEventListener('keyup', function
(event) {
        const keyCode = event.which;
        if ((keyCode < 48 || keyCode > 57) && (keyCode
< 96 || keyCode > 105) && keyCode !== 8 && keyCode !== 13 && keyCode
!== 17 && keyCode !== 20 && keyCode !== 9) {
            this.setCustomValidity(`Sólo se aceptan
caracteres numéricos`);
        } else {
            const value = parseInt(this.value, 10);
            if (value < minValue || value > maxValue)
{
                this.setCustomValidity(`Por favor,
ingrese un número entre ${minValue} y ${maxValue}`);
            } else {
                this.setCustomValidity('');
            }
        }
        if (!this.checkValidity()) {
            this.reportValidity();
        }
    });
    numberInput.addEventListener('blur', function () {
        const value = parseInt(this.value, 10);
        if (value < minValue || value > maxValue) {
            this.setCustomValidity(`Por favor, ingrese
un número entre ${minValue} y ${maxValue}`);
        } else {
            this.setCustomValidity('');
            if (isNaN(value) || value < minValue ||
value > maxValue) {
                this.value = "00";
            } else {
                this.value =
value.toString().padStart(2, '0'); // Añade un cero a la izquierda si
hay solo un dígito
            }
        }
        if (!this.checkValidity()) {
            this.reportValidity();
        }
    });
});
});

```

```

        document.addEventListener('DOMContentLoaded', function
    () {
        const numberInput =
document.getElementById('scheduledIrrigationSeconds');
        const minValue = 0;
        const maxValue = 59;

        numberInput.addEventListener('keypress', function
    (event) {
            const keyCode = event.which;
            if (keyCode < 48 || keyCode > 57) { // 48-57
son los códigos de tecla para los números del 0 al 9
                event.preventDefault();
            }
        });
        numberInput.addEventListener('keyup', function
    (event) {
            const keyCode = event.which;
            if ((keyCode < 48 || keyCode > 57) && (keyCode
< 96 || keyCode > 105) && keyCode !== 8 && keyCode !== 13 && keyCode
!== 17 && keyCode !== 20 && keyCode !== 9) {
                this.setCustomValidity(`Sólo se aceptan
caracteres numéricos`);
            } else {
                const value = parseInt(this.value, 10);
                if (value < minValue || value > maxValue)
{
                    this.setCustomValidity(`Por favor,
ingrese un número entre ${minValue} y ${maxValue}`);
                } else {
                    this.setCustomValidity('');
                }
            }
            if (!this.checkValidity()) {
                this.reportValidity();
            }
        });
        numberInput.addEventListener('blur', function () {
            const value = parseInt(this.value, 10);
            if (value < minValue || value > maxValue) {
                this.setCustomValidity(`Por favor, ingrese
un número entre ${minValue} y ${maxValue}`);
            } else {
                this.setCustomValidity('');
                if (isNaN(value) || value < minValue ||
value > maxValue) {
                    this.value = "00";
                } else {
                    this.value =
value.toString().padStart(2, '0'); // Añade un cero a la izquierda si
hay solo un dígito
                }
            }
            if (!this.checkValidity()) {
                this.reportValidity();
            }
        });
    });
</script>
</div>
<div class="control">

```

```

        <span class="controltitlespace" style="margin-bottom:
15px;"><controltitle>RIEGO AUTOMÁTICO</controltitle></span><br />
        <label id='t3'>Volumen</label><br />
        <div class="input-container" style="margin-top: 3px;
margin-bottom: 15px;">
            <input type="text" name="automaticIrrigationVolume"
id="automaticIrrigationVolume" value="">
            <label id='t3' type="unit" style="right: 8px;">±20%
mL</label><br /><br />
        </div>
        <label id='t3'>Límite de humedad en suelo</label><br />
        <div class="input-container" style="margin-top: 5px;
margin-bottom: 30px;">
            <input type="text"
name="automaticIrrigationMoistureLimit"
id="automaticIrrigationMoistureLimit" value="">
            <label id='t3' type="unit" style="right:
75px;">%</label><br /><br />
        </div>

        <hr class="myline"><br />
        <label id='t3'>Temperatura:</label>&nbsp;
        <span id="temperature" style="font-weight: bold; font-
size: 20px;">0 °C</span><br />
        <label id='t3'>Humedad ambiente:</label>&nbsp;
        <span id="humidity" style="font-weight: bold; font-size:
20px;">0 %</span><br />
        <label id='t3'>Humedad en suelo:</label>&nbsp;
        <span id="soilMoisture" style="font-weight: bold; font-
size: 20px;">0 %</span><br />
        <br />
        <hr class="myline"><br /> &nbsp;
        <label class="switch">
            <input type="checkbox" id="automaticIrrigationEnable">
            <span class="slider"></span>
        </label> &nbsp;
        <button id='cmd' type="button"
onclick="configAutomaticIrrigation(event)">Configurar</button> &nbsp;
        <script>
            function updateSensorData() {
                var xhr = new XMLHttpRequest();
                xhr.open("GET", "/getSensorData", true);
                xhr.onreadystatechange = function() {
                    if (xhr.readyState === XMLHttpRequest.DONE &&
xhr.status === 200) {
                        var response =
JSON.parse(xhr.responseText);

                        // Accediendo a los elementos del JSON
                        var temperature =
parseFloat(response.temperature).toFixed(1);
                        var humidity =
parseFloat(response.humidity).toFixed(0);
                        var soilMoisture = response.soilMoisture;

                        document.getElementById("temperature").textContent = temperature + "
°C";

                        document.getElementById("humidity").textContent = humidity + " %";

```

```

document.getElementById("soilMoisture").textContent = soilMoisture + "
%";
    }
    };
    xhr.send();
}
setInterval(() => {
    // Aquí puedes obtener el valor de los sensores de
una fuente de datos, como una API
    updateSensorData();
}, 1000);
function configAutomaticIrrigation(event) {
    event.preventDefault();
    var xhr = new XMLHttpRequest();
    var timeout = false;
    var timeoutDuration = 3000; // 3 segundos
    var timeoutMessage = "No se ha recibido respuesta
del servidor. Por favor, inténtalo de nuevo.";

    xhr.open("POST", "/configAutomaticIrrigation",
true);
    xhr.setRequestHeader("Content-type",
"application/x-www-form-urlencoded");

    // Establecer el tiempo de espera
    var timer = setTimeout(function () {
        timeout = true;
        xhr.abort(); // Abortar la solicitud
        alert(timeoutMessage);
    }, timeoutDuration);

    xhr.onreadystatechange = function () {
        if (xhr.readyState === XMLHttpRequest.DONE) {
            clearTimeout(timer); // Limpiar el
temporizador si se recibe la respuesta
            if (!timeout && xhr.status === 200) {
                // Handle response from backend here
                var responseText = xhr.responseText;
                alert(responseText);
            }
        }
    };

    // Recorre los botones y verifica cuáles están
activos
    var params = "";
    params += "&automaticIrrigationVolume=" +
encodeURIComponent(document.getElementById("automaticIrrigationVolume"
).value);
    params += "&automaticIrrigationMoistureLimit=" +
encodeURIComponent(document.getElementById("automaticIrrigationMoistur
eLimit").value);

    xhr.send(params);
}
document.addEventListener('DOMContentLoaded', function
() {
    const toggleSwitch =
document.getElementById('automaticIrrigationEnable');

```

```

toggleSwitch.addEventListener('change', function
() {
    var xhr = new XMLHttpRequest();
    var timeout = false;
    var timeoutDuration = 3000; // 3 segundos
    var timeoutMessage = "No se ha recibido
respuesta del servidor. Por favor, inténtalo de nuevo.";

    if (this.checked) {
        xhr.open("POST",
"/enableAutomaticIrrigation", true);
        xhr.send();
    } else {
        xhr.open("POST",
"/disableAutomaticIrrigation", true);
        xhr.send();
    }

    // Establecer el tiempo de espera
    var timer = setTimeout(function () {
        timeout = true;
        xhr.abort(); // Abortar la solicitud
        alert(timeoutMessage);
    }, timeoutDuration);

    xhr.onreadystatechange = function () {
        if (xhr.readyState ===
XMLHttpRequest.DONE) {
            clearTimeout(timer); // Limpiar el
temporizador si se recibe la respuesta
        }
    };
});
document.addEventListener('DOMContentLoaded', function
() {
    const numberInput =
document.getElementById('automaticIrrigationVolume');
    const minValue = 0;
    const maxValue = 1000;

    numberInput.addEventListener('keypress', function
(event) {
        const keyCode = event.which;
        if (keyCode < 48 || keyCode > 57) { // 48-57
son los códigos de tecla para los números del 0 al 9
            event.preventDefault();
        }
    });
    numberInput.addEventListener('keyup', function
(event) {
        const keyCode = event.which;
        if ((keyCode < 48 || keyCode > 57) && (keyCode
< 96 || keyCode > 105) && keyCode !== 8 && keyCode !== 13 && keyCode
!== 17 && keyCode !== 20 && keyCode !== 9) {
            this.setCustomValidity(`Sólo se aceptan
caracteres numéricos`);
        } else {
            const value = parseInt(this.value, 10);
            if (value < minValue || value > maxValue)
{

```

```

        this.setCustomValidity(`Por favor,
ingrese un número entre ${minValue} y ${maxValue}`);
    } else {
        this.setCustomValidity('');
    }
}
if (!this.checkValidity()) {
    this.reportValidity();
}
});
numberInput.addEventListener('blur', function () {
    const value = parseInt(this.value, 10);
    if (value < minValue || value > maxValue) {
        this.setCustomValidity(`Por favor, ingrese
un número entre ${minValue} y ${maxValue}`);
    } else {
        this.setCustomValidity('');
        if (isNaN(value) || value < minValue ||
value > maxValue) {
            this.value = "0";
        }
    }
    if (!this.checkValidity()) {
        this.reportValidity();
    }
});
});
document.addEventListener('DOMContentLoaded', function
() {
    const numberInput =
document.getElementById('automaticIrrigationMoistureLimit');
    const minValue = 0;
    const maxValue = 100;

    numberInput.addEventListener('keypress', function
(event) {
        const keyCode = event.which;
        if (keyCode < 48 || keyCode > 57) { // 48-57
son los códigos de tecla para los números del 0 al 9
            event.preventDefault();
        }
    });
    numberInput.addEventListener('keyup', function
(event) {
        const keyCode = event.which;
        if ((keyCode < 48 || keyCode > 57) && (keyCode
< 96 || keyCode > 105) && keyCode !== 8 && keyCode !== 13 && keyCode
!== 17 && keyCode !== 20 && keyCode !== 9) {
            this.setCustomValidity(`Sólo se aceptan
caracteres numéricos`);
        } else {
            const value = parseInt(this.value, 10);
            if (value < minValue || value > maxValue)
{
                this.setCustomValidity(`Por favor,
ingrese un número entre ${minValue} y ${maxValue}`);
            } else {
                this.setCustomValidity('');
            }
        }
        if (!this.checkValidity()) {

```

```

        this.reportValidity();
    }
});
numberInput.addEventListener('blur', function () {
    const value = parseInt(this.value, 10);
    if (value < minValue || value > maxValue) {
        this.setCustomValidity(`Por favor, ingrese
un número entre ${minValue} y ${maxValue}`);
    } else {
        this.setCustomValidity('');
        if (isNaN(value) || value < minValue ||
value > maxValue) {
            this.value = "0";
        }
    }
    if (!this.checkValidity()) {
        this.reportValidity();
    }
});
});
</script>
</div>
</center>
</div>
</body>
</html>

```

7.1.1.2 *calibrarbomba.html*

```

<!DOCTYPE html>
<html>
<head>
    <title>RIEGO REMOTO</title>
    <link rel="shortcut icon" href="https://encrypted-
tbn0.gstatic.com/images?q=tbn:ANd9GcTxYG8RqjMR8bdewCu6ZVm-
MhRCWtftogDd2AJluw0LBM5NWBtcOhaAAamAlgsjLSF_p4g&usqp=CAU">
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <style type="text/css">
        *{
padding: 0px; margin: 0px;
}

body {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}

form {
    display: flex;
    flex-direction: column;
    max-width: 300px;
    margin: auto;
}

.input-container {
    display: flex;
    justify-content: center;
    align-items: center;
    margin-bottom: 10px;
    position: relative;
}

```

```

label[type="unit"] {
    position: absolute;
}

input[type="text"] {
    text-align: center;
    border: none;
    outline: none;
    padding: 8px;
    border-radius: 4px;
    box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.1);
    font-size: 16px;
    transition: box-shadow 0.3s;
    width: 100%;
    max-width: 150px; /* Ajustar el tamaño máximo de los inputs */
}

input[type="text"]:focus {
    box-shadow: inset 0 1px 3px rgba(0, 0, 0, 0.3);
}

input[type="submit"] {
    background-color: #063d6d;
    color: white;
    border: none;
    padding: 10px 20px;
    border-radius: 4px;
    font-size: 17px;
    cursor: pointer;
    margin: auto;
    transition: background-color 0.3s;
}

input[type="submit"]:hover {
    background-color: #06345c;
}

controltitle{
    font-size: 24px;
    color: #fff;
    font-weight: bold;
}

.home{
    padding: 0px 0px 10px 0px; background-color: #ffffff;
}

#header {
    position: relative;
}

#header .nav{
    box-sizing: border-box; display: inline-block;
    padding: 10px 0px 10px 0px; background-color:#063d6d;
    width: 100%; height: auto;
    position: relative;
}

#header .icon {
    position: absolute;
    top: 50%;
    transform: translateY(-50%);
}

```



```

    left: 15px;
    background-size: contain;
    background-repeat: no-repeat;
    width: 50px; height: 50px;
}

#header .iconspace {
    position: absolute;
    top: 50%;
    transform: translateY(-50%);
    left: 10px;
    background-size: contain;
    background-repeat: no-repeat;
    width: 60px; height: 60px;
    border-radius: 15px;
    background-color: white;
}

.control{
    box-sizing: border-box; display: inline-block;
    padding: 0px 0px 15px 0px; background-color: #cccccc;
    border: 4px solid #063d6d; border-radius: 20px;
    width: auto; height: auto; margin: 30px;
}

.controltitlespace{
    display: inline-block;
    padding: 10px 0px; background-color: #063d6d;
    border-radius: 15px 15px 0 0;
    width: 100%; height: auto;
}

.myline {
    border: 2px solid #063d6d;
}

#t1{
    font-weight: bold;
    text-align: center; font-size: 50px; color: white;
}

#t2{
    font-weight: bold;
    text-align: center; font-size: 20px; color: white;
}

#t3{
    font-weight: bold;
    font-size: 20px; color: #063d6d;
}

#imag{
    width: 200px; height: 200px; padding: 25px;
}

#cmd{
    background-color: #063d6d;
    color: white;
    border: none;
    font-weight: bold; font-size: 20px;
    padding: 10px 20px;
    border-radius: 20px;
    width: 150px; height: 50px;
    cursor: pointer;
    transition: background-color 0.3s;
}

#cmd:hover{
    background-color: #06345c;
}

```

```

    }
    </style>
</head>
<body>
<div id="header">
    <div class="nav">
        <h1 id='t1'>CALIBRADO DE BOMBA</h1>
        <p id='t2'>Desde esta página podrás calibrar el caudal de tu
bomba</p>
        <a href="/">
            <div class="iconspace">
                </div>
                
            </a>
        </div>
    </div>
<div class="home">
    <center>
        <div class="control">
            <span class="controltitlespace" style="margin-bottom:
15px;"><controltitle>CALIBRADO</controltitle></span>
            <form style="margin-bottom: 20px;" method="post"
action="/updateCalibration">
                <label id='t3' for="calibrationVolume">Volumen</label>
                <div class="input-container" style="margin-top: 7px;
margin-bottom: 15px;">
                    <input type="text" name="calibrationVolume"
id="calibrationVolume" value="">
                    <label id='t3' type="unit" style="right: 27px;
for="calibrationVolume">mL</label>
                </div>
                <label id='t3' for="calibrationTime">Tiempo</label>
                <div class="input-container" style="margin-top: 7px;
margin-bottom: 35px;">
                    <input type="text" name="calibrationTime"
id="calibrationTime" value="">
                    <label id='t3' type="unit" style="right: 45px;
for="calibrationTime">s</label>
                </div>
                <input type="submit" value="Guardar calibrado"
onclick="sendCalibration(event)">
            </form>
            <script>
                function sendCalibration(event) {
                    event.preventDefault();
                    var xhr = new XMLHttpRequest();
                    var timeout = false;
                    var timeoutDuration = 3000; // 3 segundos
                    var timeoutMessage = "No se ha recibido respuesta
del servidor. Por favor, inténtalo de nuevo.";

                    xhr.open("POST", "/updateCalibration", true);
                    xhr.setRequestHeader("Content-Type",
"application/x-www-form-urlencoded");

                    // Establecer el tiempo de espera
                    var timer = setTimeout(function () {
                        timeout = true;
                        xhr.abort(); // Abortar la solicitud
                        alert(timeoutMessage);
                    }, timeoutDuration);
                }
            </script>
        </div>
    </center>
</div>

```

```

    }, timeoutDuration);

    xhr.onreadystatechange = function () {
        if (xhr.readyState === XMLHttpRequest.DONE) {
            clearTimeout(timer); // Limpiar el
temporizador si se recibe la respuesta
            if (!timeout && xhr.status === 200) {
                // Handle response from backend here
                var responseText = xhr.responseText;
                alert(responseText);
            }
        }
    };

    var params = "calibrationVolume=" +
encodeURIComponent(document.getElementById("calibrationVolume").value)
+
"&calibrationTime=" +
encodeURIComponent(document.getElementById("calibrationTime").value);
    xhr.send(params);
}
window.onload = function() {
    var xhr = new XMLHttpRequest();
    xhr.open("GET", "/getCalibration", true);
    xhr.onreadystatechange = function() {
        if (xhr.readyState === XMLHttpRequest.DONE &&
xhr.status === 200) {
            var response =
JSON.parse(xhr.responseText);

document.getElementById("calibrationVolume").value =
response.calibrationVolume;

document.getElementById("calibrationTime").value =
response.calibrationTime;
        }
    };
    xhr.send();
};
</script>
<hr class="myline"><br /> &nbsp;
<button id='cmd' type="button"
onclick="calibration('1')">START</button> &nbsp;
<button id='cmd' type="button"
onclick="calibration('0')">STOP</button> &nbsp;
<script>
function calibration(value) {
    var xhr = new XMLHttpRequest();
    var timeout = false;
    var timeoutDuration = 3000; // 3 segundos
    var timeoutMessage = "No se ha recibido respuesta
del servidor. Por favor, inténtalo de nuevo.";

    // Establecer el tiempo de espera
    var timer = setTimeout(function () {
        timeout = true;
        xhr.abort(); // Abortar la solicitud
        alert(timeoutMessage);
    }, timeoutDuration);

    if (value == 1) {

```

```

        xhr.open("GET", "/startCalibration", true);
        xhr.onreadystatechange = function () {
            if (xhr.readyState ===
XMLHttpRequest.DONE) {
                clearTimeout(timer); // Limpiar el
temporizador si se recibe la respuesta
            }
        };
        xhr.send();
    }
    if (value == 0) {
        xhr.open("GET", "/stopCalibration", true);
        xhr.onreadystatechange = function () {
            if (xhr.readyState ===
XMLHttpRequest.DONE) {
                clearTimeout(timer); // Limpiar el
temporizador si se recibe la respuesta
                if (!timeout && xhr.status === 200) {
                    var response =
JSON.parse(xhr.responseText);

document.getElementById("calibrationTime").value =
response.calibrationTime;
                }
            }
        };
        xhr.send();
    }
}
document.addEventListener('DOMContentLoaded', function
() {
    const numberInput =
document.getElementById('calibrationVolume');
    const minValue = 0;
    const maxValue = 2000;

    numberInput.addEventListener('keypress', function
(event) {
        const keyCode = event.which;
        if (keyCode < 48 || keyCode > 57) { // 48-57
son los códigos de tecla para los números del 0 al 9
            event.preventDefault();
        }
    });
    numberInput.addEventListener('keyup', function
(event) {
        const keyCode = event.which;
        if ((keyCode < 48 || keyCode > 57) && (keyCode
< 96 || keyCode > 105) && keyCode !== 8 && keyCode !== 13 && keyCode
!== 17 && keyCode !== 20 && keyCode !== 9) {
            this.setCustomValidity(`Sólo se aceptan
caracteres numéricos`);
        } else {
            const value = parseInt(this.value, 10);
            if (value < minValue || value > maxValue)
{
                this.setCustomValidity(`Por favor,
ingrese un número entre ${minValue} y ${maxValue}`);
            } else {
                this.setCustomValidity('');
            }
        }
    }
}

```

```

        }
        if (!this.checkValidity()) {
            this.reportValidity();
        }
    });
    numberInput.addEventListener('blur', function () {
        const value = parseInt(this.value, 10);
        if (value < minValue || value > maxValue) {
            this.setCustomValidity(`Por favor, ingrese
un número entre ${minValue} y ${maxValue}`);
        } else {
            this.setCustomValidity('');
            if (isNaN(value) || value < minValue ||
value > maxValue) {
                this.value = "0";
            }
        }
        if (!this.checkValidity()) {
            this.reportValidity();
        }
    });
});
document.addEventListener('DOMContentLoaded', function
() {
    const numberInput =
document.getElementById('calibrationTime');
    const minValue = 0;
    const maxValue = 1000;

    numberInput.addEventListener('keypress', function
(event) {
        const keyCode = event.which;
        if (keyCode < 48 || keyCode > 57) { // 48-57
son los códigos de tecla para los números del 0 al 9
            event.preventDefault();
        }
    });
    numberInput.addEventListener('keyup', function
(event) {
        const keyCode = event.which;
        if ((keyCode < 48 || keyCode > 57) && (keyCode
< 96 || keyCode > 105) && keyCode !== 8 && keyCode !== 13 && keyCode
!== 17 && keyCode !== 20 && keyCode !== 9) {
            this.setCustomValidity(`Sólo se aceptan
caracteres numéricos`);
        } else {
            const value = parseInt(this.value, 10);
            if (value < minValue || value > maxValue)
{
                this.setCustomValidity(`Por favor,
ingrese un número entre ${minValue} y ${maxValue}`);
            } else {
                this.setCustomValidity('');
            }
        }
        if (!this.checkValidity()) {
            this.reportValidity();
        }
    });
    numberInput.addEventListener('blur', function () {
        const value = parseInt(this.value, 10);

```

```

        if (value < minValue || value > maxValue) {
            this.setCustomValidity(`Por favor, ingrese
un número entre ${minValue} y ${maxValue}`);
        } else {
            this.setCustomValidity('');
            if (isNaN(value) || value < minValue ||
value > maxValue) {
                this.value = "0";
            }
        }
        if (!this.checkValidity()) {
            this.reportValidity();
        }
    });
});
</script>
</div>
</center>
</div>
</body>
</html>

```

7.1.2 Back-end

7.1.2.1 RiegoRemoto.ino

```

#include <Arduino.h>
#include <ESP8266WebServer.h>
#include <LittleFS.h>
#include <ArduinoJson.h>
#include <Ticker.h>
#include <Wire.h>
#include <EmailSender.h>

#include "secrets.h"
#include "HDC2010.h"
#include "config.h"
#include "mail.h"
#include "tankLevel.h"
#include "irrigation.h"
#include "schedule.h"
#include "moisture.h"
#include "automatic.h"
#include "handleServer.h"

#define UNUSED __attribute__((unused))

// TRACE output simplified, can be deactivated here
#define TRACE(...) Serial.printf(__VA_ARGS__)

// Definition of an static IP address
IPAddress ip(192, 168, 1, 200); // Static IP address assigned
IPAddress gateway(192, 168, 1, 1); // IP address of the gateway
IPAddress subnet(255, 255, 255, 0); // Subnet mask

// Name of the server

```

```

#define HOSTNAME "riegoremoto"

// The text of builtin files are in this header file
#include "builtinfiles.h"

// Definition of the default configuration
const char* CALIBRATIONVOLUME_DEFAULT = "100";
const char* CALIBRATIONTIME_DEFAULT = "5";
const int SCHEDULEDIRRIGATIONHOUR_DEFAULT[3] = {12, 0, 0};
const bool SCHEDULEDIRRIGATIONDAYS_DEFAULT[7] = {1, 1, 1, 1, 1, 1, 1};
const char* SCHEDULEDCALIBRATIONVOLUME_DEFAULT = "100";
const bool SCHEDULEDIRRIGATIONENABLED_DEFAULT = 0;
const char* AUTOMATICIRRIGATIONVOLUME_DEFAULT = "100";
const char* AUTOMATICIRRIGATIONMOISTURELIMIT_DEFAULT = "20";
const bool AUTOMATICIRRIGATIONENABLED_DEFAULT = 0;
const bool NOTIFICATIONSENABLED_DEFAULT = 0;

// Setup everything to make the webserver work
void setup(void) {
  // Initialize digital pins
  pinMode(D0, OUTPUT);
  pinMode(D6, OUTPUT);
  pinMode(D5, INPUT);
  Wire.begin(); // Starts the I2C communication

  delay(3000); // Wait for serial monitor to start completely

  // Use Serial port for some trace information
  Serial.begin(115200);
  Serial.setDebugOutput(false);

  TRACE("Starting WebServer...\n");

  // Start File system
  if (!LittleFS.begin()) {
    Serial.println("No se pudo montar el sistema de archivos LittleFS.");
    delay(2000);
    ESP.restart();
  }

  // Load config from config file checking existence of a config file and
  // creating it with default values in case of non existence
  if (!config.fileExist()) {
    config.setDefaultConfig(CALIBRATIONVOLUME_DEFAULT,
CALIBRATIONTIME_DEFAULT, SCHEDULEDIRRIGATIONHOUR_DEFAULT,
SCHEDULEDIRRIGATIONDAYS_DEFAULT, SCHEDULEDCALIBRATIONVOLUME_DEFAULT,
SCHEDULEDIRRIGATIONENABLED_DEFAULT, AUTOMATICIRRIGATIONVOLUME_DEFAULT,

```

```

AUTOMATICIRRIGATIONMOISTURELIMIT_DEFAULT,
AUTOMATICIRRIGATIONENABLED_DEFAULT, NOTIFICATIONSENABLED_DEFAULT);
    config.save();
} else if (!config.load()) {
    config.setDefaultConfig(CALIBRATIONVOLUME_DEFAULT,
CALIBRATIONTIME_DEFAULT, SCHEDULEDIRRIGATIONHOUR_DEFAULT,
SCHEDULEDIRRIGATIONDAYS_DEFAULT, SCHEDULEDCALIBRATIONVOLUME_DEFAULT,
SCHEDULEDIRRIGATIONENABLED_DEFAULT, AUTOMATICIRRIGATIONVOLUME_DEFAULT,
AUTOMATICIRRIGATIONMOISTURELIMIT_DEFAULT,
AUTOMATICIRRIGATIONENABLED_DEFAULT, NOTIFICATIONSENABLED_DEFAULT);
    config.save();
}

// start WiFi
WiFi.mode(WIFI_STA);
if (strlen(ssid) == 0) {
    WiFi.begin();
} else {
    WiFi.begin(ssid, passphrase);
}

// Allow to address the device by the given name
WiFi.setHostname(HOSTNAME);

TRACE("Connect to WiFi...\n");
while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    TRACE(".");
}
WiFi.config(ip, gateway, subnet);
TRACE("connected.\n");

// Start NTP client
TRACE("Setup ntp...\n");
timeClient.begin();
timeClient.update();
setTime(timeClient.getEpochTime());

TRACE("Register service handlers...\n");

// Serve the built-in htm pages
server.on("/", []() {
    server.setHeader("Content-Encoding", "gzip");
    server.send(200, "text/html", mainpage, mainpageSize);
});
server.on("/calibrarbomba", []() {
    server.setHeader("Content-Encoding", "gzip");
    server.send(200, "text/html", calibrarbomba, calibrarbombaSize);
});

```



```

});
server.on("/historicaldata", []() {
  server.send(200, "text/html", FPSTR(historicaldata));
});

//Functions
server.on("/startManualIrrigation", handleStartManualIrrigation);
server.on("/configScheduledIrrigation",
handleConfigScheduledIrrigation);
server.on("/getIrrigationConfig", handleGetIrrigationConfig);
server.on("/enableScheduledIrrigation",
handleEnableScheduledIrrigation);
server.on("/disableScheduledIrrigation",
handleDisableScheduledIrrigation);

server.on("/configAutomaticIrrigation",
handleConfigAutomaticIrrigation);
server.on("/getSensorData", handleGetSensorData);
server.on("/enableAutomaticIrrigation",
handleEnableAutomaticIrrigation);
server.on("/disableAutomaticIrrigation",
handleDisableAutomaticIrrigation);

server.on("/enableNotifications", handleEnableNotifications);
server.on("/disableNotifications", handleDisableNotifications);

server.on("/startCalibration", handleStartCalibration);
server.on("/stopCalibration", handleStopCalibration);
server.on("/updateCalibration", handleUpdateCalibration);
server.on("/getCalibration", handleGetCalibration);

// Enable CORS header in webserver results
server.enableCORS(true);

// Enable ETAG header in webserver results from serveStatic handler
server.enableETag(true);

// Handle not found
server.onNotFound([]() {
  server.send(404, "text/plain", "Error 404 page not found");
});

server.begin();
TRACE("hostname=%s\n", WiFi.getHostname());

if (config.isScheduledIrrigationEnabled()) {
  updateScheduledIrrigationAlarm();
} else {
  removeScheduledIrrigationAlarm();
}

```

```

    }
} // setup

// Run the server...
void loop(void) {
    server.handleClient();

    processTankLevel();

    sendAutomaticIrrigationReportingMailIfNeeded();

    if (needAutomaticIrrigation()) {
        automaticIrrigation();
    }

    if (lowTankLevel) {
        finishIrrigation();
    }

    Alarm.delay(10); // Calls the programmed alarms and waits for 10
    milliseconds
} // loop()

// end.

```

7.1.2.2 *config.h*

```

//Class used for the configuration
class Configuration {
public:
    // Constructor
    Configuration(const char* filename) {
        filename_ = filename;
    }

    // Checks if the config file exists
    bool fileExist() {
        if (LittleFS.exists(filename_)) {
            return true;
        } else {
            return false;
        }
    }
}

// Loads the default config values

```

```

    void setDefaultConfig(const String& default_calibrationVolume, const
String& default_calibrationTime, const int
default_scheduledIrrigationHour[3], const bool
default_scheduledIrrigationDays[7], const String&
default_scheduledIrrigationVolume, const bool
default_scheduledIrrigationEnabled, const String&
default_automaticIrrigationVolume, const String&
default_automaticIrrigationMoistureLimit, const bool
default_automaticIrrigationEnabled, const bool
default_notificationsEnabled) {
    calibrationVolume_ = default_calibrationVolume;
    calibrationTime_ = default_calibrationTime;
    for (int i = 0; i < 3; i++) {
        scheduledIrrigationHour_[i] = default_scheduledIrrigationHour[i];
    }
    for (int i = 0; i < 7; i++) {
        scheduledIrrigationDays_[i] = default_scheduledIrrigationDays[i];
    }
    scheduledIrrigationVolume_ = default_scheduledIrrigationVolume;
    scheduledIrrigationEnabled_ = default_scheduledIrrigationEnabled;
    automaticIrrigationVolume_ = default_automaticIrrigationVolume;
    automaticIrrigationMoistureLimit_ =
default_automaticIrrigationMoistureLimit;
    automaticIrrigationEnabled_ = default_automaticIrrigationEnabled;
    notificationsEnabled_ = default_notificationsEnabled;
}

// Reads and loads the config from the config file
bool load() {
    File configFile = LittleFS.open(filename_, "r");
    if (!configFile) {
        Serial.println("Error al abrir el archivo de configuración.");
        return false;
    }

    size_t size = configFile.size();
    if (size == 0) {
        Serial.println("El archivo de configuración está vacío.");
        return false;
    }

    std::unique_ptr<char[]> buf(new char[size]);
    configFile.readBytes(buf.get(), size);
    configFile.close();

    StaticJsonDocument<1000> doc;
    auto error = deserializeJson(doc, buf.get());
    if (error) {

```

```

        Serial.println("Error al analizar el archivo de configuración.");
        return false;
    }

    calibrationVolume_ = doc["calibrationVolume"].as<String>();
    calibrationTime_ = doc["calibrationTime"].as<String>();

    for (int i = 0; i < 3; i++) {
        scheduledIrrigationHour_[i] =
doc["scheduledIrrigationHour"].as<JSONArray>()[i].as<int>();
    }
    for (int i = 0; i < 7; i++) {
        scheduledIrrigationDays_[i] =
doc["scheduledIrrigationDays"].as<JSONArray>()[i].as<bool>();
    }
    scheduledIrrigationVolume_ =
doc["scheduledIrrigationVolume"].as<String>();
    scheduledIrrigationEnabled_ =
doc["scheduledIrrigationEnabled"].as<bool>();

    automaticIrrigationVolume_ =
doc["automaticIrrigationVolume"].as<String>();
    automaticIrrigationMoistureLimit_ =
doc["automaticIrrigationMoistureLimit"].as<String>();
    automaticIrrigationEnabled_ =
doc["automaticIrrigationEnabled"].as<bool>();
    notificationsEnabled_ = doc["notificationsEnabled"].as<bool>();

    return true;
}

// Writes the config to the config file
bool save() {
    StaticJsonDocument<1000> doc;

    doc["calibrationVolume"] = calibrationVolume_;
    doc["calibrationTime"] = calibrationTime_;

    JSONArray scheduledIrrigationHourArray =
doc.createNestedArray("scheduledIrrigationHour");
    for (int i = 0; i < 3; i++) {
        scheduledIrrigationHourArray.add(scheduledIrrigationHour_[i]);
    }

    JSONArray scheduledIrrigationDaysArray =
doc.createNestedArray("scheduledIrrigationDays");
    for (int i = 0; i < 7; i++) {
        scheduledIrrigationDaysArray.add(scheduledIrrigationDays_[i]);
    }
}

```

```

doc["scheduledIrrigationVolume"] = scheduledIrrigationVolume_;
doc["scheduledIrrigationEnabled"] = scheduledIrrigationEnabled_;

doc["automaticIrrigationVolume"] = automaticIrrigationVolume_;
doc["automaticIrrigationMoistureLimit"] =
automaticIrrigationMoistureLimit_;
doc["automaticIrrigationEnabled"] = automaticIrrigationEnabled_;
doc["notificationsEnabled"] = notificationsEnabled_;

File configFile = LittleFS.open(filename_, "w");
if (!configFile) {
    Serial.println("Error al abrir el archivo de configuración para
escritura.");
    return false;
}

serializeJson(doc, configFile);
configFile.close();

return true;
}

// Functions to set the config values
void setCalibrationVolume(const String& calibrationVolume) {
    calibrationVolume_ = calibrationVolume;
}

void setCalibrationTime(const String& calibrationTime) {
    calibrationTime_ = calibrationTime;
}

void setScheduledIrrigationHour(int scheduledIrrigationHour[3]) {
    for (int i = 0; i < 3; i++) {
        scheduledIrrigationHour_[i] = scheduledIrrigationHour[i];
    }
}

void setScheduledIrrigationDays(bool scheduledIrrigationDays[7]) {
    for (int i = 0; i < 7; i++) {
        scheduledIrrigationDays_[i] = scheduledIrrigationDays[i];
    }
}

void setScheduledIrrigationVolume(const String&
scheduledIrrigationVolume) {
    scheduledIrrigationVolume_ = scheduledIrrigationVolume;
}

```

```

void enableScheduledIrrigation() {
    scheduledIrrigationEnabled_ = true;
}

void disableScheduledIrrigation() {
    scheduledIrrigationEnabled_ = false;
}

void setAutomaticIrrigationVolume(const String&
automaticIrrigationVolume) {
    automaticIrrigationVolume_ = automaticIrrigationVolume;
}

void setAutomaticIrrigationMoistureLimit(const String&
automaticIrrigationMoistureLimit) {
    automaticIrrigationMoistureLimit_ =
automaticIrrigationMoistureLimit;
}

void enableAutomaticIrrigation() {
    automaticIrrigationEnabled_ = true;
}

void disableAutomaticIrrigation() {
    automaticIrrigationEnabled_ = false;
}

void enableNotifications() {
    notificationsEnabled_ = true;
}

void disableNotifications() {
    notificationsEnabled_ = false;
}

// Functions to return the config values
String getCalibrationVolume() const {
    return calibrationVolume_;
}

String getCalibrationTime() const {
    return calibrationTime_;
}

const int (&getScheduledIrrigationHour() const)[3] {
    return scheduledIrrigationHour_;
}

const bool (&getScheduledIrrigationDays() const)[7] {

```

```

    return scheduledIrrigationDays_;
}

String getScheduledIrrigationVolume() const {
    return scheduledIrrigationVolume_;
}

bool isScheduledIrrigationEnabled() const {
    return scheduledIrrigationEnabled_;
}

String getAutomaticIrrigationVolume() const {
    return automaticIrrigationVolume_;
}

String getAutomaticIrrigationMoistureLimit() const {
    return automaticIrrigationMoistureLimit_;
}

bool isAutomaticIrrigationEnabled() const {
    return automaticIrrigationEnabled_;
}

bool isNotificationsEnabled() const {
    return notificationsEnabled_;
}

private:
String filename_;
String calibrationVolume_;
String calibrationTime_;
int scheduledIrrigationHour_[3];
bool scheduledIrrigationDays_[7];
String scheduledIrrigationVolume_;
bool scheduledIrrigationEnabled_;
String automaticIrrigationVolume_;
String automaticIrrigationMoistureLimit_;
bool automaticIrrigationEnabled_;
bool notificationsEnabled_;
};

Configuration config("/config.json");

```

7.1.2.3 *handleServer.h*

```

// WebServer with http access on port 80
ESP8266WebServer server(80);

// Timers

```

```

float calibrationStartTime = 0;
float calibrationElapsedTime = 0;
bool calibrationTimerActive = false;

void calibrationStartTimer() {
    calibrationStartTime = millis();
    calibrationTimerActive = true;
}

void calibrationStopTimer() {
    calibrationElapsedTime = millis() - calibrationStartTime;
    calibrationTimerActive = false;
}

// Handle manual irrigation
void handleStartManualIrrigation() {
    String manualIrrigationVolume = server.arg("manualIrrigationVolume");

    irrigation(manualIrrigationVolume.toFloat());
    if (lowTankLevel) {
        server.send(200, "text/plain", "El depósito tiene poca agua,
rellénalo.");
    } else {
        server.send(200, "text/plain", "Regando...");
        sendMailReporting("Riego manual realizado", "Se ha realizado un riego
manual.<br><br>Volumen: " + manualIrrigationVolume + " mL");
    }
}

// Handle scheduled irrigation
void handleConfigScheduledIrrigation() {
    int scheduledIrrigationHour[3] = {server.arg("hours").toInt(),
server.arg("minutes").toInt(), server.arg("seconds").toInt()};
    bool scheduledIrrigationDays[7] = {
        server.arg("L") == "true",
        server.arg("M") == "true",
        server.arg("X") == "true",
        server.arg("J") == "true",
        server.arg("V") == "true",
        server.arg("S") == "true",
        server.arg("D") == "true"
    };
    String scheduledIrrigationVolume =
server.arg("scheduledIrrigationVolume");

    config.setScheduledIrrigationHour(scheduledIrrigationHour);
    config.setScheduledIrrigationDays(scheduledIrrigationDays);
    config.setScheduledIrrigationVolume(scheduledIrrigationVolume);
    config.save();
}

```



```

    if (config.isScheduledIrrigationEnabled()) {
        updateScheduledIrrigationAlarm();
    }
    server.send(200, "text/plain", "Riego programado.");
}
void handleGetIrrigationConfig() {
    int scheduledIrrigationHour[3];
    const int *scheduledIrrigationHourFromConfig =
config.getScheduledIrrigationHour();
    for (int i = 0; i < 3; i++) {
        scheduledIrrigationHour[i] = scheduledIrrigationHourFromConfig[i];
    }
    bool scheduledIrrigationDays[7];
    const bool *scheduledIrrigationDaysFromConfig =
config.getScheduledIrrigationDays();
    for (int i = 0; i < 7; i++) {
        scheduledIrrigationDays[i] = scheduledIrrigationDaysFromConfig[i];
    }
    String scheduledIrrigationVolumeValue =
config.getScheduledIrrigationVolume();
    bool scheduledIrrigationEnabled =
config.isScheduledIrrigationEnabled();
    String automaticIrrigationVolumeValue =
config.getAutomaticIrrigationVolume();
    String automaticIrrigationMoistureLimitValue =
config.getAutomaticIrrigationMoistureLimit();
    bool automaticIrrigationEnabled =
config.isAutomaticIrrigationEnabled();
    bool notificationsEnabled = config.isNotificationsEnabled();

    DynamicJsonDocument json(400);
    JSONArray scheduledIrrigationHourArray =
json.createNestedArray("scheduledIrrigationHour");
    for (int i = 0; i < 3; i++) {
        scheduledIrrigationHourArray.add(scheduledIrrigationHour[i]);
    }
    JSONArray scheduledIrrigationDaysArray =
json.createNestedArray("scheduledIrrigationDays");
    for (int i = 0; i < 7; i++) {
        scheduledIrrigationDaysArray.add(scheduledIrrigationDays[i]);
    }

    json["scheduledIrrigationVolume"] = scheduledIrrigationVolumeValue;
    json["scheduledIrrigationEnabled"] = scheduledIrrigationEnabled;
    json["automaticIrrigationVolume"] = automaticIrrigationVolumeValue;
    json["automaticIrrigationMoistureLimit"] =
automaticIrrigationMoistureLimitValue;
    json["automaticIrrigationEnabled"] = automaticIrrigationEnabled;
    json["notificationsEnabled"] = notificationsEnabled;
}

```

```

String jsonString;
serializeJson(json, jsonString);

server.send(200, "application/json", jsonString);
}
void handleEnableScheduledIrrigation() {
    config.enableScheduledIrrigation();
    config.save();
    updateScheduledIrrigationAlarm();
    server.send(200);
}
void handleDisableScheduledIrrigation() {
    config.disableScheduledIrrigation();
    config.save();
    removeScheduledIrrigationAlarm();
    server.send(200);
}

// Handle automatic irrigation
void handleConfigAutomaticIrrigation() {
    String automaticIrrigationVolume =
server.arg("automaticIrrigationVolume");
    String automaticIrrigationMoistureLimit =
server.arg("automaticIrrigationMoistureLimit");
    config.setAutomaticIrrigationVolume(automaticIrrigationVolume);
    config.setAutomaticIrrigationMoistureLimit(automaticIrrigationMoistureL
imit);
    config.save();
    server.send(200, "text/plain", "Riego configurado.");
}
void handleGetSensorData() {
    float temperature = readTemperature();
    float humidity = readHumidity();
    int soilMoisture = readSoilMoisture();

    DynamicJsonDocument json(400);
    json["temperature"] = temperature;
    json["humidity"] = humidity;
    json["soilMoisture"] = soilMoisture;

    String jsonString;
    serializeJson(json, jsonString);

    server.send(200, "application/json", jsonString);
}
void handleEnableAutomaticIrrigation() {
    config.enableAutomaticIrrigation();

```

```

        config.save();
        server.send(200);
    }
    void handleDisableAutomaticIrrigation() {
        config.disableAutomaticIrrigation();
        config.save();
        server.send(200);
    }

    // Handle calibration
    void handleUpdateCalibration() {
        String calibrationVolume = server.arg("calibrationVolume");
        String calibrationTime = server.arg("calibrationTime");
        config.setCalibrationVolume(calibrationVolume);
        config.setCalibrationTime(calibrationTime);
        config.save();
        server.send(200, "text/plain", "Calibración actualizada.");
    }
    void handleGetCalibration() {
        String calibrationVolumeValue = config.getCalibrationVolume();
        String calibrationTimeValue = config.getCalibrationTime();

        DynamicJsonDocument json(256);
        json["calibrationVolume"] = calibrationVolumeValue;
        json["calibrationTime"] = calibrationTimeValue;
        String jsonString;
        serializeJson(json, jsonString);

        server.send(200, "application/json", jsonString);
    }
    void handleStartCalibration() {
        digitalWrite(D0, HIGH);
        calibrationStartTimer();
        server.send(200);
    }
    void handleStopCalibration() {
        float calibrationElapsedTimeSeconds;
        digitalWrite(D0, LOW);
        calibrationStopTimer();
        calibrationElapsedTimeSeconds = calibrationElapsedTime/1000;

        DynamicJsonDocument json(256);
        json["calibrationTime"] = String(calibrationElapsedTimeSeconds);
        String jsonString;
        serializeJson(json, jsonString);

        server.send(200, "application/json", jsonString);
    }
}

```

```

// Handle notifications
void handleEnableNotifications() {
    config.enableNotifications();
    config.save();
    server.send(200);
}
void handleDisableNotifications() {
    config.disableNotifications();
    config.save();
    server.send(200);
}

```

7.1.2.4 irrigation.h

```

Ticker irrigationTimer;

void finishIrrigation() {
    digitalWrite(D0, LOW);
    irrigationTimer.detach();
}

void irrigation(float volume) { // Volume in mL
    if (lowTankLevel) {
        return;
    }
    float irrigationTime;
    if (config.getCalibrationVolume() != 0) {
        irrigationTime =
volume*config.getCalibrationTime().toFloat()/config.getCalibrationVolume(
).toFloat();
    } else {
        irrigationTime = 0;
    }

    irrigationTimer.once(irrigationTime, finishIrrigation);
    digitalWrite(D0, HIGH);
}

```

7.1.2.5 schedule.h

```

#include <WiFiUdp.h>
#include <TimeLib.h>
#include <TimeAlarms.h>
#include <NTPClient.h>

```

```

WiFiUDP ntpUDP;

```

```

int timezone = 2; // UTC +2
NTPClient timeClient(ntpUDP, "pool.ntp.org", timezone * 3600, 60000);

AlarmID_t scheduledIrrigationAlarmID;

void scheduledIrrigation() {
    time_t t = now(); // Obtains the time now
    int weekDay = weekday(t)-2; // Obtains the week day (1: Sunday, 2:
Monday, ..., 7: Saturday)
    if (weekDay == -1) {
        weekDay = 6;
    }
    const bool *scheduledIrrigationDaysFromConfig =
config.getScheduledIrrigationDays();
    String scheduledIrrigationVolume =
config.getScheduledIrrigationVolume();

    if (scheduledIrrigationDaysFromConfig[weekDay]) {
        if (!lowTankLevel) {
            irrigation(scheduledIrrigationVolume.toFloat());
            sendMailReporting("Riego programado realizado", "Se ha realizado un
riego programado.<br><br>Volumen: " + scheduledIrrigationVolume + " mL");
        }
    }
}

void updateScheduledIrrigationAlarm() {
    const int *scheduledIrrigationHourFromConfig =
config.getScheduledIrrigationHour();
    Alarm.free(scheduledIrrigationAlarmID); // Remove the existing alarm
    scheduledIrrigationAlarmID =
Alarm.alarmRepeat(scheduledIrrigationHourFromConfig[0],
scheduledIrrigationHourFromConfig[1],
scheduledIrrigationHourFromConfig[2], scheduledIrrigation); // Creates a
new alarm with the new execution time
}

void removeScheduledIrrigationAlarm() {
    Alarm.free(scheduledIrrigationAlarmID);
}

```

7.1.2.6 HDC2010.h

```

const uint8_t HDC2010_ADDR = 0x40;

void writeRegister(uint8_t reg, uint8_t value) {
    Wire.beginTransmission(HDC2010_ADDR);
    Wire.write(reg);
    Wire.write(value);
}

```

```

    Wire.endTransmission();
}

uint16_t readRegister(uint8_t reg) {
    Wire.beginTransaction(HDC2010_ADDR);
    Wire.write(reg);
    Wire.endTransmission(false);
    Wire.requestFrom(HDC2010_ADDR, (uint8_t)2);
    uint16_t value = Wire.read() | Wire.read() << 8;
    return value;
}

float readTemperature() {
    float temperature = 0;
    uint16_t rawTemperature = 0;

    for (int i = 0; i < 9; i++) {
        writeRegister(0x0E, 0x00);
        // Start measurement of temperature and humidity
        writeRegister(0x0F, 0x01);
        // Wait to complete the measurement
        delay(15);
        rawTemperature = readRegister(0x00);
        temperature = ((float)rawTemperature / 65536) * 165 - 40;
        if (temperature < 100) {
            return temperature;
        }
    }
    return temperature;
}

float readHumidity() {
    float humidity = 0;
    uint16_t rawHumidity = 0;

    for (int i = 0; i < 9; i++) {
        writeRegister(0x0E, 0x00);
        // Start measurement of temperature and humidity
        writeRegister(0x0F, 0x01);
        // Wait to complete the measurement
        delay(15);
        rawHumidity = readRegister(0x02);
        humidity = ((float)rawHumidity / 65536) * 100;
        if (humidity < 99) {
            return humidity;
        }
    }
    return humidity;
}

```

7.1.2.7 *moisture.h*

```
const int SoilMoistureSensorAirValue = 1024; // 1024 value in air
const int SoilMoistureSensorWaterValue = 550; // 550 value in water
int soilMoistureValue = 0;
int soilMoisturePercent = 0;
```

```
int readSoilMoisture() {
    soilMoistureValue = analogRead(A0);
    soilMoisturePercent = map(soilMoistureValue,
    SoilMoistureSensorAirValue, SoilMoistureSensorWaterValue, 0, 100);
    if (soilMoisturePercent >= 100) {
        return 100;
    } else if (soilMoisturePercent <=0 ) {
        return 0;
    } else {
        return soilMoisturePercent;
    }
}
```

7.1.2.8 *automatic.h*

```
Ticker automaticIrrigationTimer;
Ticker mailReportingTimer;
bool recentAutomaticIrrigationDone = false;
bool automaticReportingMailSent = true;

int soilMoistureWhenIrrigation;
float temperatureWhenIrrigation;
float humidityWhenIrrigation;
String automaticIrrigationVolumeWhenIrrigation;

float volumeModificationTemperature;
float volumeModificationHumidity;

bool needAutomaticIrrigation() {
    return (readSoilMoisture() <
    config.getAutomaticIrrigationMoistureLimit().toInt()) &&
    config.isAutomaticIrrigationEnabled() && !recentAutomaticIrrigationDone;
}

void recentAutomaticIrrigationGone() {
    recentAutomaticIrrigationDone = false;
}

float getVolumeModification() {
    volumeModificationTemperature = 0.5 * readTemperature() - 10;
```

```

    if (volumeModificationTemperature < -5) {
        volumeModificationTemperature = -5;
    } else if (volumeModificationTemperature > 5) {
        volumeModificationTemperature = 5;
    }
    volumeModificationHumidity = -0.3 * readHumidity() + 15;
    if (volumeModificationHumidity < -15) {
        volumeModificationHumidity = -15;
    } else if (volumeModificationHumidity > 15) {
        volumeModificationHumidity = 15;
    }
    return volumeModificationTemperature + volumeModificationHumidity;
}

int getModifiedVolume(int volume) {
    return volume + (volume * getVolumeModification() / 100);
}

void updateIrrigationConditions() {
    automaticIrrigationVolumeWhenIrrigation =
String(getModifiedVolume(config.getAutomaticIrrigationVolume().toInt()));
    soilMoistureWhenIrrigation = readSoilMoisture();
    temperatureWhenIrrigation = readTemperature();
    humidityWhenIrrigation = readHumidity();
}

void reportAutomaticIrrigation() {
    if (config.isNotificationsEnabled()) {
        automaticReportingMailSent = false;
    }
}

void automaticIrrigation() {
    String automaticIrrigationVolume =
String(getModifiedVolume(config.getAutomaticIrrigationVolume().toInt()));
    if (!lowTankLevel) {
        irrigation(automaticIrrigationVolume.toFloat());
        recentAutomaticIrrigationDone = true;
        automaticIrrigationTimer.once(21600, recentAutomaticIrrigationGone);
        updateIrrigationConditions();
        mailReportingTimer.once(120, reportAutomaticIrrigation);
    }
}

void sendAutomaticIrrigationReportingMailIfNeeded() {
    if (!automaticReportingMailSent && config.isNotificationsEnabled()) {
        automaticReportingMailSent = true;
        sendMailReporting("Riego automático realizado", "Se ha realizado un
riego automático hace 2 minutos.<br><br>Volumen: " +

```



```

automaticIrrigationVolumeWhenIrrigation + " mL<br>Humedad en suelo antes
de regar: " + String(soilMoistureWhenIrrigation) + " %<br>Humedad en
suelo ahora: " + String(readSoilMoisture()) + " %<br>Temperatura
ambiente: " + String(temperatureWhenIrrigation, 1) + " °C<br>Humedad
ambiente: " + String(humidityWhenIrrigation, 0) + " %<br><br>No puede
realizarse otro riego automático hasta que no hayan pasado 6 horas desde
el último riego automático.");
    }
}

```

7.1.2.9 *tankLevel.h*

```

Ticker tankLowLevelTimer;
Ticker tankNotLowLevelTimer;
bool lowTankLevel = false;
bool tankLevelMailSent = false;

void setLowTankLevel() {
    lowTankLevel = true;
    digitalWrite(D6, HIGH);
    tankLevelMailSent = false;
}

void removeLowTankLevel() {
    lowTankLevel = false;
    digitalWrite(D6, LOW);
}

void processTankLevel() {
    if (digitalRead(D5)) {
        if (!tankLowLevelTimer.active() && !lowTankLevel) {
            tankLowLevelTimer.once(5, setLowTankLevel);
        }
        tankNotLowLevelTimer.detach();
    } else {
        if (!tankNotLowLevelTimer.active() && lowTankLevel) {
            tankNotLowLevelTimer.once(5, removeLowTankLevel);
        }
        tankLowLevelTimer.detach();
    }

    if (lowTankLevel && !tankLevelMailSent &&
config.isNotificationsEnabled()) {
        tankLevelMailSent = true;
        sendMailLowTankLevel();
    }
}

```

7.1.2.10 mail.h

```
const char* emailUser = "xxxxx@xxx.com";
const char* emailPassword = "xxxxx";
const char* nameFrom = "Riego remoto notificación";
const char* recipientEmail = "xxxxx@xxx.com";

EmailSender emailSend(emailUser, emailPassword, emailUser, nameFrom);
EmailSender::EmailMessage message;
EmailSender::Response resp;

void sendMailLowTankLevel() {
    if (!config.isNotificationsEnabled()) {
        return;
    }

    message.subject = "Depósito vacío";
    message.message = "El depósito de agua de tu planta está casi vacío,
    llénalo por favor!";

    resp = emailSend.send(recipientEmail, message);
}

void sendMailReporting(String subject, String body) {
    if (!config.isNotificationsEnabled()) {
        return;
    }

    message.subject = subject;
    message.message = body;

    resp = emailSend.send(recipientEmail, message);
}
```

7.1.2.11 secrets.h

```
#ifndef STASSID
#define STASSID "xxxxx"
#define STAPSK "xxxxx"
#endif

const char *ssid = STASSID;
const char *passPhrase = STAPSK;
```

7.1.2.12 builtinfiles.h

```
//https://notisrac.github.io/FileToArray/
```

```
// Main page
```

```

static const byte mainpage[] PROGMEM = {
    0x1f, 0x8b, 0x08, 0x08, 0x97, 0x7b, 0x52, 0x64, 0x00, 0x03, 0x6d, 0x61,
    0x69, 0x6e, 0x70, 0x61,
    0x67, 0x65, 0x2e, 0x68, 0x74, 0x6d, 0x6c, 0x00, 0xed, 0x5d, 0x59, 0x73,
    0xdb, 0x38, 0xf2, 0x7f,
    0xdf, 0xaa, 0xfd, 0x0e, 0x18, 0xce, 0x26, 0x56, 0x26, 0xd6, 0x65, 0xf9,
    0x48, 0x1c, 0xc9, 0x5b,
    0x1e, 0xdb, 0x49, 0x3c, 0xe5, 0x23, 0x65, 0x3b, 0xd9, 0x99, 0xff, 0xd4,
    0xd4, 0x7f, 0x21, 0x12,
    0x92, 0x38, 0xe1, 0xa1, 0x90, 0xa0, 0x8f, 0xc9, 0xe6, 0x61, 0x3f, 0x4a,
    0x1e, 0xf7, 0x61, 0x1e,
    0xa6, 0xf6, 0x6d, 0x1f, 0xd7, 0x5f, 0x6c, 0x1b, 0x00, 0x49, 0x81, 0x14,
    0x78, 0xe8, 0xb0, 0xc7,
    0x93, 0x22, 0x12, 0x5b, 0x12, 0x08, 0x34, 0x1a, 0x8d, 0x46, 0xe3, 0xd7,
    0xd0, 0x08, 0xee, 0x7e,
    0xb5, 0x7f, 0xba, 0x77, 0xf1, 0xc3, 0x9b, 0x03, 0x34, 0xa2, 0xb6, 0xb5,
    0xf3, 0xe7, 0x3f, 0x75,
    0xe3, 0x57, 0x82, 0xd0, 0x78, 0x45, 0x90, 0xba, 0xd4, 0xa4, 0x16, 0xd9,
    0x39, 0x3b, 0x3c, 0x78,
    0x75, 0x8a, 0xce, 0x0e, 0x8e, 0x4f, 0x2f, 0x4e, 0xbb, 0x4d, 0x91, 0x17,
    0x16, 0xb0, 0x4c, 0xe7,
    0x3d, 0xf2, 0x88, 0xd5, 0xd3, 0xfc, 0x91, 0xeb, 0x51, 0x3d, 0xa0, 0xc8,
    0xd4, 0x5d, 0x47, 0x43,
    0x23, 0x8f, 0x0c, 0x7a, 0xda, 0x88, 0xd2, 0xb1, 0xbf, 0xdd, 0x6c, 0x12,
    0x47, 0xf7, 0x6e, 0xc6,
    0x94, 0x18, 0x75, 0xda, 0x77, 0x5a, 0x8d, 0xa1, 0x4f, 0x31, 0x35, 0xf5,
    0x86, 0xee, 0xda, 0x4d,
    0xd3, 0xc6, 0x43, 0xe2, 0xff, 0xf5, 0x43, 0x0f, 0x9e, 0x6c, 0xef, 0x9e,
    0x18, 0xcf, 0x5f, 0xe9,
    0x17, 0xd7, 0x3f, 0xbc, 0x7a, 0x76, 0xf6, 0xe1, 0xe7, 0xe3, 0xb3, 0x67,
    0x7d, 0x83, 0x5c, 0xed,
    0x05, 0x9b, 0xff, 0xf7, 0xce, 0xae, 0x1f, 0xbf, 0x3e, 0xdb, 0xfb, 0xdb,
    0xc5, 0x80, 0xba, 0xc3,
    0x7d, 0x63, 0x6d, 0xf7, 0xbb, 0x76, 0x70, 0xd5, 0x3a, 0xfa, 0xf6, 0x78,
    0xe3, 0xe4, 0x6f, 0xdf,
    0x52, 0xfd, 0x74, 0x84, 0x77, 0x77, 0xb1, 0xbd, 0x6b, 0xd0, 0xfd, 0x9f,
    0x8f, 0xce, 0x5f, 0xfe,
    0xff, 0x78, 0x7d, 0xf8, 0x38, 0xf0, 0x3f, 0x8c, 0x7b, 0x7b, 0xbb, 0x6f,
    0xb5, 0x88, 0x57, 0x9b,
    0x50, 0x8c, 0x18, 0x47, 0x75, 0xf2, 0x21, 0x30, 0x2f, 0x7b, 0xda, 0x9e,
    0xeb, 0x50, 0xe2, 0xd0,
    0xfa, 0xc5, 0xcd, 0x98, 0x68, 0x48, 0x17, 0x9f, 0x7a, 0x1a, 0x25, 0xd7,
    0xb4, 0xc9, 0xa4, 0xf1,
    0x42, 0x1f, 0x61, 0xcf, 0x27, 0xb4, 0xf7, 0xf6, 0xe2, 0x65, 0xfd, 0x59,
    0x92, 0x8e, 0x83, 0x6d,
    0xd2, 0xd3, 0x2e, 0x4d, 0x72, 0x35, 0x86, 0x7e, 0x4b, 0xb5, 0xaf, 0x4c,
    0x83, 0x8e, 0x7a, 0x06,
    0xb9, 0x34, 0x75, 0x52, 0xe7, 0x1f, 0x56, 0x91, 0xe9, 0x98, 0xd4, 0xc4,
    0x56, 0xdd, 0xd7, 0xb1,

```

0x45, 0x7a, 0xed, 0x98, 0x92, 0x4f, 0x6f, 0x2c, 0x82, 0x28, 0x34, 0x1f,
0xb6, 0xaa, 0xfb, 0x7e,
0xf4, 0x50, 0xa4, 0x6f, 0x3e, 0x8a, 0x4f, 0x63, 0x6c, 0x18, 0xa6, 0x33,
0xdc, 0x46, 0xad, 0xf1,
0xf5, 0x0b, 0x64, 0x63, 0x6f, 0x68, 0x3a, 0xe2, 0x83, 0x78, 0xfe, 0x49,
0xbc, 0xf4, 0x5d, 0xe3,
0x06, 0x7d, 0x9c, 0x10, 0x18, 0x00, 0x57, 0xf5, 0x01, 0xb6, 0x4d, 0xeb,
0x66, 0x1b, 0xad, 0x9c,
0x93, 0xa1, 0x4b, 0xd0, 0xdb, 0xc3, 0x95, 0x55, 0x74, 0x81, 0x47, 0xae,
0x8d, 0x57, 0xd1, 0x2b,
0xe2, 0x90, 0x4b, 0x78, 0x7d, 0x47, 0x3c, 0x03, 0x3b, 0xf0, 0xc6, 0xc7,
0x8e, 0x5f, 0xf7, 0x89,
0x67, 0x0e, 0x26, 0x94, 0xc5, 0x9b, 0x81, 0xeb, 0xd9, 0x32, 0x6d, 0xc3,
0xf4, 0xc7, 0x16, 0x06,
0xba, 0x03, 0x8b, 0x44, 0x6c, 0xf0, 0x72, 0xf0, 0xb1, 0x6e, 0x98, 0x1e,
0xd1, 0xa9, 0xe9, 0x02,
0x93, 0xba, 0x6b, 0x05, 0xb6, 0x23, 0x15, 0xb0, 0xf1, 0xb5, 0x10, 0xcc,
0x36, 0xea, 0xb4, 0x26,
0x5d, 0x10, 0x8f, 0x44, 0xc7, 0x70, 0x40, 0xdd, 0x74, 0xfb, 0x0d, 0xd3,
0x19, 0x07, 0xb4, 0xce,
0x04, 0x8d, 0x4d, 0x87, 0x78, 0x65, 0x78, 0xf9, 0x39, 0xf0, 0xa9, 0x39,
0xb8, 0xa9, 0x87, 0xa3,
0x03, 0xcc, 0xc0, 0x6f, 0xe2, 0x49, 0x25, 0xb0, 0x65, 0x0e, 0x9d, 0xba,
0x49, 0x89, 0xed, 0x2b,
0x9e, 0x0a, 0x7e, 0xea, 0x7d, 0x97, 0x52, 0xd7, 0xde, 0x46, 0xed, 0x24,
0xb7, 0x63, 0xd7, 0x37,
0x45, 0x1f, 0x61, 0x06, 0x80, 0x42, 0x5f, 0x92, 0x34, 0xcf, 0x16, 0xee,
0x13, 0xeb, 0x47, 0x31,
0xc0, 0x01, 0x28, 0x81, 0xf6, 0x93, 0xcc, 0xf5, 0xa4, 0x3e, 0xee, 0xfb,
0x20, 0x25, 0x3a, 0x55,
0x9f, 0x77, 0xf9, 0xc7, 0x89, 0x82, 0x24, 0xeb, 0xb3, 0x9c, 0x3a, 0xef,
0x80, 0x82, 0xf5, 0xbe,
0xeb, 0x19, 0xc4, 0xdb, 0x46, 0x8e, 0xeb, 0x10, 0x29, 0xdb, 0x0d, 0x28,
0xcc, 0x59, 0x32, 0x95,
0x1f, 0x2b, 0xd8, 0xb3, 0x44, 0x17, 0x05, 0x95, 0xba, 0x87, 0x0d, 0x33,
0x00, 0x01, 0xad, 0xa7,
0x1e, 0x5e, 0xd7, 0xfd, 0x11, 0x36, 0xdc, 0xab, 0x6d, 0xe0, 0x14, 0x26,
0x0a, 0x6a, 0xa1, 0xf6,
0xf8, 0x1a, 0x75, 0xe0, 0xc7, 0x1b, 0xf6, 0x71, 0xad, 0xb5, 0x8a, 0xc2,
0xff, 0x8d, 0xf6, 0x93,
0x17, 0x29, 0xb5, 0xf4, 0xcd, 0x5f, 0x80, 0x8d, 0xf6, 0x66, 0x82, 0x24,
0xf5, 0x40, 0xff, 0x42,
0xa1, 0x4c, 0xc8, 0x43, 0xfd, 0x8e, 0x2f, 0x95, 0x0a, 0xb5, 0xa7, 0xdd,
0x6a, 0x3d, 0x52, 0xeb,
0x55, 0x7b, 0x8d, 0xcf, 0x93, 0xe6, 0x37, 0x68, 0x97, 0xa9, 0x00, 0xf6,
0x10, 0xb1, 0x10, 0xc5,
0x36, 0xbe, 0xfd, 0xcd, 0x45, 0xf6, 0xed, 0xe7, 0x6b, 0xd3, 0x76, 0x91,
0x41, 0x90, 0xe5, 0xfa,

0x42, 0xc6, 0x3e, 0xfa, 0xa6, 0x59, 0x28, 0xfb, 0xed, 0x81, 0xab, 0x07,
0xbe, 0x3c, 0x02, 0xb3,
0x48, 0xa0, 0xf3, 0x24, 0x6f, 0x74, 0xfd, 0xa0, 0x6f, 0xa7, 0xf5, 0xa3,
0x8f, 0xf5, 0xf7, 0x43,
0xcf, 0x0d, 0x1c, 0x03, 0x14, 0xd8, 0x72, 0x61, 0x30, 0xbf, 0x6e, 0x6d,
0x76, 0x8c, 0x4d, 0x43,
0xea, 0x74, 0xf8, 0xe0, 0x6a, 0x64, 0x52, 0x52, 0x3c, 0xfa, 0xf1, 0x28,
0x33, 0x4d, 0x46, 0x6b,
0xad, 0xf2, 0x63, 0x2d, 0x0f, 0xd9, 0x56, 0xe2, 0x89, 0x1e, 0x78, 0x3e,
0x63, 0x61, 0xec, 0x9a,
0xca, 0xd9, 0xb3, 0xcd, 0xdb, 0xe1, 0x53, 0x7a, 0xf2, 0x2b, 0x6b, 0xc8,
0x53, 0x3d, 0x96, 0x07,
0x3e, 0x57, 0x6c, 0xdb, 0x23, 0xf7, 0x32, 0x69, 0x12, 0xd4, 0xc2, 0x5b,
0xdf, 0xd0, 0xa7, 0xec,
0x0a, 0xac, 0x28, 0x43, 0x30, 0xc3, 0x1f, 0xa7, 0xf4, 0x6b, 0x3d, 0x29,
0x9f, 0x11, 0x31, 0x87,
0x23, 0x3a, 0x95, 0x9d, 0x12, 0x5b, 0x6a, 0xfe, 0x4c, 0x73, 0x81, 0x0d,
0xf6, 0xaf, 0x78, 0xac,
0xa2, 0xf2, 0x83, 0xc1, 0x40, 0x3d, 0x0e, 0xa9, 0xe1, 0xe3, 0x4f, 0xae,
0x42, 0x1e, 0xfb, 0xae,
0x25, 0x37, 0x91, 0x6b, 0x29, 0xb2, 0x07, 0x70, 0xe6, 0xa1, 0x69, 0x60,
0x9d, 0x99, 0xc1, 0xa2,
0x71, 0x68, 0x6d, 0xf5, 0x07, 0x53, 0xeb, 0x0b, 0xb3, 0xd1, 0x9e, 0x6b,
0x71, 0x78, 0x11, 0xd7,
0x97, 0xbb, 0x2b, 0x2b, 0xa4, 0x4a, 0x38, 0x59, 0x02, 0x08, 0xd7, 0xc6,
0x06, 0xac, 0x78, 0x13,
0xc2, 0xf2, 0x82, 0xca, 0x7f, 0xda, 0xe1, 0x9b, 0x17, 0x2a, 0x8e, 0x07,
0x3c, 0xa5, 0x39, 0xfe,
0x9a, 0x41, 0x25, 0x59, 0xeb, 0x8a, 0x17, 0x84, 0xa8, 0x4a, 0xc3, 0xc1,
0x97, 0x71, 0x35, 0x6e,
0x45, 0xcc, 0x5f, 0x38, 0x3b, 0xa1, 0x36, 0x41, 0xd6, 0x8b, 0xc9, 0xaa,
0x66, 0x3a, 0xcc, 0x66,
0xd7, 0xfb, 0x96, 0xab, 0xbf, 0x7f, 0x31, 0xd5, 0x83, 0x76, 0x89, 0x2e,
0xa4, 0x0d, 0x87, 0x6c,
0x41, 0x63, 0xc5, 0x4e, 0x4c, 0xcc, 0x19, 0xfa, 0xc2, 0x10, 0x9f, 0x4a,
0x08, 0xa9, 0x55, 0x0d,
0xf4, 0xc9, 0x1d, 0x6f, 0xa3, 0x0d, 0xc9, 0x68, 0x73, 0x05, 0x63, 0xc0,
0x62, 0x5b, 0xbc, 0x85,
0x86, 0xc8, 0x0f, 0xb5, 0x3a, 0x14, 0x99, 0x2c, 0x17, 0x9e, 0x60, 0xae,
0xbd, 0x21, 0x0d, 0xbf,
0xd4, 0x3f, 0xa1, 0x1c, 0x21, 0x2a, 0x50, 0x15, 0xf0, 0xc8, 0x98, 0x60,
0xca, 0xa6, 0x57, 0xf8,
0x36, 0x2d, 0x84, 0x0d, 0x2e, 0xb2, 0x48, 0x08, 0x1b, 0x09, 0x50, 0xa5,
0xe8, 0xa9, 0x3f, 0xc6,
0x3a, 0xb9, 0xfb, 0xee, 0xb6, 0xee, 0xaa, 0xbb, 0x9b, 0x89, 0xee, 0x6e,
0x26, 0xda, 0x49, 0x9a,
0xb2, 0x2c, 0x91, 0x2b, 0xd6, 0x9c, 0x69, 0x51, 0xf9, 0x96, 0x09, 0xaf,
0x8e, 0x0b, 0xf8, 0xeb,

0x9e, 0x24, 0xb6, 0x75, 0x67, 0x22, 0x5b, 0x7b, 0x7e, 0xa7, 0x32, 0x6b,
0x84, 0xa6, 0x6f, 0x59,
0x06, 0x21, 0xb6, 0x07, 0x1b, 0x39, 0x26, 0x4d, 0xe7, 0x29, 0xd5, 0x0d,
0xbe, 0xea, 0x23, 0x18,
0x14, 0xd3, 0x88, 0xb1, 0x46, 0xba, 0x87, 0x89, 0x65, 0x27, 0x14, 0x10,
0xb7, 0x1c, 0x49, 0x3b,
0x12, 0xaf, 0xfe, 0x9d, 0x29, 0x2f, 0xa5, 0x21, 0x9b, 0x7a, 0xae, 0x1c,
0x71, 0xcf, 0x67, 0x34,
0x79, 0xca, 0x9e, 0xa5, 0x4c, 0x9d, 0x62, 0x80, 0x42, 0xc9, 0xa0, 0x56,
0x69, 0x7b, 0x18, 0x71,
0x6e, 0xdf, 0x30, 0xb6, 0xd0, 0xc7, 0xb4, 0xd8, 0xd6, 0xa6, 0xc4, 0x96,
0xa8, 0xf7, 0x35, 0x6d,
0x27, 0x97, 0x34, 0xe5, 0x3a, 0xad, 0x58, 0xa5, 0xe5, 0x05, 0x50, 0xd8,
0x29, 0xb5, 0x22, 0x89,
0x46, 0xd6, 0x16, 0x6f, 0x64, 0xad, 0xb0, 0x91, 0x4e, 0x89, 0x46, 0xb2,
0x08, 0xaa, 0x85, 0xc3,
0x42, 0x00, 0x1f, 0xd3, 0x53, 0xae, 0x95, 0x98, 0x72, 0xe1, 0xc7, 0x58,
0x01, 0xd6, 0x36, 0xd2,
0x4a, 0xf5, 0xb5, 0x6e, 0x1b, 0x77, 0x8a, 0x9d, 0xa7, 0x7b, 0x9b, 0x03,
0xc6, 0x66, 0x01, 0xda,
0xa9, 0xa7, 0x91, 0x2a, 0x66, 0x2e, 0x4b, 0x9c, 0xf5, 0x65, 0x60, 0xb6,
0x48, 0x6c, 0x02, 0x3c,
0x4f, 0x74, 0xba, 0x34, 0x72, 0xf6, 0xaf, 0x4c, 0xaa, 0x8f, 0xd4, 0x2e,
0x6d, 0x0a, 0x35, 0xb0,
0x94, 0x3f, 0xb9, 0xe5, 0x49, 0xa8, 0xc4, 0xdd, 0x29, 0x11, 0x48, 0x41,
0x10, 0xe1, 0x61, 0xb0,
0x9f, 0x8e, 0x62, 0xf5, 0x8e, 0xd8, 0xe4, 0x7e, 0x83, 0xcc, 0xac, 0x0b,
0xa6, 0xc7, 0xa4, 0xc0,
0x4f, 0x6b, 0x9a, 0x89, 0x96, 0x82, 0x81, 0x4e, 0x5b, 0x45, 0x9c, 0xaf,
0x73, 0xa5, 0xdc, 0xfa,
0x82, 0x81, 0x63, 0x6b, 0x9f, 0xdc, 0xac, 0x45, 0x06, 0x34, 0x99, 0x13,
0xae, 0x74, 0xad, 0x84,
0x2e, 0x89, 0xd0, 0x44, 0x22, 0xaf, 0xb4, 0xd3, 0x31, 0xd1, 0xc1, 0x8d,
0x71, 0x96, 0x1b, 0xde,
0x6a, 0xac, 0x4f, 0xe3, 0x7c, 0xd1, 0xeb, 0xed, 0x3e, 0x81, 0xb5, 0x99,
0x94, 0xef, 0x7c, 0x14,
0x86, 0xd1, 0x34, 0x95, 0x78, 0x9f, 0xa9, 0xe6, 0x41, 0x2a, 0x57, 0x08,
0x65, 0x33, 0x35, 0x9f,
0x84, 0x0c, 0x52, 0xb9, 0xb9, 0x8b, 0xaf, 0x42, 0x06, 0x1b, 0x89, 0x68,
0x42, 0x91, 0x08, 0xb8,
0x36, 0x6d, 0xeb, 0x23, 0xa2, 0xbf, 0x27, 0x06, 0x7a, 0xaa, 0x52, 0x84,
0x22, 0x1b, 0x54, 0x44,
0x4b, 0x21, 0x5e, 0x15, 0x1e, 0xfa, 0xbe, 0xc6, 0x66, 0xc6, 0x54, 0x74,
0x21, 0x24, 0xd2, 0xe0,
0x0c, 0x24, 0x83, 0x16, 0x89, 0x6e, 0x77, 0xd6, 0x33, 0xb5, 0x5a, 0xd4,
0x55, 0xb0, 0x91, 0x29,
0xb9, 0xd0, 0xaa, 0x74, 0x9b, 0x3c, 0xb4, 0xc9, 0xc2, 0xc9, 0xcd, 0x30,
0x9e, 0xdc, 0x65, 0xb1,

0x49, 0xf6, 0x6a, 0x98, 0x97, 0xc8, 0x34, 0x7a, 0x9a, 0x00, 0x8a, 0x71,
0x2c, 0x94, 0x65, 0xeb,
0x16, 0xf6, 0xfd, 0x9e, 0x06, 0xbe, 0x91, 0x1c, 0x05, 0xed, 0x8e, 0xda,
0xac, 0xc2, 0x0a, 0x6d,
0xaf, 0xa4, 0x42, 0xd1, 0xa3, 0xb6, 0x5c, 0x6c, 0x2c, 0x4a, 0xad, 0xad,
0xec, 0xec, 0xb9, 0xce,
0xc0, 0x1c, 0x06, 0x1e, 0x46, 0x34, 0x80, 0x49, 0x43, 0x86, 0xee, 0x2a,
0x7b, 0x07, 0xc6, 0xc7,
0x01, 0xcc, 0x01, 0xb9, 0x2c, 0xf2, 0x83, 0xf0, 0x10, 0xf8, 0x27, 0x3a,
0xf1, 0x6e, 0x3f, 0x77,
0x9b, 0x63, 0x99, 0x92, 0xc4, 0x4b, 0x1a, 0xc6, 0x26, 0xc3, 0xb3, 0x2c,
0x86, 0x3b, 0xc6, 0x0e,
0xef, 0x0f, 0xed, 0x68, 0x88, 0xf7, 0xba, 0xa7, 0x29, 0x26, 0x41, 0x22,
0x80, 0xf2, 0x9c, 0x59,
0x76, 0x3e, 0xdf, 0x79, 0xf8, 0x2b, 0x54, 0x6a, 0x0e, 0x1d, 0xb5, 0x9d,
0x93, 0xd3, 0x8b, 0xc3,
0x97, 0x87, 0x7b, 0xbb, 0x7b, 0x87, 0xa7, 0x27, 0x07, 0xe7, 0x20, 0x49,
0x68, 0x20, 0xdd, 0x28,
0x8f, 0x2b, 0xc6, 0x2c, 0x72, 0xf3, 0x96, 0xdb, 0xb8, 0x1c, 0x30, 0x8e,
0xac, 0xc8, 0x56, 0xcc,
0x84, 0x68, 0x37, 0xd9, 0x04, 0x6f, 0x46, 0x58, 0x4c, 0x11, 0x69, 0xe1,
0xda, 0x09, 0x28, 0x54,
0xe3, 0x9d, 0xe5, 0x02, 0x31, 0x75, 0xcc, 0x1a, 0xf2, 0x0f, 0x1c, 0xdc,
0xb7, 0xa6, 0x24, 0x33,
0x91, 0x4e, 0x42, 0x94, 0xda, 0x8e, 0xba, 0x4b, 0x4d, 0xde, 0x27, 0x79,
0x14, 0x9a, 0x30, 0x0c,
0xf2, 0x67, 0x1c, 0x6e, 0x2a, 0x34, 0x75, 0x80, 0x2f, 0x7d, 0x0f, 0x7b,
0x7d, 0xd7, 0xee, 0xe3,
0xa9, 0x01, 0x91, 0x06, 0x2f, 0x76, 0xd7, 0xa6, 0x0a, 0xa5, 0x68, 0x8b,
0xde, 0xda, 0x43, 0xb9,
0x22, 0x08, 0xd4, 0xd3, 0x27, 0x5b, 0x18, 0xba, 0xe1, 0xd4, 0x39, 0xbd,
0xfa, 0xd8, 0x19, 0x36,
0x06, 0x6c, 0x71, 0x83, 0x4f, 0x7c, 0x0b, 0x63, 0xa3, 0xbd, 0xd6, 0xdc,
0x6a, 0xc1, 0xff, 0x4e,
0x7b, 0xbd, 0x01, 0x4f, 0x35, 0x84, 0x2d, 0xda, 0xd3, 0xf6, 0x42, 0x36,
0xd1, 0x14, 0x9f, 0xdd,
0x26, 0x56, 0x76, 0x6c, 0x64, 0xfa, 0xd4, 0xf5, 0x40, 0xac, 0x96, 0x81,
0x69, 0xc9, 0x9e, 0x45,
0xc3, 0x1e, 0x8e, 0x2a, 0xf7, 0x54, 0xe6, 0xec, 0x6e, 0x92, 0x50, 0x87,
0x11, 0x4a, 0xca, 0x80,
0xb7, 0xda, 0x00, 0xac, 0x70, 0x13, 0xf7, 0x1d, 0xba, 0xdb, 0x74, 0x9b,
0xb6, 0xe9, 0xeb, 0xc4,
0x82, 0xe9, 0x45, 0xdc, 0xc0, 0x6f, 0x12, 0x87, 0x78, 0x43, 0x1e, 0x74,
0xf7, 0x03, 0x7b, 0xcc,
0x54, 0xa4, 0x6e, 0x63, 0x07, 0x0f, 0x89, 0x0d, 0xc6, 0x5f, 0xea, 0x64,
0x9d, 0xf5, 0xb2, 0xde,
0x9e, 0x51, 0x64, 0x72, 0x67, 0xba, 0xbe, 0xee, 0x99, 0x63, 0x2a, 0x95,
0x33, 0x5c, 0x3d, 0x60,

0xcd, 0x34, 0x00, 0x80, 0x1d, 0x5c, 0xc2, 0x9b, 0x23, 0x68, 0x8e, 0x31,
0x54, 0x5b, 0xd9, 0x3f,
0x3d, 0x0e, 0x77, 0x7c, 0x8e, 0x5c, 0x98, 0xf8, 0xc6, 0xca, 0x2a, 0x1a,
0x04, 0x0e, 0xdf, 0xa5,
0x40, 0xb5, 0x27, 0xb2, 0x79, 0x63, 0x89, 0x71, 0x0f, 0xba, 0xcf, 0x63,
0x83, 0xe7, 0x02, 0x40,
0xf4, 0x26, 0xd4, 0x87, 0x84, 0x1e, 0x58, 0xbc, 0x3f, 0xdf, 0xde, 0x1c,
0x1a, 0xb5, 0x15, 0xc5,
0x7c, 0x58, 0x61, 0x16, 0x39, 0x49, 0x53, 0xa6, 0xa6, 0x60, 0x50, 0x1f,
0x61, 0x67, 0x48, 0xf2,
0xd9, 0x62, 0xe9, 0x12, 0x04, 0x74, 0x3d, 0xf2, 0x80, 0x1f, 0x87, 0x5c,
0xa1, 0xef, 0x8f, 0x8f,
0x5e, 0xc3, 0xf8, 0x9c, 0x91, 0x0f, 0x01, 0xf1, 0x69, 0x4d, 0x8e, 0xb3,
0xcb, 0x15, 0xa8, 0x69,
0xc3, 0xe0, 0x50, 0xa8, 0x34, 0xc0, 0x96, 0x4f, 0xf2, 0x4b, 0xed, 0x83,
0xed, 0xe4, 0x0c, 0xf4,
0xd8, 0x16, 0x4d, 0xeb, 0x05, 0x6a, 0x36, 0x51, 0x07, 0xf9, 0x64, 0x08,
0xcb, 0x81, 0xeb, 0xe7,
0x56, 0x3d, 0x26, 0xbe, 0x0f, 0x43, 0x0d, 0x35, 0xb5, 0x13, 0x17, 0xaa,
0xa0, 0x11, 0x06, 0x2c,
0xa8, 0x9b, 0x7d, 0xd3, 0x70, 0xe1, 0x8d, 0x3f, 0x66, 0x5c, 0x62, 0x64,
0x80, 0xfd, 0xf2, 0x89,
0x77, 0x09, 0xb9, 0x5e, 0x03, 0xbd, 0x01, 0x6c, 0x3a, 0xc0, 0x97, 0xae,
0xc7, 0xf6, 0xca, 0xe8,
0xed, 0xbf, 0xc0, 0x55, 0xb7, 0x78, 0x80, 0xde, 0x09, 0xc8, 0xa5, 0xdb,
0xd0, 0xa6, 0x24, 0xc9,
0x92, 0x39, 0x40, 0x35, 0x0a, 0xda, 0xd4, 0x08, 0x17, 0x4e, 0xa5, 0xac,
0x58, 0x02, 0x59, 0x35,
0xdc, 0x31, 0x71, 0x6a, 0xda, 0x9b, 0xd3, 0xf3, 0x0b, 0x6d, 0x15, 0x69,
0xa0, 0xa2, 0x6c, 0x88,
0x4e, 0xe4, 0x51, 0x83, 0x7c, 0xea, 0x05, 0x44, 0x25, 0xbf, 0x88, 0x88,
0x4f, 0x1c, 0x43, 0x29,
0xe0, 0x4f, 0x88, 0x80, 0x4c, 0x67, 0x68, 0x1f, 0xa0, 0xf0, 0x72, 0x19,
0x50, 0x09, 0x08, 0x06,
0xed, 0x00, 0x64, 0x0d, 0xed, 0xc0, 0x22, 0xc7, 0x77, 0x40, 0x4c, 0x62,
0x8f, 0xb9, 0x5c, 0x61,
0x1c, 0x88, 0x87, 0xb3, 0x07, 0x92, 0x29, 0x97, 0x4f, 0xe8, 0x85, 0x18,
0xd3, 0x5a, 0x81, 0x42,
0xb2, 0x34, 0xd1, 0x2f, 0xd6, 0x89, 0x9c, 0x3e, 0x60, 0x00, 0x0f, 0x4c,
0x4d, 0x19, 0x7b, 0xbb,
0xec, 0x3d, 0x34, 0x69, 0x61, 0xee, 0x3d, 0x03, 0x1a, 0x0f, 0x0c, 0x75,
0x4d, 0x6c, 0x11, 0xa8,
0x95, 0xd4, 0x31, 0xa5, 0x20, 0x56, 0xd3, 0x3a, 0x3c, 0x3d, 0x0d, 0x23,
0x4e, 0x5c, 0xc7, 0x03,
0x0c, 0x72, 0xc3, 0x36, 0xa3, 0x89, 0x98, 0x7b, 0x6c, 0x76, 0x14, 0x77,
0x95, 0x29, 0x1e, 0xab,
0xcf, 0x6b, 0x9f, 0xb3, 0xda, 0xa8, 0xd7, 0xeb, 0xa5, 0x26, 0x62, 0x63,
0x1f, 0x96, 0xef, 0x4c,

0x12, 0x2c, 0xe9, 0x16, 0xc1, 0x5e, 0x24, 0x61, 0x2e, 0x74, 0x21, 0x94,
0x23, 0xd3, 0x1e, 0x9b,
0xe1, 0x96, 0x15, 0x1b, 0x2f, 0xcf, 0xfc, 0x05, 0xc3, 0x2c, 0x41, 0xbe,
0xc9, 0xa6, 0x13, 0x9f,
0x4b, 0x84, 0x49, 0x2c, 0x9e, 0x4c, 0xea, 0x26, 0x3e, 0x29, 0x84, 0x93,
0x12, 0xd8, 0x27, 0x59,
0x82, 0xf1, 0x07, 0x58, 0x9f, 0x23, 0xab, 0x1a, 0xd9, 0x5a, 0x79, 0xe9,
0x61, 0x61, 0xf4, 0x18,
0xb5, 0x89, 0x08, 0x42, 0x06, 0x76, 0x0a, 0xc3, 0x3c, 0x6a, 0xc8, 0x94,
0x2c, 0x33, 0x09, 0x05,
0xc5, 0xeb, 0x50, 0x7a, 0xef, 0x55, 0xa0, 0x94, 0xae, 0x5c, 0x21, 0x84,
0x83, 0xc7, 0xbb, 0x27,
0x6f, 0x77, 0x8f, 0xba, 0xcd, 0xc4, 0xa3, 0x10, 0x65, 0x74, 0xfb, 0x1e,
0x6a, 0xa6, 0x19, 0xe0,
0x5b, 0xd9, 0x36, 0xa1, 0x23, 0xd7, 0xe0, 0x78, 0x89, 0xc2, 0xf2, 0xc3,
0x07, 0x1d, 0xd6, 0x60,
0xb6, 0x5b, 0x48, 0x8f, 0xb1, 0x13, 0x60, 0xeb, 0xd0, 0x83, 0xb5, 0x90,
0xeb, 0x90, 0x86, 0xd8,
0x5a, 0xc6, 0xb6, 0x9a, 0x00, 0xc6, 0xa8, 0x0a, 0xd4, 0x08, 0x33, 0xe5,
0x30, 0x7e, 0x1e, 0xa1,
0x81, 0xe7, 0x84, 0x36, 0x56, 0x09, 0x89, 0x04, 0x76, 0xe3, 0x50, 0xb5,
0xb3, 0xc2, 0x76, 0xd5,
0x59, 0x57, 0x93, 0xd4, 0xde, 0xb1, 0x2d, 0x73, 0x10, 0xb2, 0x78, 0x75,
0xa6, 0xb1, 0x91, 0x4a,
0xda, 0xa9, 0xed, 0xf1, 0xb4, 0x1c, 0x39, 0xd2, 0xdb, 0x9a, 0x9c, 0x1e,
0x88, 0x05, 0xdb, 0x51,
0x21, 0x86, 0xb8, 0x01, 0x19, 0x02, 0xf2, 0x5d, 0xd0, 0xf0, 0xe4, 0x43,
0x06, 0xcb, 0x1c, 0x1c,
0x66, 0x3d, 0xbb, 0xc4, 0x56, 0x00, 0x55, 0xc1, 0xbf, 0xcf, 0x6c, 0x2e,
0x29, 0x1c, 0x69, 0xe3,
0x3c, 0x85, 0x4e, 0x44, 0x54, 0x29, 0x57, 0x78, 0xf6, 0x51, 0x8e, 0xdc,
0x54, 0x80, 0xa8, 0xc9,
0xd4, 0x22, 0x9d, 0x09, 0xcb, 0x6c, 0x28, 0x5f, 0x11, 0xfa, 0xd3, 0x42,
0x8d, 0x42, 0x8f, 0x09,
0x00, 0xb0, 0x17, 0xf9, 0xbf, 0x53, 0xb4, 0xfa, 0x01, 0x48, 0x9c, 0x3b,
0x0b, 0x2b, 0xba, 0x6d,
0x44, 0xdd, 0x0b, 0xb7, 0x30, 0x41, 0xc3, 0x74, 0xb0, 0x80, 0xef, 0xf3,
0x15, 0x4c, 0xdb, 0x39,
0x23, 0x43, 0xec, 0x75, 0x9b, 0x82, 0xd6, 0x5c, 0x6c, 0x4c, 0x61, 0xa6,
0x28, 0xc5, 0xa6, 0x2f,
0x8f, 0x83, 0x2c, 0x8b, 0xc6, 0x9f, 0x36, 0xc6, 0x1e, 0x7f, 0xdd, 0x27,
0x03, 0x1c, 0x58, 0x6a,
0x30, 0xc2, 0xd2, 0xcc, 0x08, 0x26, 0xaa, 0x54, 0x8c, 0x62, 0x52, 0x25,
0x67, 0x45, 0x32, 0xa9,
0xea, 0xf7, 0x87, 0x66, 0x58, 0x52, 0x00, 0x05, 0xb5, 0x39, 0x2a, 0x83,
0x14, 0x68, 0x28, 0xcf,
0xd7, 0xdc, 0xcf, 0xae, 0x25, 0x4f, 0x3b, 0x01, 0x69, 0x3c, 0x1e, 0x5b,
0x21, 0xf4, 0x68, 0x5e,

0xd7, 0xaf, 0xae, 0xae, 0xea, 0x4c, 0xff, 0xeb, 0x81, 0x67, 0x11, 0x47,
0x77, 0x01, 0x1f, 0x6b,
0xea, 0x95, 0x93, 0xa5, 0x39, 0x90, 0x85, 0x2c, 0xd8, 0xd9, 0xd1, 0x05,
0x4b, 0xa5, 0x10, 0x46,
0xd4, 0xff, 0xf9, 0x50, 0x06, 0x4b, 0x65, 0x91, 0x06, 0x4b, 0x65, 0xd1,
0x46, 0xc4, 0xd5, 0xbc,
0x88, 0x83, 0xa5, 0x25, 0xa1, 0x0e, 0x96, 0xee, 0x18, 0x79, 0xc8, 0x1c,
0x7f, 0x15, 0x8d, 0xda,
0xe3, 0xc7, 0x42, 0x2f, 0x81, 0xf1, 0xc0, 0xe7, 0x9c, 0xaf, 0xb5, 0x5a,
0x85, 0x9c, 0xb2, 0x04,
0x8c, 0xbd, 0xc6, 0x8e, 0x61, 0x11, 0xde, 0x2e, 0x2c, 0xc4, 0x04, 0x0d,
0x3c, 0xd7, 0xe6, 0xe1,
0x35, 0xc0, 0xc3, 0x68, 0x44, 0x3c, 0x52, 0x4c, 0x85, 0xe9, 0x5e, 0x54,
0xff, 0x02, 0x16, 0x32,
0x10, 0xbc, 0x90, 0xe6, 0x24, 0x2b, 0x47, 0xa5, 0xa2, 0x24, 0x94, 0x43,
0xae, 0x94, 0xa5, 0x1a,
0x51, 0x52, 0xc0, 0xb0, 0x82, 0x47, 0x9f, 0x32, 0x55, 0x88, 0xf5, 0x61,
0x8c, 0x3d, 0x6c, 0xfb,
0xcc, 0x20, 0xa9, 0x17, 0xbe, 0x9e, 0x86, 0x9e, 0x22, 0x31, 0x83, 0xdf,
0x9e, 0x1d, 0xee, 0xb9,
0x30, 0x8c, 0x0e, 0x4c, 0xfb, 0x5a, 0x96, 0xeb, 0x9a, 0xb5, 0x7e, 0x3e,
0x69, 0xf0, 0xe5, 0xba,
0xd0, 0x1b, 0x11, 0xfc, 0x64, 0xf8, 0x24, 0xf2, 0x47, 0x09, 0x56, 0x4e,
0xb2, 0x52, 0x21, 0x9e,
0xdf, 0x01, 0x3c, 0xbe, 0x39, 0x3b, 0x7d, 0x75, 0xb6, 0x7b, 0xbc, 0xbb,
0x7f, 0x3a, 0x03, 0x80,
0x4c, 0xc0, 0x94, 0x34, 0x4c, 0x53, 0xd6, 0x98, 0x11, 0xa8, 0x75, 0x14,
0x40, 0xad, 0x5d, 0x2a,
0x4e, 0x27, 0x83, 0x34, 0x1f, 0x3c, 0x63, 0x23, 0xb0, 0x88, 0xa1, 0xc6,
0x69, 0x39, 0x8f, 0x43,
0xa8, 0x56, 0x02, 0xc0, 0x66, 0x63, 0xb4, 0x4d, 0x1e, 0x41, 0x92, 0x70,
0x98, 0x10, 0x8c, 0x52,
0x3c, 0x2a, 0x3c, 0x96, 0x14, 0xf2, 0xfe, 0xed, 0xaf, 0xd8, 0x4f, 0x8b,
0x58, 0x21, 0x64, 0x85,
0x30, 0xdb, 0xad, 0x99, 0xa4, 0x19, 0xc2, 0xb4, 0x70, 0xb4, 0x44, 0xcc,
0x46, 0x82, 0x67, 0x22,
0xe3, 0x5b, 0x5e, 0x88, 0xc7, 0x1f, 0x00, 0x94, 0x1d, 0xc5, 0x80, 0x6c,
0x39, 0xf4, 0x8e, 0x97,
0x4c, 0xef, 0xfb, 0x25, 0xd3, 0xfb, 0x6e, 0xc9, 0xf4, 0xde, 0x2d, 0x99,
0xde, 0xf9, 0x92, 0xe9,
0xed, 0x67, 0xd0, 0x8b, 0xf4, 0x36, 0x4f, 0x73, 0xb5, 0x9d, 0xd7, 0xae,
0x87, 0x73, 0x8d, 0xc3,
0xf4, 0xfc, 0x8d, 0xb5, 0x58, 0x3a, 0x2a, 0x2d, 0x29, 0x71, 0xb6, 0x5a,
0x0b, 0x6f, 0x2e, 0x7b,
0xfe, 0xbf, 0x76, 0x03, 0xcf, 0xcf, 0x9c, 0xfe, 0xe1, 0xd3, 0x78, 0xf6,
0x2b, 0xfd, 0x87, 0x54,
0xef, 0xb6, 0xa3, 0xae, 0x29, 0x0b, 0x97, 0xee, 0x5a, 0x0e, 0xcf, 0xc7,
0xa6, 0x13, 0x50, 0x92,

0xcd, 0x75, 0xfc, 0xfc, 0x81, 0xf1, 0x7d, 0x4e, 0xc0, 0xc8, 0x1b, 0xd9,
0x7c, 0xc7, 0xcf, 0x63,
0xbe, 0x33, 0xec, 0x5a, 0x81, 0x13, 0x5a, 0x62, 0x73, 0x68, 0xb6, 0xdd,
0x1d, 0x05, 0xaf, 0xcb,
0xdc, 0xe3, 0x51, 0xb3, 0xad, 0xf6, 0x8e, 0x45, 0xae, 0x34, 0x3d, 0x75,
0xbe, 0xb7, 0x77, 0x3e,
0xcd, 0x61, 0xec, 0x21, 0xbf, 0xf1, 0xdc, 0x21, 0x60, 0x92, 0x69, 0x2f,
0x79, 0x66, 0x3f, 0x38,
0x61, 0x0a, 0x04, 0xad, 0xdc, 0xa0, 0xa0, 0x28, 0xd2, 0xe0, 0xb2, 0x60,
0x61, 0xfd, 0x46, 0xb8,
0xc8, 0xfb, 0x35, 0x4d, 0x1c, 0x8c, 0xd5, 0x9e, 0xe4, 0x22, 0xde, 0xa9,
0xfa, 0x1e, 0xb1, 0xdd,
0x4b, 0x32, 0xa9, 0x9d, 0xe5, 0x86, 0xe4, 0x86, 0x9f, 0x95, 0x94, 0xb1,
0x61, 0x14, 0x93, 0x2d,
0xc4, 0x74, 0x09, 0x69, 0x15, 0x8d, 0x4c, 0x15, 0x39, 0xb8, 0xd7, 0xc8,
0x41, 0xe6, 0x70, 0x2c,
0x14, 0x3d, 0xa0, 0x55, 0xf4, 0xa0, 0x8a, 0x1e, 0x54, 0xd1, 0x83, 0x3f,
0x62, 0xf4, 0x00, 0x24,
0x71, 0x06, 0x88, 0xc3, 0xf3, 0xc4, 0xf7, 0x9e, 0x00, 0x31, 0xba, 0x0e,
0xf1, 0xd1, 0x0d, 0xba,
0x64, 0x5f, 0xf7, 0x83, 0xc9, 0x8c, 0xf4, 0xe0, 0xf6, 0x33, 0x78, 0xd2,
0x30, 0x0b, 0xe9, 0xed,
0x67, 0x87, 0xef, 0x89, 0x5c, 0xe6, 0x99, 0xba, 0x49, 0x2c, 0x42, 0xcb,
0xb1, 0xa7, 0x51, 0x43,
0xd2, 0xf6, 0x39, 0x68, 0x9b, 0x77, 0x73, 0x4e, 0xc0, 0x00, 0x50, 0xd7,
0xdb, 0xb5, 0xac, 0x9a,
0x16, 0x7e, 0x0b, 0x27, 0x73, 0x2d, 0x0a, 0x9b, 0x7a, 0x0a, 0x6d, 0x1d,
0xf1, 0x10, 0x47, 0x48,
0xf5, 0xc7, 0xd6, 0x4f, 0xb9, 0xeb, 0x6d, 0x31, 0xb9, 0xc7, 0xc7, 0x09,
0x7a, 0xed, 0x85, 0xe9,
0x7d, 0x9f, 0xa0, 0xb7, 0xb6, 0x30, 0xbd, 0xef, 0x12, 0xf4, 0x3a, 0x0b,
0xd3, 0x7b, 0x97, 0xa0,
0xb7, 0xbe, 0x30, 0xbd, 0xf3, 0x04, 0xbd, 0x8d, 0x85, 0xe9, 0xed, 0x27,
0xe8, 0x6d, 0x2e, 0x4c,
0x6f, 0xc4, 0x9c, 0x9a, 0x99, 0xc3, 0x62, 0x99, 0xde, 0x51, 0x41, 0x60,
0x4c, 0x6e, 0xd9, 0x16,
0x8e, 0xc9, 0x32, 0xda, 0x8e, 0x7c, 0x9c, 0x19, 0x5a, 0xf7, 0x85, 0x7b,
0xb1, 0x8c, 0xd6, 0x23,
0x4f, 0x65, 0x96, 0xd6, 0xb3, 0x82, 0x4b, 0xcb, 0xe0, 0x67, 0x3a, 0x46,
0xb9, 0xac, 0x28, 0x25,
0x4b, 0x4b, 0x3b, 0x43, 0x14, 0xa5, 0xd9, 0xce, 0x12, 0x65, 0x7a, 0x5f,
0x8a, 0x13, 0x45, 0x51,
0x5a, 0xd2, 0xc9, 0xa2, 0x28, 0xcd, 0x85, 0xb2, 0xa3, 0x8a, 0xe5, 0x90,
0x76, 0xaa, 0xf4, 0x3c,
0x68, 0x3b, 0x45, 0xe2, 0x7e, 0x11, 0x37, 0x4b, 0xa5, 0x4f, 0x20, 0x45,
0x29, 0xf3, 0x24, 0xd2,
0xec, 0x30, 0x5d, 0x26, 0x99, 0x79, 0x2c, 0x28, 0x4a, 0xc5, 0x0e, 0x62,
0x06, 0x77, 0xe1, 0x39,

0xa5, 0xbb, 0x66, 0x2f, 0x4f, 0xc8, 0x73, 0x7a, 0x0b, 0x2c, 0x2d, 0xe6,
0x31, 0xb0, 0x54, 0xda,
0x6b, 0x88, 0xfa, 0x3a, 0xbf, 0xe7, 0xc0, 0xd2, 0x2c, 0xde, 0x03, 0x4b,
0xb3, 0x78, 0x10, 0x11,
0x87, 0x8b, 0x78, 0x11, 0x2c, 0x2d, 0xd1, 0x93, 0x60, 0xe9, 0x9e, 0xbc,
0x89, 0x3c, 0x20, 0x9d,
0xe5, 0x9e, 0x29, 0x97, 0x0b, 0x55, 0xe6, 0x95, 0x09, 0x46, 0xea, 0x0a,
0x24, 0x6b, 0xc1, 0xa2,
0x20, 0x49, 0x33, 0x5b, 0x98, 0x73, 0x19, 0xd8, 0xc9, 0xfc, 0x7c, 0x75,
0x20, 0xa6, 0x27, 0x2c,
0x1d, 0x93, 0x19, 0x29, 0x4e, 0xbd, 0x97, 0xf0, 0xee, 0x73, 0x35, 0x60,
0x59, 0x6e, 0xe4, 0x9c,
0x2e, 0x9b, 0xec, 0x68, 0x01, 0x57, 0xdf, 0x9d, 0x9f, 0x9e, 0x34, 0xc6,
0xec, 0x16, 0x96, 0x5a,
0xda, 0xdf, 0xca, 0xd7, 0x74, 0x96, 0xd8, 0x14, 0xd4, 0x75, 0x62, 0x98,
0x60, 0x7f, 0x5c, 0x84,
0xb9, 0xd7, 0x43, 0xc4, 0x4a, 0x0b, 0xef, 0xd8, 0x3a, 0xc0, 0xc8, 0x17,
0xf3, 0x93, 0x01, 0x05,
0x81, 0xbd, 0x88, 0x9f, 0x46, 0x46, 0x91, 0x02, 0x83, 0x91, 0x41, 0x7c,
0x1f, 0xdf, 0xf8, 0x05,
0xc4, 0x59, 0x91, 0xf9, 0x88, 0x0b, 0xe8, 0x54, 0x40, 0x5e, 0x14, 0x9a,
0xaf, 0x01, 0x81, 0x55,
0x8c, 0x82, 0x16, 0xc2, 0x52, 0x25, 0x9a, 0x60, 0xdf, 0xc2, 0xb4, 0xd9,
0x69, 0xfb, 0xbc, 0x3e,
0x64, 0x16, 0x9a, 0xaf, 0x81, 0x63, 0x17, 0xa0, 0x53, 0xe0, 0x11, 0x30,
0x42, 0x26, 0x2d, 0x68,
0x27, 0x51, 0x76, 0xbe, 0xe6, 0x14, 0x22, 0xcb, 0x2e, 0x55, 0xa2, 0x09,
0xc5, 0x69, 0xf4, 0x04,
0x71, 0xd5, 0x73, 0x3e, 0x9b, 0x4a, 0xcc, 0xa8, 0x37, 0x9e, 0xab, 0x13,
0x1f, 0xb3, 0x19, 0x35,
0x3d, 0x9f, 0xb0, 0xe7, 0xe1, 0x9b, 0xac, 0xf9, 0x92, 0x4f, 0x7a, 0x6e,
0x47, 0x8c, 0xad, 0xed,
0xea, 0x22, 0x2c, 0x36, 0x40, 0xdd, 0x73, 0xea, 0x99, 0xce, 0xb0, 0xf6,
0x04, 0xcc, 0x88, 0x71,
0xce, 0x4e, 0x53, 0xd5, 0xd6, 0x56, 0xd1, 0x4a, 0x6b, 0xa5, 0x08, 0xb9,
0x2c, 0xe0, 0x9d, 0xe5,
0xb0, 0xd4, 0xbe, 0x2f, 0x96, 0x52, 0x2e, 0x5b, 0x0e, 0x4b, 0x6b, 0x45,
0x2c, 0xdd, 0x91, 0x4e,
0x30, 0x1b, 0x56, 0xac, 0xcb, 0x4b, 0x89, 0x1e, 0x45, 0x69, 0x00, 0x00,
0xa2, 0xc6, 0xa8, 0x9a,
0x40, 0x0f, 0x9c, 0x0c, 0x13, 0x75, 0xb3, 0x38, 0x6b, 0x58, 0xc4, 0x19,
0xd2, 0x11, 0x94, 0x79,
0xfa, 0xb4, 0x14, 0x90, 0x61, 0x4b, 0x63, 0x06, 0xad, 0x1f, 0xcd, 0x9f,
0x4a, 0x91, 0x60, 0x29,
0x8a, 0x7b, 0x98, 0x3f, 0xcd, 0xb8, 0x5b, 0x23, 0xa7, 0x1c, 0xb4, 0x13,
0x3e, 0x5e, 0x64, 0x4c,
0xb3, 0x97, 0x95, 0xe5, 0xa9, 0x6f, 0xd2, 0xc3, 0x57, 0x6b, 0x6f, 0x6c,
0xe1, 0x17, 0xea, 0x0c,

0xff, 0x7e, 0xe0, 0xc4, 0x2a, 0xbb, 0x73, 0xf6, 0x21, 0x73, 0x05, 0x92,
0xfa, 0x30, 0xef, 0x2a,
0x35, 0x4b, 0xa3, 0x89, 0xe5, 0x28, 0xbf, 0xed, 0xf4, 0xca, 0xb5, 0x90,
0x18, 0xb3, 0x17, 0xac,
0x62, 0x77, 0x22, 0xbb, 0x6e, 0xa9, 0x69, 0x33, 0x8b, 0x78, 0xc2, 0x7d,
0xec, 0x27, 0x91, 0xcb,
0x5e, 0xce, 0xb3, 0xbb, 0x83, 0x09, 0x53, 0x5a, 0x38, 0xd9, 0x75, 0x17,
0x13, 0x4e, 0xf6, 0x26,
0xff, 0x3d, 0x0b, 0x47, 0x85, 0x46, 0x8a, 0xc5, 0xa2, 0xaa, 0xb5, 0x98,
0x40, 0x54, 0xdf, 0x69,
0x9d, 0x5d, 0x14, 0x33, 0x3e, 0xca, 0x72, 0x3e, 0xf3, 0xbf, 0xed, 0xa5,
0xc8, 0xbb, 0xa3, 0xa0,
0xa5, 0x13, 0xd8, 0x7d, 0xe2, 0x1d, 0xf2, 0xa3, 0x22, 0x39, 0x31, 0x4b,
0xf5, 0x21, 0xd2, 0x4c,
0x40, 0x23, 0x68, 0xdb, 0xa6, 0xf3, 0x2e, 0xb4, 0x4e, 0xad, 0xfc, 0x82,
0xf8, 0x3a, 0x2a, 0xd8,
0x66, 0xc1, 0xc1, 0x2c, 0x85, 0x93, 0x98, 0x55, 0x88, 0xe1, 0x3d, 0xb9,
0x19, 0x03, 0x04, 0xf6,
0x13, 0xdd, 0xcf, 0x3f, 0x9a, 0x30, 0x61, 0x01, 0x2a, 0xef, 0xb9, 0x06,
0xe3, 0x40, 0x1c, 0x57,
0xb8, 0x1a, 0x99, 0xfa, 0x28, 0x47, 0x1b, 0x98, 0x8a, 0x46, 0x75, 0xba,
0x68, 0xfd, 0x19, 0xfa,
0xc7, 0x3f, 0x62, 0x1a, 0x3b, 0x68, 0x63, 0x0b, 0x9a, 0x64, 0xf3, 0x62,
0xfd, 0x59, 0x7d, 0x63,
0x0b, 0xf9, 0xc0, 0x09, 0x9b, 0x16, 0xfa, 0xed, 0xbf, 0x0d, 0x73, 0x28,
0x66, 0x05, 0xb8, 0xe6,
0x16, 0xe6, 0x11, 0x75, 0xfe, 0xc8, 0xb9, 0xfd, 0x8f, 0x4d, 0xbc, 0x10,
0x52, 0xb5, 0x10, 0xb6,
0xd0, 0xf3, 0x7c, 0x55, 0x9c, 0xe9, 0x54, 0x05, 0x4b, 0x59, 0x0a, 0x9a,
0x55, 0xa7, 0x50, 0xd8,
0xc1, 0xf8, 0xde, 0x24, 0x5d, 0x20, 0xea, 0xc7, 0x8f, 0xe5, 0xc1, 0x78,
0xbe, 0x99, 0x2c, 0xd1,
0x6e, 0x6d, 0xf0, 0x22, 0x51, 0xce, 0x57, 0xbd, 0x1e, 0x7a, 0x96, 0xce,
0x68, 0x77, 0xa6, 0x72,
0xb6, 0x50, 0x3a, 0x6b, 0xad, 0x95, 0xce, 0x79, 0x5e, 0x1c, 0x61, 0x64,
0xf1, 0x63, 0x9f, 0xd0,
0xbd, 0xc0, 0x87, 0xc5, 0x0a, 0xd4, 0xdc, 0x34, 0x4c, 0x7a, 0x53, 0xfb,
0xfb, 0xf9, 0xed, 0xbf,
0x2d, 0x1e, 0xc5, 0xc6, 0x3a, 0x19, 0x53, 0x76, 0xc2, 0x0a, 0x7b, 0xb7,
0x9f, 0x75, 0x4a, 0x40,
0x85, 0x99, 0xec, 0x6f, 0xff, 0xe5, 0x01, 0x58, 0xf1, 0xff, 0xbe, 0x78,
0xd8, 0x57, 0x88, 0x3d,
0xc2, 0x09, 0x3c, 0xd8, 0x72, 0xe8, 0x50, 0x11, 0xd8, 0xe6, 0xb9, 0xab,
0x20, 0xa2, 0x22, 0xd4,
0xc9, 0x46, 0x41, 0x90, 0xe8, 0x4e, 0xe6, 0x35, 0x88, 0x59, 0xe4, 0xed,
0xc4, 0x53, 0xb8, 0x94,
0x89, 0xce, 0x92, 0x49, 0x22, 0x58, 0x3f, 0x04, 0x41, 0x10, 0x14, 0x38,
0xd1, 0xd4, 0x40, 0xa0,

0x2c, 0x1e, 0x41, 0x7f, 0xf9, 0x18, 0x35, 0xff, 0x09, 0xdd, 0xb0, 0x4f,
0x61, 0xc3, 0x9f, 0x72,
0x25, 0x55, 0x5e, 0x5a, 0x39, 0xfc, 0xad, 0x14, 0x3a, 0x70, 0xb3, 0xaf,
0x0f, 0x2c, 0x89, 0x93,
0x10, 0xf1, 0x3e, 0x43, 0xdc, 0x5e, 0xfe, 0x61, 0xb2, 0x98, 0x51, 0x8f,
0xb0, 0x2b, 0x8a, 0x27,
0xb5, 0xee, 0xcd, 0x06, 0xf4, 0xad, 0xc0, 0x2b, 0xbf, 0xe7, 0xb4, 0xb0,
0x1a, 0x2e, 0x4d, 0x05,
0xef, 0x5f, 0xfd, 0xca, 0xa9, 0xde, 0xdc, 0x6a, 0xc7, 0x24, 0x63, 0xfa,
0x27, 0xf8, 0x44, 0xc8,
0xe7, 0xc9, 0x44, 0x2a, 0xcb, 0x9a, 0xac, 0xd1, 0xb0, 0x69, 0xad, 0xac,
0x93, 0x20, 0x71, 0x57,
0xbf, 0xa0, 0x29, 0xa0, 0xcc, 0xfc, 0x7d, 0x81, 0x58, 0xa6, 0x9b, 0x5c,
0x61, 0xb1, 0x0a, 0x8b,
0x55, 0x58, 0xac, 0xc2, 0x62, 0x15, 0x16, 0x8b, 0x9a, 0xf8, 0x82, 0x16,
0x22, 0x96, 0x2a, 0x2c,
0x56, 0x61, 0xb1, 0x0a, 0x8b, 0x25, 0xd3, 0x83, 0xc3, 0x62, 0x7c, 0xe7,
0xf2, 0x0e, 0xa0, 0xd8,
0x5a, 0xa7, 0x02, 0x62, 0x15, 0x10, 0x4b, 0x4a, 0xba, 0x02, 0x62, 0xea,
0x21, 0xad, 0x80, 0x58,
0x05, 0xc4, 0x2a, 0x20, 0x56, 0x01, 0xb1, 0x7b, 0x04, 0x62, 0xc5, 0x48,
0x6c, 0xb6, 0x69, 0x16,
0x51, 0xe6, 0xaf, 0xb9, 0xa7, 0x77, 0xf8, 0x21, 0xc8, 0xdb, 0xdf, 0x00,
0xe1, 0xb0, 0x61, 0xd1,
0xd9, 0x98, 0x60, 0x76, 0x6c, 0xd6, 0xfc, 0xe5, 0x43, 0x60, 0xb2, 0x3f,
0xfa, 0xc6, 0x0e, 0xd4,
0x8e, 0xd8, 0x01, 0x1d, 0x17, 0x2c, 0x2b, 0x14, 0x31, 0x6e, 0x7f, 0x1d,
0x9a, 0xb4, 0xe0, 0x14,
0xc4, 0x97, 0x34, 0x65, 0xff, 0x18, 0xc0, 0x31, 0x3c, 0x61, 0x76, 0x07,
0xd0, 0x71, 0xe3, 0x79,
0x05, 0x1d, 0x2b, 0xe8, 0x98, 0x94, 0x74, 0x05, 0x1d, 0xd5, 0x43, 0x5a,
0x41, 0xc7, 0x0a, 0x3a,
0x56, 0xd0, 0xb1, 0x82, 0x8e, 0x15, 0x74, 0xac, 0xa0, 0x63, 0x56, 0xe6,
0x83, 0x83, 0x8e, 0xe1,
0x37, 0x01, 0x2a, 0xe8, 0x58, 0x41, 0xc7, 0x0a, 0x3a, 0xe6, 0x8d, 0x53,
0x05, 0x1d, 0xcb, 0xcb,
0xa4, 0x82, 0x8e, 0xf, 0x68, 0x1d, 0x62, 0xa9, 0x82, 0x8e, 0x15, 0x74,
0xac, 0xa0, 0xa3, 0x22,
0x3d, 0xcc, 0x29, 0x3b, 0x95, 0xf9, 0x40, 0xaf, 0xc7, 0xde, 0x7d, 0x7b,
0x71, 0x7a, 0x7c, 0xfb,
0xcf, 0x8b, 0xc3, 0xbd, 0x2f, 0xf0, 0x7e, 0xec, 0xec, 0xef, 0xec, 0xf1,
0x6b, 0x50, 0x73, 0x1e,
0xf3, 0x29, 0xb0, 0xe0, 0xfd, 0xd8, 0xcf, 0x38, 0xbb, 0xff, 0xfd, 0x6d,
0xad, 0xf5, 0x08, 0x2d,
0xe9, 0x92, 0xec, 0xa3, 0xdb, 0x5f, 0x6d, 0x93, 0x12, 0x06, 0x5a, 0x47,
0xc0, 0xa7, 0x81, 0x0d,
0xb0, 0x85, 0xc8, 0xf, 0x88, 0xe5, 0x2e, 0x53, 0xfa, 0x1b, 0xb3, 0xfc,
0x19, 0x99, 0x59, 0xa4,

0x9f, 0xfc, 0xf2, 0x62, 0xd6, 0x20, 0xa4, 0x4a, 0x2d, 0x63, 0x2c, 0xb6,
0x84, 0xee, 0x3c, 0x2a,
0x3f, 0x0a, 0xa9, 0xdc, 0xac, 0x2b, 0x7a, 0x73, 0x87, 0xeb, 0x82, 0xd8,
0xec, 0x02, 0x1b, 0xe8,
0x0b, 0xce, 0xbf, 0x8d, 0x78, 0xf2, 0x47, 0x26, 0xe3, 0x1a, 0x93, 0xc9,
0x5c, 0xe2, 0x0f, 0x13,
0x6b, 0x3b, 0x2d, 0xf4, 0xdf, 0xff, 0xec, 0x95, 0x9f, 0xb2, 0xaf, 0x43,
0xed, 0xc1, 0x76, 0xdf,
0x84, 0xd5, 0x94, 0x94, 0x64, 0x0f, 0x94, 0x8e, 0x9b, 0xc9, 0x59, 0x79,
0x7b, 0x34, 0x3b, 0x67,
0x91, 0x5e, 0x97, 0xe4, 0xcc, 0x77, 0x4d, 0x2b, 0xd2, 0x9b, 0xa5, 0x72,
0xa7, 0xca, 0xbb, 0xef,
0xfb, 0x9a, 0xb3, 0xbf, 0xe7, 0xfa, 0x50, 0xee, 0x6b, 0xde, 0x9d, 0xe6,
0x30, 0xbe, 0xaf, 0x39,
0xfe, 0x6b, 0xad, 0x8b, 0x5f, 0xd8, 0x1c, 0x8c, 0x0d, 0x4c, 0xc9, 0x39,
0x54, 0x76, 0xbd, 0x7d,
0x4c, 0xf1, 0x7d, 0x5c, 0xc0, 0x33, 0x69, 0xad, 0xba, 0x7a, 0x27, 0x4e,
0xcb, 0xba, 0x7a, 0x47,
0xb2, 0x77, 0x91, 0x27, 0xf2, 0xd2, 0x72, 0xf1, 0xe4, 0x12, 0xd5, 0x86,
0x54, 0xe2, 0x09, 0x60,
0xd2, 0x97, 0xe6, 0x35, 0x31, 0x6a, 0xed, 0x22, 0x18, 0xce, 0x68, 0x47,
0xc6, 0x2a, 0x83, 0x70,
0xf4, 0x78, 0x42, 0xb5, 0xd0, 0xf3, 0xe6, 0x37, 0xe2, 0x48, 0x86, 0x26,
0x71, 0x07, 0x8e, 0x94,
0xff, 0xa2, 0xf8, 0xa6, 0x95, 0xcc, 0x6f, 0x29, 0xcb, 0x2b, 0x00, 0xb0,
0x06, 0xa3, 0x11, 0x86,
0x14, 0xd9, 0x37, 0x94, 0x25, 0x69, 0x3d, 0x45, 0x1a, 0x33, 0xf9, 0x45,
0x2e, 0x40, 0x66, 0x3b,
0xb1, 0x29, 0x4f, 0x37, 0x12, 0x8b, 0x8d, 0xb5, 0xf0, 0x68, 0x6e, 0xfa,
0x09, 0x83, 0x9c, 0x6e,
0x23, 0x21, 0xc4, 0xc2, 0x76, 0x96, 0xfa, 0xd5, 0xea, 0xe9, 0x2c, 0x70,
0xf3, 0x0e, 0xd9, 0x5f,
0x39, 0x04, 0x9c, 0x51, 0x83, 0x29, 0xda, 0xdb, 0xc9, 0x9a, 0x48, 0x4c,
0xe9, 0x3f, 0x04, 0xb7,
0xbf, 0xa2, 0x71, 0x40, 0x0c, 0xe2, 0x23, 0xb7, 0xcf, 0xbd, 0x6f, 0x76,
0x85, 0x19, 0xd4, 0x05,
0x8f, 0xd5, 0x10, 0x37, 0xef, 0xfa, 0xdc, 0x62, 0x10, 0x1e, 0x5b, 0x0c,
0x1c, 0x0c, 0xf3, 0x9f,
0x2d, 0xb1, 0xec, 0x13, 0x58, 0x2f, 0xd7, 0x5f, 0x05, 0xff, 0xdb, 0x76,
0xf9, 0x93, 0xdd, 0x37,
0x87, 0xea, 0xa6, 0xa6, 0x0d, 0x9d, 0xaa, 0x2f, 0xab, 0xfc, 0x2b, 0x3a,
0xaa, 0x47, 0xa9, 0x4b,
0xdb, 0xb3, 0xcd, 0x73, 0x75, 0x69, 0xfb, 0xef, 0x70, 0x69, 0xbb, 0x62,
0x38, 0xaa, 0x4b, 0xdb,
0xab, 0x4b, 0xdb, 0xab, 0x4b, 0xdb, 0xab, 0x4b, 0xdb, 0xef, 0xe3, 0xd2,
0x76, 0xf9, 0x1a, 0xe7,
0xcc, 0x18, 0xc8, 0xcc, 0xd7, 0x38, 0x17, 0x5e, 0x90, 0x54, 0xe6, 0x56,
0xe9, 0xa2, 0x68, 0xc0,


```

    0x32, 0xb8, 0x52, 0xde, 0xa0, 0xf4, 0x85, 0xdd, 0x31, 0x9d, 0xe9, 0x31,
    0x56, 0x77, 0x4c, 0xe7,
    0x92, 0xf8, 0x23, 0xdf, 0x31, 0x3d, 0x3b, 0xaa, 0x90, 0x49, 0xde, 0xf9,
    0x1d, 0xd3, 0x77, 0xcd,
    0x5e, 0x75, 0xc7, 0xb4, 0x48, 0xd5, 0x1d, 0xd3, 0x7f, 0xc8, 0x3b, 0xa6,
    0x7f, 0xdf, 0x63, 0x45,
    0x99, 0x8b, 0x77, 0x75, 0xad, 0x44, 0x75, 0xae, 0xa8, 0x3a, 0x57, 0x54,
    0x62, 0x1f, 0xb9, 0x3a,
    0x57, 0x54, 0x4a, 0x26, 0xd5, 0xb9, 0xa2, 0x07, 0x74, 0x48, 0x81, 0xa5,
    0xea, 0x5c, 0x51, 0x75,
    0xae, 0xa8, 0xba, 0x56, 0x22, 0x99, 0x1e, 0x1c, 0x16, 0x4b, 0x84, 0x2c,
    0xee, 0x06, 0x92, 0x55,
    0x88, 0xac, 0x42, 0x64, 0x49, 0x49, 0x57, 0x88, 0x4c, 0x3d, 0xa4, 0x15,
    0x22, 0xab, 0x10, 0x59,
    0x85, 0xc8, 0x2a, 0x44, 0x56, 0x21, 0xb2, 0x07, 0x71, 0x72, 0xba, 0xdb,
    0xd4, 0xd9, 0xf1, 0x0a,
    0x0f, 0x3e, 0x46, 0xd9, 0xdd, 0x66, 0xdf, 0x35, 0x6e, 0xf8, 0x9b, 0x11,
    0xb5, 0xad, 0x9d, 0xff,
    0x01, 0xec, 0x97, 0x55, 0x65, 0x0d, 0xb3, 0x00, 0x00
};
const size_t mainpageSize = sizeof(mainpage);

// Pump calibration page
static const byte calibrarbomba[] PROGMEM = {
    0x1f, 0x8b, 0x08, 0x08, 0x1f, 0x46, 0x39, 0x64, 0x00, 0x03, 0x63, 0x61,
    0x6c, 0x69, 0x62, 0x72,
    0x61, 0x72, 0x62, 0x6f, 0x6d, 0x62, 0x61, 0x2e, 0x68, 0x74, 0x6d, 0x6c,
    0x00, 0xed, 0x5b, 0x5b,
    0x73, 0xdb, 0x36, 0x16, 0x7e, 0xef, 0x4c, 0xff, 0x03, 0xc2, 0xec, 0x46,
    0x72, 0x6a, 0x5d, 0x6d,
    0x39, 0x8e, 0x2d, 0xb9, 0xa3, 0xd8, 0x6e, 0xe2, 0x8e, 0x2f, 0x19, 0x5b,
    0x49, 0x37, 0xdb, 0xd9,
    0xd9, 0x42, 0x24, 0x24, 0x21, 0x21, 0x09, 0x06, 0x00, 0x25, 0x3b, 0x69,
    0x7e, 0x4c, 0x1e, 0xfb,
    0xb8, 0xd3, 0xb7, 0x7d, 0xf5, 0x1f, 0xdb, 0x03, 0xf0, 0x22, 0x92, 0x22,
    0x29, 0x39, 0x71, 0x3c,
    0x3b, 0xbb, 0xc6, 0xd4, 0x21, 0x85, 0xcb, 0xc1, 0xb9, 0x7c, 0xe7, 0xe0,
    0x1c, 0x48, 0xed, 0x3e,
    0x38, 0x38, 0xdb, 0x1f, 0xbc, 0x79, 0x79, 0x88, 0x26, 0xd2, 0xb1, 0xf7,
    0xbe, 0xff, 0xae, 0x1b,
    0x3f, 0x09, 0xb6, 0xe0, 0x89, 0xa0, 0x75, 0x25, 0x95, 0x36, 0xd9, 0x3b,
    0x3f, 0x3a, 0x7c, 0x7e,
    0x86, 0xce, 0x0f, 0x4f, 0xce, 0x06, 0x67, 0xdd, 0x46, 0xd0, 0x17, 0x4e,
    0xb0, 0xa9, 0xfb, 0x0e,
    0x71, 0x62, 0xf7, 0x0c, 0x31, 0x61, 0x5c, 0x9a, 0xbe, 0x44, 0xd4, 0x64,
    0xae, 0x81, 0x26, 0x9c,
    0x8c, 0x7a, 0xc6, 0x44, 0x4a, 0x4f, 0xec, 0x34, 0x1a, 0xc4, 0x35, 0xf9,
    0x95, 0x27, 0x89, 0x55,

```


0x93, 0x43, 0xb7, 0x59, 0x1f, 0x0b, 0x89, 0x25, 0x35, 0xeb, 0x26, 0x73,
0x1a, 0xd4, 0xc1, 0x63,
0x22, 0x7e, 0x7c, 0xdf, 0x83, 0x91, 0x9d, 0xfe, 0xa9, 0xf5, 0xf4, 0xb9,
0x39, 0xb8, 0x7c, 0xf3,
0x7c, 0xfb, 0xfc, 0xfd, 0xdb, 0x93, 0xf3, 0xed, 0xa1, 0x45, 0x66, 0xfb,
0xfe, 0xd6, 0xdf, 0x5f,
0x3b, 0xb5, 0x93, 0x17, 0xe7, 0xfb, 0xbf, 0x0c, 0x46, 0x92, 0x8d, 0x0f,
0xac, 0x76, 0xff, 0xe7,
0x96, 0x3f, 0x6b, 0x1e, 0x3f, 0x3b, 0xe9, 0x9c, 0xfe, 0xf2, 0x4c, 0x9a,
0x67, 0x13, 0xdc, 0xef,
0x63, 0xa7, 0x6f, 0x8f, 0xc5, 0xdb, 0xe3, 0x8b, 0x9f, 0xfe, 0xe9, 0x6d,
0x8e, 0x1f, 0xf9, 0xe2,
0xbd, 0xd7, 0xdb, 0xef, 0xbf, 0x32, 0x22, 0x5e, 0x1d, 0x22, 0x31, 0x52,
0x1c, 0xd5, 0xc8, 0x7b,
0x9f, 0x4e, 0x7b, 0xc6, 0x3e, 0x73, 0x25, 0x71, 0x65, 0x6d, 0x70, 0xe5,
0x11, 0x03, 0x99, 0xc1,
0xa7, 0x9e, 0x21, 0xc9, 0xa5, 0x6c, 0x28, 0x6d, 0xec, 0x9a, 0x13, 0xcc,
0x05, 0x91, 0xbd, 0x57,
0x83, 0x9f, 0x6a, 0xdb, 0x69, 0x3a, 0x2e, 0x76, 0x48, 0xcf, 0x98, 0x52,
0x32, 0xf3, 0x40, 0xee,
0xc4, 0xea, 0x19, 0xb5, 0xe4, 0xa4, 0x67, 0x91, 0x29, 0x35, 0x49, 0x4d,
0x7f, 0x58, 0x47, 0xd4,
0xa5, 0x92, 0x62, 0xbb, 0x26, 0x4c, 0x6c, 0x93, 0x5e, 0x2b, 0xa6, 0x24,
0xe4, 0x95, 0x4d, 0x90,
0x84, 0xed, 0xc3, 0x5d, 0x4d, 0x21, 0xa2, 0xc1, 0xa0, 0x3d, 0xfe, 0x18,
0x7c, 0xf2, 0xb0, 0x65,
0x51, 0x77, 0xbc, 0x83, 0x9a, 0xde, 0xe5, 0x2e, 0x72, 0x30, 0x1f, 0x53,
0x37, 0xf8, 0x10, 0x8c,
0x7f, 0xfa, 0xfe, 0xbb, 0xe0, 0x65, 0xc8, 0xac, 0x2b, 0xf4, 0x71, 0x4e,
0x62, 0x04, 0x7c, 0xd5,
0x46, 0xd8, 0xa1, 0xf6, 0xd5, 0x0e, 0xaa, 0x5c, 0x90, 0x31, 0x23, 0xe8,
0xd5, 0x51, 0x65, 0x1d,
0x0d, 0xf0, 0x84, 0x39, 0x78, 0x1d, 0x3d, 0x27, 0x2e, 0x99, 0xc2, 0xf3,
0x35, 0xe1, 0x16, 0x76,
0xe1, 0x45, 0x60, 0x57, 0xd4, 0x04, 0xe1, 0x74, 0x94, 0xa5, 0x3d, 0x62,
0xdc, 0x49, 0xd2, 0xb6,
0xa8, 0xf0, 0x6c, 0x0c, 0x74, 0x47, 0x36, 0x89, 0x18, 0xd1, 0xf3, 0xe0,
0x63, 0xcd, 0xa2, 0x9c,
0x98, 0x92, 0x32, 0x60, 0xd3, 0x64, 0xb6, 0xef, 0xb8, 0x89, 0x09, 0x0e,
0xbe, 0x0c, 0x54, 0xb3,
0x83, 0x36, 0x9a, 0x73, 0x21, 0x82, 0xa1, 0x40, 0x34, 0xec, 0x4b, 0x96,
0xdd, 0xbf, 0x4e, 0x5d,
0xcf, 0x97, 0x35, 0xa5, 0x6a, 0x4c, 0x5d, 0xc2, 0x57, 0xe1, 0xe5, 0xad,
0x2f, 0x24, 0x1d, 0x5d,
0xd5, 0x42, 0xfb, 0x00, 0x33, 0xf0, 0x2f, 0xe1, 0x89, 0x19, 0xd8, 0xa6,
0x63, 0xb7, 0x46, 0x25,
0x71, 0x44, 0xce, 0x68, 0xc0, 0x4f, 0x6d, 0xc8, 0xa4, 0x64, 0xce, 0x0e,
0x6a, 0xa5, 0xb9, 0xf5,

0x98, 0xa0, 0x81, 0x8c, 0xe0, 0x03, 0x00, 0xe9, 0x29, 0xc9, 0xf2, 0x6c,
0xe3, 0x21, 0xb1, 0x7f,
0x0d, 0x4c, 0xec, 0x03, 0x0c, 0x8c, 0x7f, 0x24, 0xb9, 0x9e, 0xaf, 0xc7,
0x43, 0x01, 0x5a, 0x92,
0x0b, 0xeb, 0xb5, 0xc8, 0xbf, 0xce, 0x21, 0x92, 0x5e, 0xaf, 0x7a, 0x6a,
0x5a, 0x80, 0x1c, 0xd6,
0x87, 0x8c, 0x5b, 0x84, 0xef, 0x20, 0x97, 0xb9, 0x24, 0xd1, 0xcd, 0x7c,
0x09, 0x5e, 0x4b, 0x16,
0xfa, 0x63, 0x88, 0x6d, 0xa7, 0x44, 0x0c, 0xa8, 0xd4, 0x38, 0xb6, 0xa8,
0x0f, 0x0a, 0xda, 0xcc,
0x0c, 0x5e, 0xd6, 0xc4, 0x04, 0x5b, 0x6c, 0xb6, 0x03, 0x9c, 0x82, 0xab,
0xa0, 0x26, 0x6a, 0x79,
0x97, 0x68, 0x03, 0xfe, 0xf8, 0x78, 0x88, 0xab, 0xcd, 0x75, 0x14, 0xfe,
0x57, 0x6f, 0xad, 0xed,
0x66, 0x60, 0x29, 0xe8, 0x07, 0x60, 0xa3, 0xb5, 0x95, 0x22, 0x29, 0x39,
0xe0, 0x2f, 0x54, 0xca,
0x9c, 0x3c, 0xac, 0xdf, 0x10, 0x89, 0x59, 0x21, 0x7a, 0x5a, 0xcd, 0xe6,
0x5f, 0xf3, 0x71, 0xd5,
0xea, 0x68, 0x4f, 0x69, 0x3c, 0x46, 0x7d, 0x05, 0x01, 0xcc, 0x11, 0xb1,
0x91, 0xc4, 0x0e, 0xbe,
0xfe, 0x17, 0x43, 0xce, 0xf5, 0xe7, 0x4b, 0xea, 0x30, 0x64, 0x11, 0x64,
0x33, 0x11, 0xe8, 0x58,
0xa0, 0xc7, 0x8d, 0xa5, 0xba, 0xdf, 0x19, 0x31, 0xd3, 0x17, 0x49, 0x0b,
0xdc, 0x44, 0x03, 0x1b,
0x6b, 0x65, 0xd6, 0x15, 0xfe, 0xd0, 0xc9, 0xe2, 0x63, 0x88, 0xcd, 0x77,
0x63, 0xce, 0x7c, 0xd7,
0x02, 0x00, 0xdb, 0x0c, 0x8c, 0xf9, 0xb0, 0xb9, 0xb5, 0x61, 0x6d, 0x59,
0x09, 0xa1, 0xc3, 0x81,
0xd9, 0x84, 0x4a, 0xb2, 0xdc, 0xfa, 0xb1, 0x95, 0x15, 0x92, 0x51, 0xbb,
0xb9, 0xba, 0xad, 0x93,
0x26, 0x7b, 0x92, 0x1a, 0x31, 0x7d, 0x2e, 0x14, 0x0b, 0x1e, 0xa3, 0xb9,
0xde, 0x93, 0xf2, 0xe6,
0x05, 0x1b, 0x67, 0x44, 0x4c, 0x5a, 0xba, 0x54, 0x4f, 0x3b, 0x13, 0x36,
0x4d, 0xc7, 0x80, 0x7c,
0x6d, 0x6d, 0x76, 0xcc, 0x2c, 0x39, 0x15, 0x0c, 0x38, 0xb3, 0xf5, 0x49,
0x16, 0xaf, 0x4f, 0xc8,
0xd7, 0x4e, 0x4a, 0x1e, 0x91, 0x1a, 0x8d, 0x46, 0xbb, 0xa9, 0xb9, 0x33,
0x42, 0xc7, 0x13, 0xa9,
0x50, 0x6a, 0x5b, 0xf3, 0x1d, 0xd4, 0xa3, 0x0e, 0xa1, 0x75, 0x4e, 0x38,
0x19, 0xbb, 0xf5, 0x5f,
0x2b, 0x7c, 0xd9, 0xcd, 0xe3, 0x78, 0xa4, 0x5b, 0x96, 0xe3, 0x87, 0xea,
0x54, 0x4e, 0x4a, 0xbb,
0x3c, 0xf2, 0x44, 0x4b, 0xea, 0x2e, 0x9e, 0xc6, 0xcb, 0x34, 0x5c, 0xe9,
0x07, 0xcd, 0x4e, 0x68,
0x6d, 0xe8, 0xda, 0x9d, 0x87, 0x4f, 0xea, 0xaa, 0xe0, 0x50, 0x1b, 0xda,
0xcc, 0x7c, 0xb7, 0xbb,
0x20, 0x41, 0x6b, 0x05, 0x11, 0xb2, 0x08, 0x4d, 0xba, 0x2a, 0x9a, 0x84,
0x3a, 0x4b, 0x01, 0xe2,

0x06, 0xb2, 0xa8, 0xe4, 0x22, 0x4f, 0x09, 0x99, 0xf0, 0x09, 0x10, 0x63,
0xde, 0x0e, 0xea, 0x24,
0xa2, 0x83, 0xc6, 0x9c, 0x3a, 0xc1, 0x76, 0x82, 0x57, 0xd8, 0x88, 0xbc,
0xa9, 0xd6, 0x60, 0xca,
0x3c, 0x2e, 0xd9, 0x64, 0x24, 0x55, 0xe8, 0x48, 0x58, 0x3f, 0x21, 0x5e,
0x80, 0x8d, 0xf0, 0xf4,
0xc9, 0x9b, 0xc0, 0x89, 0x47, 0xb0, 0x54, 0x2e, 0x17, 0xbe, 0x66, 0x75,
0x10, 0xc4, 0xa4, 0x48,
0x07, 0x9d, 0x9c, 0xe3, 0x3b, 0x25, 0xa8, 0xf0, 0xb0, 0x49, 0xbe, 0xb9,
0xb4, 0xcd, 0x6f, 0x25,
0xed, 0x56, 0x4a, 0xda, 0xad, 0xd4, 0x3e, 0xe9, 0x40, 0x53, 0xa4, 0xf1,
0x9c, 0xd0, 0x36, 0x4f,
0x06, 0x42, 0x27, 0xbe, 0x2d, 0x68, 0xc7, 0xc8, 0xee, 0x94, 0x38, 0xa7,
0xa9, 0x5b, 0x46, 0x0c,
0x1d, 0x28, 0x11, 0x58, 0x84, 0x5a, 0x71, 0x78, 0xce, 0x4a, 0x98, 0x0a,
0xb4, 0xa1, 0x7e, 0xb4,
0x0f, 0xa4, 0x3d, 0x22, 0x0e, 0x98, 0x1b, 0x29, 0x6c, 0x24, 0xe5, 0xd5,
0x41, 0x4b, 0x23, 0x23,
0x96, 0xfc, 0x86, 0xce, 0x9b, 0x2b, 0x59, 0xc6, 0x69, 0x73, 0x0c, 0x14,
0x6a, 0x06, 0x35, 0x57,
0xf6, 0xec, 0x88, 0x73, 0xe7, 0x4a, 0xb1, 0x85, 0x3e, 0x66, 0xd5, 0xd6,
0x5e, 0x50, 0x5b, 0x6a,
0xdd, 0x43, 0xd9, 0x4a, 0x07, 0xe7, 0x9c, 0x80, 0x9b, 0x9b, 0x02, 0x25,
0x43, 0x79, 0xe0, 0x72,
0xf9, 0x40, 0x0a, 0x36, 0x69, 0x7f, 0xfd, 0x26, 0xed, 0xa5, 0x9b, 0x6c,
0xac, 0xb0, 0x49, 0x11,
0xc1, 0x7c, 0xe5, 0xa8, 0xba, 0xe9, 0x63, 0xc6, 0x12, 0xed, 0x66, 0xca,
0xe5, 0xc2, 0x8f, 0x31,
0x00, 0xda, 0x9d, 0x2c, 0xa8, 0x1e, 0x9a, 0x8e, 0xf5, 0x4d, 0xd3, 0x8d,
0x45, 0x69, 0x17, 0xc4,
0xfc, 0xb2, 0xdc, 0x24, 0x33, 0x9a, 0xce, 0xfa, 0x72, 0x22, 0xac, 0x66,
0xbd, 0x30, 0x4f, 0xb9,
0x41, 0x42, 0x12, 0xa9, 0x2d, 0x48, 0x3f, 0xe6, 0x98, 0x5e, 0x9e, 0x7b,
0xa8, 0x47, 0xb7, 0xa1,
0x6b, 0x3e, 0x55, 0x67, 0x37, 0xc2, 0x42, 0xbb, 0xab, 0x4a, 0x36, 0xf5,
0xb4, 0xe8, 0x14, 0x51,
0x0b, 0x8a, 0x66, 0x7d, 0x04, 0xc4, 0x45, 0xa2, 0xea, 0x36, 0x6d, 0x2c,
0x44, 0xcf, 0x80, 0x93,
0x3c, 0x59, 0x1e, 0x76, 0x27, 0x2d, 0xb5, 0xa0, 0x22, 0x5b, 0x95, 0xbd,
0xfd, 0xfe, 0xf1, 0xd1,
0xb3, 0xf3, 0xfe, 0xc1, 0x19, 0x3a, 0x38, 0x44, 0xcf, 0xce, 0x4e, 0x9e,
0xf5, 0x61, 0x83, 0x56,
0x72, 0xb2, 0x17, 0xcc, 0x6d, 0x57, 0xf6, 0x0e, 0x88, 0x80, 0xcc, 0x97,
0x40, 0x56, 0x8c, 0xbc,
0xeb, 0xcf, 0x10, 0x6f, 0xe0, 0xc9, 0x2c, 0x7e, 0xfd, 0x59, 0x20, 0x28,
0x50, 0xe9, 0x90, 0x07,
0xd9, 0xb2, 0x89, 0x7d, 0x0b, 0xdb, 0x2a, 0x49, 0x96, 0x3e, 0xa8, 0xdf,
0x19, 0xe2, 0x6e, 0xc3,

0x4b, 0x52, 0xc4, 0x61, 0x95, 0xdf, 0x48, 0xd7, 0xac, 0x69, 0x9e, 0xe3,
0x83, 0x6c, 0x61, 0x52,
0x03, 0x66, 0x65, 0xfb, 0xa8, 0x33, 0x4e, 0x2e, 0x34, 0x90, 0xe0, 0xe6,
0xfc, 0x1a, 0xc1, 0xb4,
0xa0, 0x50, 0x53, 0xf4, 0x6a, 0x9e, 0x3b, 0xae, 0x8f, 0x54, 0xae, 0x00,
0x9f, 0xf4, 0x35, 0x42,
0xa7, 0xd5, 0x6e, 0x6c, 0x6f, 0x35, 0xd5, 0xdf, 0x76, 0xbb, 0x53, 0x87,
0x71, 0x03, 0x2a, 0x3b,
0xa8, 0xc7, 0x4f, 0xe0, 0xf8, 0x02, 0x64, 0x8d, 0x53, 0xfb, 0x77, 0x1b,
0x38, 0x52, 0x6f, 0xc8,
0x45, 0xfc, 0x4c, 0xb0, 0xae, 0xb2, 0xb8, 0xd8, 0x0c, 0x81, 0xdb, 0x27,
0x69, 0x24, 0x66, 0x86,
0xb1, 0x79, 0x41, 0x44, 0x10, 0xdc, 0xcd, 0xcc, 0x99, 0xc7, 0x6f, 0x10,
0x4e, 0x61, 0xa1, 0x67,
0x64, 0x6b, 0x4c, 0xe5, 0xa6, 0xc6, 0x5e, 0x37, 0xb9, 0x60, 0x6e, 0xdf,
0x6e, 0x23, 0xd5, 0x0f,
0x80, 0x82, 0x3d, 0xb2, 0xfb, 0xea, 0x4a, 0x3d, 0x9f, 0xba, 0x76, 0x1c,
0x03, 0x39, 0x44, 0x4e,
0x18, 0x80, 0x0d, 0x92, 0x0a, 0x09, 0x9a, 0xd2, 0x95, 0x3a, 0x58, 0xd2,
0xf7, 0x2c, 0xc8, 0x14,
0xf6, 0x03, 0x14, 0xa8, 0xbe, 0xac, 0x48, 0x9a, 0xbc, 0x2e, 0x6a, 0x03,
0x38, 0x6d, 0x54, 0xd4,
0xb5, 0x00, 0x48, 0x37, 0x5f, 0xf2, 0x5a, 0x95, 0xfb, 0xa0, 0xb8, 0xe0,
0xe9, 0x76, 0x1b, 0x7a,
0x7a, 0x1e, 0x9d, 0x24, 0x4c, 0xd2, 0xa5, 0x7d, 0x56, 0x37, 0x3a, 0xc1,
0x79, 0x32, 0xbf, 0xfb,
0xc8, 0x2a, 0x6b, 0x91, 0x78, 0x00, 0x27, 0x45, 0x35, 0x71, 0xc1, 0x62,
0x84, 0xf7, 0x36, 0x8b,
0xdc, 0x6a, 0xcf, 0xcb, 0xe9, 0x9e, 0x62, 0xdb, 0x87, 0x05, 0x85, 0x3b,
0xa4, 0x55, 0x91, 0xa8,
0xf3, 0x23, 0x01, 0x78, 0x18, 0x8b, 0x35, 0xf3, 0x45, 0xaa, 0x72, 0x8e,
0x4b, 0xb4, 0x94, 0xe3,
0x27, 0x2b, 0x58, 0x61, 0x40, 0x15, 0xe1, 0x01, 0x25, 0x8e, 0xc7, 0xbe,
0xad, 0x09, 0x36, 0xbe,
0xd6, 0x04, 0x9a, 0xd5, 0xac, 0x01, 0x82, 0xce, 0xdb, 0x52, 0xff, 0x66,
0x27, 0x4f, 0xfd, 0x81,
0x8e, 0xc4, 0x17, 0xe8, 0x3e, 0x29, 0x55, 0x58, 0x90, 0x46, 0xbc, 0x3e,
0xf7, 0x31, 0xb7, 0x20,
0x82, 0x86, 0xfb, 0x58, 0xcc, 0x40, 0xcc, 0x35, 0x6d, 0x6a, 0xbe, 0x83,
0xa9, 0xc4, 0xb5, 0x12,
0xce, 0x55, 0x25, 0x53, 0x88, 0x2a, 0x6b, 0x8b, 0x91, 0x51, 0xf9, 0xef,
0x42, 0x2c, 0x31, 0x39,
0xf5, 0x64, 0x0e, 0x33, 0x23, 0xdf, 0xd5, 0xee, 0x8b, 0xf2, 0xa9, 0x27,
0x2b, 0xe4, 0x64, 0xd3,
0xa3, 0x75, 0x8f, 0xeb, 0xe7, 0x01, 0x19, 0x61, 0xdf, 0x96, 0xd5, 0xe4,
0x1d, 0x4d, 0xb2, 0x4d,
0x41, 0xa0, 0xcb, 0x09, 0x47, 0x3d, 0xe4, 0x92, 0x19, 0xfa, 0xdb, 0xc9,
0xf1, 0x0b, 0x08, 0xc9,

0xe7, 0xe4, 0xbd, 0x0f, 0xa7, 0x48, 0xe9, 0x22, 0x09, 0x2a, 0x66, 0xa0,
0xaa, 0x1e, 0x1a, 0x61,
0x5b, 0x90, 0xe5, 0x33, 0x0f, 0xfc, 0x80, 0x7b, 0x58, 0xb1, 0xd1, 0x6c,
0x36, 0x77, 0x51, 0xa3,
0x81, 0x36, 0x40, 0xb4, 0x31, 0x1c, 0xad, 0x4c, 0x2c, 0x5d, 0x7e, 0x42,
0x84, 0x80, 0x60, 0x0f,
0xab, 0x8d, 0x53, 0x06, 0xcb, 0xd0, 0x04, 0x43, 0x55, 0x69, 0xd2, 0x21,
0xb5, 0x18, 0xbc, 0x08,
0xcf, 0xd7, 0xe7, 0x9e, 0x05, 0xa8, 0x11, 0x84, 0x4f, 0xa1, 0x97, 0xd7,
0xd1, 0x4b, 0x38, 0xe5,
0x47, 0x78, 0xca, 0xb8, 0xba, 0xaa, 0x95, 0xd7, 0x7f, 0x80, 0x07, 0xd8,
0xfa, 0x76, 0xc8, 0xf5,
0xc9, 0x94, 0xd5, 0x8d, 0xdd, 0xa8, 0xe8, 0xc8, 0x36, 0x50, 0x49, 0x9d,
0x79, 0xc4, 0xad, 0x1a,
0x2f, 0xcf, 0x2e, 0x06, 0xc6, 0x3a, 0xca, 0x8b, 0xa0, 0xeb, 0x90, 0x5e,
0xf8, 0xa4, 0x48, 0x49,
0x8a, 0x84, 0x20, 0x32, 0xd4, 0xe5, 0x0b, 0x7d, 0xf6, 0x57, 0xd3, 0x57,
0xd3, 0x40, 0x16, 0x7b,
0x1e, 0x20, 0x48, 0xd3, 0x6b, 0x5c, 0xd6, 0x66, 0xb3, 0x59, 0x4d, 0x41,
0xa4, 0xe6, 0x73, 0x9b,
0xb8, 0x26, 0xb3, 0x88, 0x65, 0xac, 0x15, 0x32, 0x09, 0x0a, 0x3c, 0x04,
0x99, 0x87, 0x36, 0x31,
0x49, 0x70, 0x0d, 0xa6, 0x83, 0x02, 0xd2, 0x39, 0x80, 0x47, 0x38, 0x2e,
0x57, 0xaa, 0x32, 0x3a,
0x30, 0x38, 0x08, 0xf4, 0x5b, 0x8d, 0xd1, 0x56, 0x2d, 0x44, 0x96, 0x6a,
0x73, 0xbb, 0x2b, 0xe1,
0x0b, 0x64, 0x8f, 0xe4, 0xc7, 0x90, 0xdc, 0x29, 0x18, 0x29, 0x56, 0xfb,
0xea, 0x1d, 0xb6, 0xb6,
0xb1, 0xae, 0x0d, 0x4c, 0x2a, 0x7d, 0xab, 0x78, 0x35, 0xb6, 0x09, 0xac,
0x4c, 0xdb, 0xbe, 0x48,
0xd3, 0x9f, 0xd6, 0xb3, 0x18, 0x2b, 0xd6, 0x99, 0x36, 0xac, 0xcb, 0xc1,
0x1a, 0x57, 0xea, 0xbb,
0x0a, 0x62, 0x4e, 0xb0, 0xab, 0x51, 0xb5, 0xa2, 0xf8, 0x74, 0x84, 0xaa,
0x8a, 0x86, 0xa6, 0x70,
0xa1, 0x28, 0xa0, 0x5e, 0xaf, 0x97, 0x71, 0x9c, 0xfa, 0xc1, 0xd9, 0xe9,
0x61, 0x29, 0x19, 0xd5,
0x4c, 0x9b, 0x60, 0x1e, 0x69, 0x5f, 0x1b, 0x24, 0x50, 0xd4, 0x31, 0x75,
0x3c, 0x1a, 0xde, 0x6b,
0x2a, 0x7b, 0x72, 0xfa, 0x01, 0x82, 0x0d, 0x47, 0x82, 0x2a, 0xd8, 0x6b,
0xcc, 0x13, 0xa5, 0xc5,
0x18, 0xf4, 0xe5, 0xdb, 0x28, 0x8e, 0x1f, 0x44, 0x56, 0x7b, 0xf4, 0x28,
0xc0, 0x25, 0x30, 0xee,
0x0b, 0xcd, 0x39, 0xd4, 0x12, 0x4b, 0x39, 0x55, 0x0d, 0x18, 0x7b, 0x81,
0x5d, 0xcb, 0x26, 0x7a,
0x5f, 0x48, 0xd5, 0x08, 0x1a, 0x71, 0xe6, 0xe8, 0xfc, 0x18, 0xc2, 0x13,
0xa4, 0xe6, 0x9c, 0x2c,
0xa7, 0xa2, 0xb0, 0x17, 0xad, 0x1f, 0xc0, 0xa1, 0x01, 0x8a, 0x0f, 0xb4,
0x39, 0xef, 0x2a, 0x81,

0x54, 0xd4, 0x02, 0x70, 0x24, 0x17, 0x15, 0x41, 0x23, 0x6a, 0x9f, 0x8a,
0x87, 0x0b, 0x86, 0x3e,
0x15, 0x42, 0x48, 0xc9, 0xe0, 0x61, 0x8e, 0x1d, 0xa1, 0x82, 0xd1, 0xc2,
0xa1, 0xdf, 0x33, 0xd0,
0x0f, 0x28, 0x70, 0xde, 0x57, 0xe7, 0x47, 0xfb, 0x0c, 0x2c, 0xe8, 0x82,
0xc7, 0x57, 0x2d, 0x66,
0xaa, 0xac, 0x49, 0xd6, 0xc7, 0x44, 0x1e, 0xda, 0x44, 0xbd, 0x3e, 0xbb,
0x3a, 0xb2, 0xaa, 0x39,
0x69, 0xc3, 0x5a, 0x5d, 0x1f, 0x39, 0x6b, 0xe8, 0x87, 0x62, 0xb6, 0x8d,
0x47, 0x99, 0xf3, 0xee,
0x6b, 0xf6, 0xd5, 0xe7, 0x65, 0xb4, 0x6b, 0x69, 0x3c, 0x73, 0xad, 0x6a,
0x20, 0x7b, 0xde, 0xb4,
0x1c, 0x55, 0xce, 0x28, 0x44, 0xf7, 0x19, 0xb8, 0x9c, 0xcd, 0xb0, 0x95,
0x70, 0xb3, 0x62, 0x2f,
0xfb, 0xa2, 0x13, 0x69, 0x1e, 0xaf, 0x9f, 0x1f, 0x06, 0xe1, 0x1a, 0xa4,
0xbd, 0x59, 0xac, 0x2e,
0x8d, 0x0a, 0xb7, 0x15, 0x14, 0xbe, 0xd0, 0x01, 0x93, 0x6e, 0x03, 0x5c,
0xfd, 0x7c, 0x71, 0x76,
0x5a, 0xf7, 0xd4, 0x57, 0x98, 0xd5, 0xac, 0xf7, 0x2c, 0x73, 0x84, 0x9b,
0xa3, 0x10, 0xf6, 0x8b,
0x36, 0xa8, 0x2f, 0x4c, 0xba, 0x85, 0xdd, 0x92, 0xd8, 0x2b, 0xd8, 0x4b,
0x4d, 0x29, 0xd9, 0xa9,
0xc4, 0x83, 0xf3, 0xfa, 0x63, 0x20, 0xe7, 0x42, 0x38, 0xd3, 0x07, 0x65,
0x58, 0x5e, 0x7a, 0xd6,
0x05, 0x80, 0x86, 0xd9, 0x75, 0x70, 0x09, 0x06, 0x65, 0xdd, 0x90, 0xa3,
0xc6, 0x1e, 0x7a, 0x44,
0xa0, 0x2a, 0xce, 0x12, 0x19, 0xfa, 0x90, 0x51, 0xbb, 0x3a, 0x9d, 0x35,
0x1d, 0x2b, 0xca, 0x67,
0x83, 0xde, 0x44, 0x16, 0x99, 0x10, 0xb9, 0x5a, 0x69, 0x55, 0x20, 0x7f,
0xbc, 0x18, 0xf4, 0xcf,
0x07, 0xdd, 0x46, 0x30, 0xf3, 0x16, 0x89, 0x37, 0x03, 0xe2, 0x67, 0x2f,
0x97, 0xd0, 0x5e, 0x9e,
0x9b, 0x26, 0xc9, 0x86, 0x51, 0xeb, 0x36, 0x1d, 0xfb, 0x7f, 0x3f, 0xd5,
0xbc, 0xcf, 0xe2, 0x82,
0x76, 0x93, 0x2c, 0x4e, 0x05, 0xdc, 0x30, 0x60, 0xf4, 0x50, 0xab, 0x54,
0xd2, 0xc5, 0x93, 0x41,
0x7d, 0x69, 0x7c, 0x83, 0xb3, 0x21, 0x26, 0xf2, 0x15, 0x59, 0x63, 0xc4,
0xf3, 0x2d, 0x65, 0x8e,
0xaa, 0xdd, 0x51, 0xf6, 0x58, 0x96, 0x38, 0x2d, 0x51, 0x58, 0x61, 0x8c,
0x2d, 0xa1, 0x9b, 0x32,
0x6c, 0xf9, 0xb9, 0x98, 0x67, 0x58, 0xe6, 0xdd, 0xdb, 0x75, 0x35, 0xbb,
0x46, 0x9c, 0xdf, 0x42,
0x65, 0xa0, 0xda, 0xed, 0x25, 0x28, 0x51, 0xbb, 0x9b, 0xd4, 0x21, 0x6a,
0x25, 0x30, 0x5f, 0x32,
0x7c, 0xdb, 0x5e, 0x90, 0xd3, 0x15, 0xab, 0x02, 0x5b, 0xd6, 0xa1, 0xba,
0xd4, 0x39, 0xa6, 0x42,
0x12, 0x97, 0xf0, 0x6a, 0xe5, 0xe0, 0xec, 0x24, 0xbc, 0x51, 0x38, 0x86,
0xfc, 0x9a, 0x58, 0x95,

0xf5, 0x55, 0xa0, 0xab, 0x6e, 0xda, 0x25, 0x1c, 0x4c, 0xce, 0x90, 0xf0,
0x23, 0x7d, 0xe1, 0xd5,
0x2b, 0xd4, 0x77, 0x65, 0x21, 0xe7, 0xab, 0x14, 0x49, 0x13, 0x90, 0x75,
0xa8, 0xfb, 0x3a, 0x34,
0x48, 0xb3, 0x7c, 0x22, 0xbe, 0x8c, 0x26, 0xb6, 0xd5, 0x29, 0x5d, 0x14,
0xec, 0x13, 0x7c, 0xe6,
0x68, 0xe0, 0x1d, 0xb9, 0xf2, 0xc0, 0xf0, 0x22, 0x25, 0x79, 0xf9, 0x05,
0xd9, 0x9c, 0x05, 0x58,
0xbc, 0x0f, 0xf5, 0x12, 0x70, 0x10, 0x5c, 0x9a, 0xcd, 0x26, 0xd4, 0x9c,
0x94, 0x18, 0x54, 0x79,
0x4c, 0xb4, 0xa6, 0x8b, 0x36, 0xb7, 0xd1, 0xef, 0xbf, 0xc7, 0x34, 0xf6,
0x50, 0xe7, 0x09, 0x6c,
0xa9, 0x7c, 0x75, 0x73, 0xbb, 0xd6, 0x79, 0x02, 0xe7, 0xa3, 0xab, 0x7f,
0x85, 0x64, 0x5e, 0xff,
0x69, 0xd1, 0x31, 0xbc, 0xa8, 0x2f, 0x5c, 0x08, 0x64, 0x8d, 0xba, 0x7a,
0xd4, 0x43, 0xee, 0xf5,
0xbf, 0xc1, 0xc1, 0xf5, 0x90, 0x8d, 0x9a, 0x70, 0x5c, 0xa2, 0xa7, 0xe5,
0x40, 0xbc, 0xd1, 0xdd,
0x9e, 0x6a, 0x45, 0xb9, 0x71, 0xd1, 0x9a, 0xa5, 0xca, 0xf6, 0xbd, 0x3b,
0xd3, 0xf4, 0x12, 0x55,
0x43, 0xc4, 0x4a, 0xcc, 0x78, 0xba, 0x95, 0x9e, 0xd1, 0x6a, 0x76, 0xf4,
0x94, 0xa8, 0xe7, 0x01,
0x44, 0xb4, 0xed, 0x6c, 0x47, 0x6b, 0x63, 0xa1, 0xe7, 0x09, 0xca, 0x76,
0xb5, 0x9b, 0xd9, 0x9e,
0xa7, 0x4b, 0xe3, 0xa2, 0x9c, 0x50, 0xa1, 0x6e, 0xff, 0xf6, 0x7d, 0x38,
0x94, 0x1c, 0x80, 0x39,
0xb5, 0xa8, 0xbc, 0xaa, 0xfe, 0x76, 0x71, 0xfd, 0xa7, 0xad, 0x53, 0x4a,
0x6c, 0x12, 0x4f, 0xaa,
0xef, 0x90, 0x30, 0xbf, 0xfe, 0x6c, 0x4a, 0x02, 0x10, 0x56, 0xba, 0xbf,
0xfe, 0x83, 0x53, 0x93,
0x89, 0xdf, 0x4a, 0x4d, 0x0a, 0x67, 0x80, 0x20, 0x4b, 0x6f, 0x97, 0xb4,
0xda, 0xa3, 0xd0, 0xa8,
0x63, 0xf1, 0x91, 0x0b, 0x27, 0x8a, 0x62, 0x4c, 0xf7, 0xae, 0x83, 0x8a,
0x96, 0x05, 0xe3, 0xf9,
0x71, 0xdc, 0x9d, 0xfb, 0x35, 0xa8, 0x39, 0xe8, 0xdb, 0x8b, 0x5d, 0x78,
0xa5, 0x73, 0xa2, 0x48,
0x27, 0xa9, 0xcc, 0x79, 0x0c, 0x8a, 0x20, 0xc8, 0x77, 0x23, 0xd7, 0x40,
0x00, 0x16, 0x4e, 0xd0,
0x5f, 0x3e, 0x46, 0xdb, 0x7f, 0x42, 0x57, 0xea, 0x53, 0xb8, 0xf1, 0xa7,
0x52, 0x4d, 0xad, 0xae,
0xad, 0x12, 0xfe, 0x2a, 0x85, 0xf1, 0x2e, 0xde, 0xe2, 0xc6, 0xee, 0xa7,
0x5a, 0x70, 0xfa, 0xaa,
0x3d, 0xcd, 0x09, 0x31, 0xdf, 0xc5, 0xfb, 0xad, 0xad, 0x06, 0x2e, 0x4e,
0xd4, 0xef, 0x93, 0xe7,
0xab, 0xee, 0x2c, 0x06, 0x0c, 0x6d, 0x9f, 0xaf, 0x76, 0xcc, 0xa8, 0xf6,
0xd5, 0x30, 0xbc, 0x35,
0x08, 0xde, 0x3d, 0xfc, 0x56, 0x83, 0xde, 0x17, 0xc3, 0x4e, 0x69, 0x86,
0x8a, 0x53, 0x7c, 0x1a,

```

    0xd5, 0xdd, 0xb1, 0x56, 0x6e, 0xcb, 0x59, 0x23, 0xb3, 0x19, 0x4d, 0xe3,
    0xff, 0xc8, 0x05, 0x72,
    0x3b, 0xff, 0x6b, 0x72, 0x30, 0x95, 0xd0, 0x7e, 0x83, 0x0c, 0xac, 0x75,
    0x9f, 0x81, 0xdd, 0x67,
    0x60, 0x0b, 0x9a, 0xbe, 0xcf, 0xc0, 0xf2, 0x4d, 0x7a, 0x9f, 0x81, 0xdd,
    0x67, 0x60, 0xf7, 0x19,
    0xd8, 0x7d, 0x06, 0x76, 0x9f, 0x81, 0xdd, 0x4d, 0x06, 0x96, 0xf3, 0x2d,
    0x5c, 0xf2, 0x17, 0x5c,
    0xdd, 0x46, 0xfc, 0x23, 0xce, 0xf9, 0xcf, 0x3e, 0xa3, 0x5f, 0xdf, 0xea,
    0xff, 0xe3, 0x73, 0xef,
    0x3f, 0x85, 0xdd, 0x0a, 0x5e, 0x17, 0x3b, 0x00, 0x00
};
const size_t calibrarbombaSize = sizeof(calibrarbomba);

// Historical data page
static const char historicaldata[] PROGMEM = R"==(
)=="

```