



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

CAMPUS D'ALCOI

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Politécnica Superior de Alcoy

Sistema Básico para la Monitorización de los Parámetros
de Explotación de una Piscina

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Cerdán Galvañ, Víctor

Tutor/a: Micó Tormos, Pau

CURSO ACADÉMICO: 2022/2023

Sistema Básico para la Monitorización de los Parámetros de Explotación de una Piscina

Resumen

El mantenimiento de una piscina mediante la correcta explotación de todos los sistemas provistos para ello no resulta una tarea sencilla. En muchos casos la falta de actuación (o la actuación incorrecta) de alguno de estos sistemas revierte en daños cuya reparación supone costes económicos importantes. El daño se previene si podemos identificar la causa potencial que lo puede provocar. Esta prevención se puede realizar mediante la consulta periódica de los parámetros de explotación de los sistemas de mantenimiento de la piscina, lo que permite corregir actuaciones inadecuadas. Para ello necesitamos registrar de alguna forma dichos parámetros de explotación. Algunos ejemplos son:

- (daño) las luces del interior de la piscina no se encienden → (causa) se han dejado encendidas toda la noche por lo que se ha fundido el transformador de 12V → (parámetro) consumo eléctrico del transformador
- (daño) se ha quemado la bomba del filtro → (causa) no se ha abierto alguna de las tomas del circuito de recirculación → (parámetros) caudal de agua, presión del filtro
- (daño) el agua de la piscina se ha vuelto opaca → (causa) no se ha realizado filtrado → (parámetro) consumo eléctrico de la bomba
- (daño) el agua de la piscina se ha vuelto verde → (causa) no se ha clorado lo suficiente → (parámetro) nivel de cloro del agua
- (daño) el agua de la piscina se ha vuelto verde → (causa) no se ha filtrado el agua el tiempo suficiente → (parámetro) temperatura del agua. Para ajustar el tiempo de funcionamiento de la bomba tenemos: $\text{horas_filtración} = \text{temperatura_agua} / 2$
- (daño) el caudal de agua filtrada es cada vez más débil → (causa) no se ha realizado el proceso de lavado/enjuague del filtro → (parámetro) residuo sólido a la salida del filtro. Si el residuo sólido supera cierto valor umbral salta la alarma de cambio del agente filtrante (arena/cristal). Esto permitiría comprobar la vida útil real del agente filtrante

Por estos motivos se propone el diseño e implementación de un Sistema Básico para la Monitorización de los Parámetros de Explotación de una Piscina (monitorPEP).

Palabras Clave: IoT, MQTT, Monitorización piscina, Automatización

Sistema Bàsic per a la Monitorització dels Paràmetres d'Explotació d'una Piscina

Resum

El manteniment d'una piscina mitjançant la correcta explotació de tots els sistemes proveïts per a això no resulta una tasca senzilla. En molts casos la falta d'actuació (o l'actuació incorrecta) d'alguns d'aquests sistemes reverteix en danys la reparació dels quals suposa costos econòmics importants. El mal es prevé si podem identificar la causa potencial que ho pot provocar. Aquesta prevenció es pot realitzar mitjançant la consulta periòdica dels paràmetres d'explotació dels sistemes de manteniment de la piscina, la qual cosa permet corregir actuacions inadequades. Per a això necessitem registrar d'alguna forma aquests paràmetres d'explotació. Alguns exemples són:

- (mal) les llums de l'interior de la piscina no s'encenen → (causa) s'han deixat enceses tota la nit pel que s'ha fos el transformador de 12V → (paràmetre) consum elèctric del transformador
- (mal) s'ha cremat la bomba del filtre → (causa) no s'ha obert alguna de les preses del circuit de recirculació → (paràmetres) cabal d'aigua, pressió del filtre
- (mal) l'aigua de la piscina s'ha tornat opaca → (causa) no s'ha realitzat filtrat → (paràmetre) consum elèctric de la bomba
- (mal) l'aigua de la piscina s'ha tornat verda → (causa) no s'ha clorat prou → (paràmetre) nivell de clor de l'aigua
- (mal) l'aigua de la piscina s'ha tornat verda → (causa) no s'ha filtrat l'aigua el temps suficient → (paràmetre) temperatura de l'aigua. Per a ajustar el temps de funcionament de la bomba tenim: $\text{hores_filtració} = \text{temperatura_aigua} / 2$
- (mal) el cabal d'aigua filtrada és cada vegada més feble → (causa) no s'ha realitzat el procés de rentada/esbandisca del filtre → (paràmetre) residu sòlid a l'eixida del filtre. Si el residu sòlid supera cert valor llindar salta l'alarma de canvi de l'agent filtrant (arena/cristall). Això permetria comprovar la vida útil real de l'agent filtrant

Per aquests motius es proposa el disseny i implementació d'un Sistema Bàsic per al Monitoratge dels Paràmetres d'Explotació d'una Piscina (monitorPEP).

Paraules Clau: IoT, MQTT, Monitoratge piscina, Automatització

Basic System for Monitoring the Operation Parameters of a Swimming Pool

Summary

The maintenance of a swimming pool through the correct use of all the systems provided for it is not an easy task. In many cases, the lack of action (or incorrect action) of any of these systems results in damage whose repair involves significant economic costs. Damage is prevented if we can identify the potential cause that can cause it. This prevention can be carried out by periodically consulting the operating parameters of the pool maintenance systems, which makes it possible to correct anomalies. To do this, we need to record these operating parameters in some way. Some examples are:

- (damage) the lights inside the pool do not come on → (cause) they have been left on all night so the 12V transformer has blown → (parameter) electrical consumption of the transformer
- (damage) the filter pump has burned out → (cause) any of the recirculation circuit outlets has not been opened → (parameters) water flow, filter pressure
- (damage) the pool water has become opaque → (cause) no filtering has taken place → (parameter) power consumption of the pump
- (damage) the pool water has turned green → (cause) it has not been sufficiently chlorinated → (parameter) chlorine level of the water
- (damage) the pool water has turned green → (cause) the water has not been filtered long enough → (parameter) water temperature. To adjust the operating time of the pump we have: $\text{hours_filtration} = \text{temperature_water} / 2$
- (damage) the filtered water flow is getting weaker → (cause) the filter washing/rinsing process has not been carried out → (parameter) solid residue at the filter outlet. If the solid residue exceeds a certain threshold value, the filtering agent change alarm (sand/glass) goes off. This would make it possible to check the real useful life of the filtering agent.

For these reasons, the design and implementation of a Basic System for Monitoring the Operation Parameters of a Swimming Pool (monitorPEP) is proposed.

Keywords: IoT, MQTT, Pool monitoring, Automation

Agradecimientos

Quisiera agradecer brevemente a todos aquellos que me han ayudado a seguir un buen camino y mantenerme centrado en realizar un buen proyecto, a Míriam Romero por todo su apoyo y cercanía durante estos años tan importantes, a mi familia por estar siempre para todo, a Pau Micó por ofrecerte a tutorizar mi proyecto, por estar presente en cada duda que me ha surgido y tu interés en el seguimiento del proyecto, a Ignacio Miró por estar presente y ofrecerme consejos durante la carrera y en este proyecto, en especial, la recomendación para utilizar el microcontrolador ESP32 en lugar de Arduino, a David Cuesta por incentivarme a estudiar la mención de Ingeniería de Computadores a causa de las asignaturas cursadas, y a la empresa Materiales de Construcción Azulena S.L. (Azulena Piscinas) porque nada de esto habría sido posible sin ellos, por ofrecerme un contrato en prácticas y permitirme continuar trabajando con ellos una vez terminado en un entorno amistoso y profesional.

Contenidos

1 Introducción

- 1.1 Antecedentes
- 1.2 Motivación
- 1.3 Objetivos
- 1.4 Requerimientos

2 Anteproyecto

- 2.1 Estado del arte
 - 2.1.1 Sistemas de monitorización integrales
 - 2.1.2 Microprocesadores industriales
 - 2.1.3 Sensores y periféricos
 - 2.1.4 Buses de conexión
 - 2.1.5 Protocolos de comunicación
 - 2.1.6 Lenguajes de programación
 - 2.1.7 Plataformas IoT online
- 2.2 Estudio de propuestas y estimación de recursos
 - 2.2.1 Escenario 1
 - 2.2.2 Escenario 2
- 2.3 Justificación
 - 2.3.1 Impacto económico
 - 2.3.2 Propuesta final

3 Implementación

- 3.1 Entorno de desarrollo
- 3.2 Implementación práctica
 - 3.2.1 Diseño del producto
 - 3.2.2 Diagramas de funcionamiento
 - 3.2.3 Maquetas
 - 3.2.4 Desarrollo
- 3.3. Pruebas
 - 3.3.1 De sistema
 - 3.3.2 De integración de sistemas
 - 3.3.3 De volumen

4 Resultados

- 4.1 Migración al entorno de producción
- 4.2 Manual de usuario
 - 4.2.1 Instalación de la aplicación
 - 4.2.2 Explotación

5 Conclusiones

- 5.1 Conclusiones personales



[5.2 Futuras líneas de desarrollo](#)

[6 Bibliografía](#)

[7 Acrónimos](#)

[8 Anexos](#)

[8.1 Código](#)

1 Introducción

1.1 Antecedentes

La informática actualmente está muy presente en nuestra vida, en ocasiones más a la vista y en otras ocasiones menos, lo que está claro es que con el tiempo la presencia de la misma pasará cada vez más desapercibida aunque por el contrario se hará más útil, obvia y necesaria, y, este es el caso de los microcontroladores que podrán estar ocultos en pequeños lugares y serán cada vez más complejos.

Los sistemas IoT son cada vez más utilizados y este crecimiento supone una evolución en el día a día de los usuarios la cual se integra a su alrededor de forma inadvertida pero siempre presente a pesar de que pueda no tener un visualizador integrado.

Estos sistemas son inteligentes y hay muchas formas de crear el software necesario para ello, los dispositivos tienen la capacidad de comunicarse entre sí de varias formas posibles y tanto recibir como dar órdenes entre ellos. El IoT se encarga y se encargará cada vez más de las tareas cotidianas o sencillas que nos facilitan el día a día aunque con dedicación al desarrollo y esfuerzo podemos ser capaces de crear sistemas que se encarguen de tareas más complejas como en nuestro caso.

Aunque cada vez vemos más integradas a las personas con estos sistemas, no deja de crecer y tenemos que comentar que esto puede ser incluso llevado mucho más allá. Los datos recogidos de los sistemas IoT pueden ser almacenados y utilizados en Big Data, creando redes neuronales y usando un sinfín de algoritmos que pueden crear redes neuronales con el fin de automatizar nuestro entorno y ayudarnos a tener una vida más fácil. Es usado en casi todos los campos que se puedan imaginar para ayudar a crear soluciones ante problemas, o incluso salvar vidas si tenemos los datos necesarios para poder actuar en función de la necesidad.

Para mantener la seguridad del sistema se pueden usar varios sistemas de control, una seguridad en redes para la conexión de los diferentes protocolos de comunicación y contratos de confidencialidad de datos recogidos por el sistema.

En la industria por ejemplo actualmente se está prescindiendo de muchos empleados que son sustituidos por máquinas inteligentes que hacen el trabajo más preciso, más rápido y son más baratos a la larga dejando otras funciones a las personas como la revisión o control de las mismas. Este cambio repercute mucho a nivel social, y, aunque no entremos en este tema debemos admitir que puede generar mucho conflicto moral.

1.2 Motivación

Este proyecto nace a partir de la experiencia del último curso de la carrera de Ingeniería Informática donde he cursado asignaturas de IoT, Inteligencia Artificial, Seguridad y Redes, todo

esto junto con una empresa que me quiso contratar para trabajar con ellos hizo que fuera posible la propuesta de un proyecto que estuviera compuesto de todas estas.

Una de las preguntas fundamentales es si podemos crear el sistema que muchas piscinas ya tienen pero podemos añadirle el componente IoT para poder mantener completamente la piscina de manera autónoma y tener la información disponible en todo momento, aparte de solucionar problemas específicos en cada caso, de hacer un sistema general que se puede modificar o ampliar dependiendo de las necesidades del usuario, ya sea aumentar la seguridad de la piscina, calentar el agua, añadir productos, securizar el acceso a la piscina, etc...

La inteligencia artificial y el machine learning están a la orden del día y esto me ha motivado a poder crear el sistema que tenemos en mente para más adelante o en futuros proyectos añadir el componente de ML o IA que haría nuestro producto mucho más interesante y llamativo.

En el mercado ya existen sistemas desarrollados y maduros que funcionan muy bien pero en este caso tenemos la ventaja de concretar funcionalidades en el desarrollo, utilizar dispositivos de bajo coste y crear el sistema de forma propia sin tener que depender de un fabricante externo.

1.3 Objetivos

El objetivo del proyecto será el desarrollo e implementación de un sistema básico para la monitorización de los parámetros de explotación y automatización de una piscina totalmente funcional, se utilizarán actuadores y se dotará de la suficiente inteligencia para funcionar de forma independiente y hacer uso de Internet de las cosas para recoger los datos y tener constancia del estado de la piscina de cada usuario (en nuestro caso 1 usuario).

Trataremos de alcanzar los siguientes objetivos por campos y utilizaremos en la medida de lo posible recursos que nos faciliten el desarrollo y sean eficientes:

- Hardware:
 - a) Robusto
 - b) Económico
 - c) Funcional
 - d) Ampliable
 - e) Pequeño en todo su conjunto
 - f) Preparado para impactos ambientales

- Software:
 - a) Intuitivo
 - b) Económico

- c) Fácil de utilizar
- Prototipo:
 - a) Que permita medir Temperatura
 - b) Que permita medir Humedad
 - c) Que permita medir PH
 - d) Que permita medir ORP (Redox)
 - e) Que permita medir Presión
 - f) Que permita medir Temperatura del agua
 - g) Que permita guardar la información en una tarjeta SD
 - h) Que permita mostrar la información mediante una interfaz LCD
 - i) Que permita activar y apagar la bomba de la piscina de forma remota
 - j) Que permita enviar los datos por MQTT a una aplicación web
 - k) Que permita apagar la bomba mediante detección de flujo

1.4 Requerimientos

- Sistema básico para el registro de una serie de parámetros de explotación. Inicialmente se proponen los siguientes: temperatura del agua, pH/ORP (Oxidation/Reduction Potential), cloro (libre, total y combinado)
- frecuencia de registro de los parámetros determinada experimentalmente a partir de las pruebas realizadas sobre un prototipo
- sistema hardware modular que permita aumentar el número de sensores. Facilidad en la conexión del sensor al microcontrolador
- sistema con un registro local de los parámetros. El acceso a los datos se realizará 'in situ' a través de una interfaz física por determinar (USB, tarjeta SD, Ethernet)
- sistema aislado (NO está conectado a ninguna red de comunicaciones)
- sistema NO autónomo (no hay restricciones energéticas por lo que se alimenta conectado a la red eléctrica)
- hardware resistente a la intemperie IP67
- alarma local (indicador luminoso) que informe de gestión incorrecta del sistema monitorizado (parámetro fuera de límites)
- sistema económicamente viable, lo que significa que el TFG debe incluir el impacto económico de su implementación

2 Anteproyecto

2.1 Estado del arte

En una versión inicial, el hardware necesario para el sistema de monitorización es:

- un microcontrolador/microprocesador encargado de leer periódicamente la red de sensores para registrar en memoria los parámetros proporcionados
- una colección de sensores conectados al microcontrolador. Cada sensor mide un tipo de parámetro

En el apartado de estado del arte vamos a realizar un análisis tanto de los sistemas integrales de monitorización de piscinas que existen actualmente en el mercado como de los microcontroladores industriales, sensores y buses de conexión que podemos utilizar para la implementación 'ad hoc' del sistema propuesto.

2.1.1 Sistemas de monitorización integrales

Un sistema de monitorización integral para piscinas es un conjunto de dispositivos y software que permiten monitorizar y controlar diferentes aspectos de una piscina, incluyendo la calidad del agua, la temperatura, la iluminación, la seguridad y el consumo de energía. Algunas características de un sistema de monitorización integral para piscinas pueden incluir:

1. Sensores de calidad del agua: Los sensores pueden medir el pH, la temperatura, la conductividad y otros parámetros para asegurar que el agua de la piscina esté en condiciones óptimas.
2. Sistema de filtración: El sistema de filtración ayuda a mantener el agua limpia y clara, y el sistema de monitorización puede controlar y programar los tiempos de funcionamiento del sistema de filtración.
3. Control de iluminación: El sistema de monitorización puede controlar la iluminación de la piscina, incluyendo la intensidad y el color, para crear diferentes efectos.
4. Control de temperatura: El sistema de monitorización puede controlar la temperatura del agua de la piscina y ajustarla automáticamente según las preferencias del usuario.
5. Seguridad: El sistema de monitorización puede incluir cámaras de vigilancia, sensores de movimiento y alarmas para detectar y alertar sobre posibles peligros, como la presencia de personas no autorizadas en la piscina.
6. Conectividad: El sistema de monitorización puede estar conectado a una aplicación móvil o plataforma en línea que permita controlar y supervisar la piscina desde cualquier lugar.

En resumen, un sistema de monitorización integral para piscinas puede mejorar la calidad del agua, la eficiencia energética, la seguridad y la comodidad del usuario, al proporcionar datos precisos y en tiempo real para su análisis y control.

Automatic PH/CL Bayrol PVP: 1250€

Automatic Cl/pH mide y regula el pH y el cloro mediante la inyección continua de productos de tratamiento de agua según sea necesario.

Diseñado 100% para el usuario, Automatic Cl/pH le da el beneficio de su tecnología Smart & Easy. De hecho, el centro de control integrado adapta automáticamente las "reacciones" del equipo a su propia piscina. Equipado con inteligencia contextual, simplifica su uso al extremo y guía en todas las circunstancias. No hay necesidad de ser un técnico muy capacitado o un profesional para equilibrar el pH de su piscina!

Con el Automatic Cl/pH, el tratamiento automático del pH se convierte en un juego de niños.



Imagen 1. Automatic PH/CL Bayrol

- Dosificador automático de pH y cloro (Redox) Bayrol
- Análisis desde las tuberías.
- Tecnología Smart&Easy, instalación sencilla y rápida mediante pantalla grande y de fácil lectura
- Ideal para piscinas de hasta 90 m3
- Caudal de 1.5 l/h

Equipo compacto 2 en 1

Automatic Cl/pH mide y regula el pH y el cloro. Sus dimensiones lo convierten en un aliado en espacios confinados.

Simple y seguro

Gracias a la tecnología Smart & Easy, es fácil de usar y fiable en su funcionamiento. Incluye alertas visuales y auditivas.

Intuitivo

Su interfaz de usuario es muy fácil de usar. Automatic Cl/pH adapta los ajustes al volumen de la piscina.

Calidad comprobada

Automatic Cl/pH se beneficia de rigurosos controles de calidad en toda la cadena de producción, desde sus componentes mecánicos y electrónicos hasta su montaje.

Interfaz

- Pantalla LCD 4 líneas de 20 caracteres
- Teclado de 5 teclas que incluye un atajo para la calibración de la sonda

Medición

-pH, Redox

Seguridad

- Alarmas de medición alta y baja
- Alarma de tiempo de dosificación inteligente

Medidas

Carcasa: Altura 320 × Longitud 210 × Profundidad 93.5 mm

Control Connect AstralPool PVP: 1625€

REGULACIÓN INTEGRAL MULTIPARAMÉTRICA

Control Connect controla la calidad del agua y el equipamiento de la piscina a través de una única interfaz (equilibrio del agua con dosificación del pH, desinfección con dosificación de cloro líquido, 4 entradas y 4 salidas, con control de bomba del filtro).

CONTROL SIN LÍMITES

Control Connect está disponible en 3 versiones distintas para cubrir todas las necesidades: Regulación y dosificación de pH + ORP de manera independiente (instalación en línea) Regulación y dosificación de pH + ORP en un panel Regulación y dosificación de pH + PPM (sin cloro) en un panel 4 entradas y 4 salidas con control de bomba del filtro

CONECTIVIDAD INTEGRADA

Toda la información relativa a la piscina disponible a distancia mediante el puerto Ethernet integrado y la aplicación PoolStation.



Imagen 2. Control Connect AstralPool

Pantalla multi-función TFT táctil color.
4 relés temporizados programables y 4 entradas digitales.
Poolstation (Ethernet de Serie).
Fluidra Connect Compatible.

Clorador Salino Sugar Valley Hidrolife PVP: 969€



Imagen 3. Clorador Salino Sugar Valley Hidrolife

Equipo de electrólisis salina.

Controles automáticos opcionales: pH, redox, cloro libre, conductividad.

Controles remoto opcionales: (wifi / ethernet + app)

Electrólisis salina a partir de 3 g / l de sal , Puede trabajar con agua de mar.

Célula autolimpiable con placas de titanio.

7 Relés para el control de la piscina: filtración, iluminación, pH + 4 auxiliares.

Este equipo tiene incluido : Pantalla táctil y sensor de temperatura.

Genera cloro a partir de sal común disuelta en el agua de la piscina. Sustituye la compra y manipulación de productos químicos para esterilizar el agua y acaba con los malos olores causados por las cloraminas.

En cuanto a estos sistemas de monitorización, después de investigar por internet e intentar comprar algún sistema que sea similar a nuestro proyecto podemos ver que a pesar de ser muy caros, las opciones son limitadas, no existe un sistema programado o programable que sea tan preciso, meticuloso y completo como el nuestro. Por un lado existen varios cloradores salinos pero no aplican las mismas funcionalidades que el nuestro y por otro lado los sistemas de “automatización” están basados en dosificadores automáticos o relés que son activados por tiempo programable. Para los sistemas de seguridad a diferencia de nuestro flujostato o nuestro sensor de presión vemos que son independientes y por ejemplo no apagarían la bomba sinó

que te indicarían que algo va mal enviando una alarma pero sin la posibilidad de poder apagar el sistema si estás lejos del lugar en cuestión.

Nuestro sistema por otra parte tiene toda la complejidad suficiente para poder tanto avisar como apagar la bomba cuando algo va mal o indicar si necesitas realizar algún mantenimiento para poder continuar con la piscina perfecta, aparte tiene todos los sensores disponibles y no hay varias versiones distintas para cada cosa, sino que crearemos un todo en uno. Nos olvidamos de tener que activar la bomba manualmente (aunque tenemos la opción) porque ya hemos configurado las horas necesarias que necesita estar activa con el modo automático.

Otra de las ventajas es poder acceder a todos los datos y al sistema completo desde cualquier lugar del mundo tan solo teniendo acceso a internet y la cuenta que se proporcionará.

En un futuro se añadirían más funcionalidades sobre las cuales este acceso remoto vendría incluso mejor, por ejemplo añadir una bomba de calor y poder encenderla antes de ir para tener la piscina a la temperatura que tu quieras sin comprometer ninguna parte de tu instalación porque la tienes controlada en todos los aspectos.

2.1.2 Microprocesadores industriales

ESP32 Wemos D1 R32



Imagen 4. Microcontrolador ESP32 Wemos D1 R32

- Alimentación: 5-12VDC
- WiFi 802.11 b/g/n/e/i (802.11n hasta 150 Mbps)
- Bluetooth v4.2 BR/EDR y BLE.
- Compatible con shields de Arduino Uno.
- 6 Entrada analógica.
- 20 Entradas/Salidas digitales (3.3V) (con funciones PWM, interrupción).
- Comunicación UART, SPI, I2C.
- Tamaño: 68Å--53 mm
- Conexión micro USB.
- 4Mb Flash Memoria 520Kb
- Reloj: 240Mhz (un núcleo dedicado al procesaro WiFi)
- Temperatura: -40C+85C
- Corriente: 250mA (max)

Corriente en modo ahorro: 0.15mA
Corriente de funcionamiento: 20mA (sin WiFi)

NodeMCU

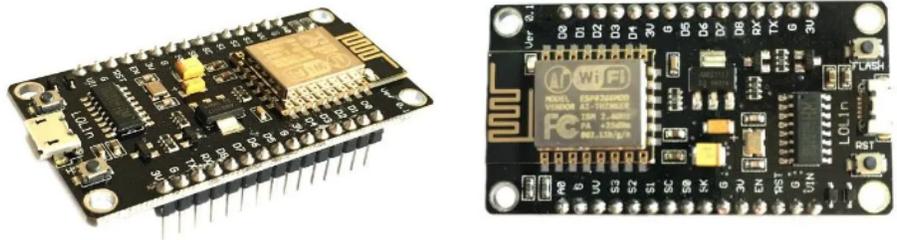


Imagen 5. Microprocesador ESP32 NodeMCU

Voltaje de alimentación: 5v
 Voltaje de entrada/salida: 3.3V
 Corriente de funcionamiento: min. 500mA
 SoC: ESP32-WROOM 32
 Frecuencia de Reloj: 80MHz/240MHz
 RAM: 512kB
 Memoria flash externa: 4MB
 Pines E/S: 34
 Interfaces: SPI, I2C, I2S, CAN, UART
 Protocolos Wi-Fi: 802.11 b/g/n (802.11n hasta 150Mbps)
 Frecuencia Wi-Fi: 2.4 GHz - 2.5 GHz
 Bluetooth: V4.2 - BLE y Bluetooth clásico
 Antena inalámbrica: PCB
 Dimensiones 56x28x13mm

Adafruit Feather M0 WiFi - ATSAM21

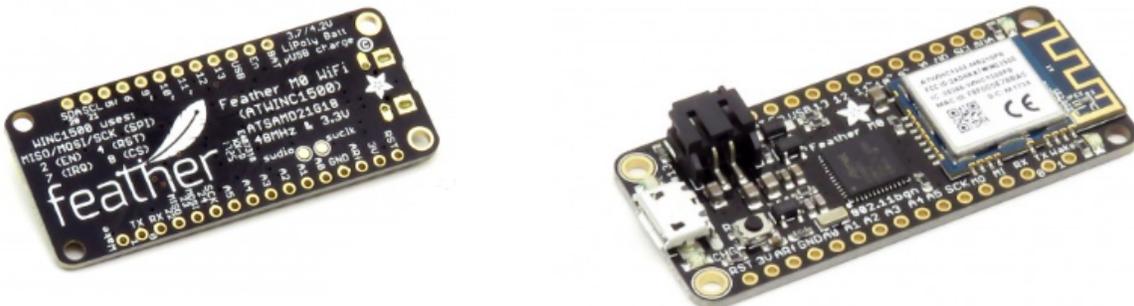


Imagen 6. Microprocesador Adafruit Feather M0

Dimensiones: 53.65 x 23 x 8mm
Peso: 6.1 gramos
Microcontrolador: ATSAM21G18 @ 48MHz (3.3V)
Flash: 256KB,
SRAM: 32KB (no tiene EEPROM)
Regulador integrado: 3.3V (AP2112K-3.3) 600mA max
Soporte USB nativo con bootloader
Pines IO: 20
Interfaces por hardware: Serial UART, I2C, SPI
Pines PWM :8
Entradas analógicas: 10
Salidas analógicas: 1
Cargador LiPo integrado (200mA) con indicador LED
LED integrado programable conectado al pin #13
Pin Power/enable
4 agujeros de sujeción
Pulsador de Reset

Arduino Elegoo Mega 2560 r3

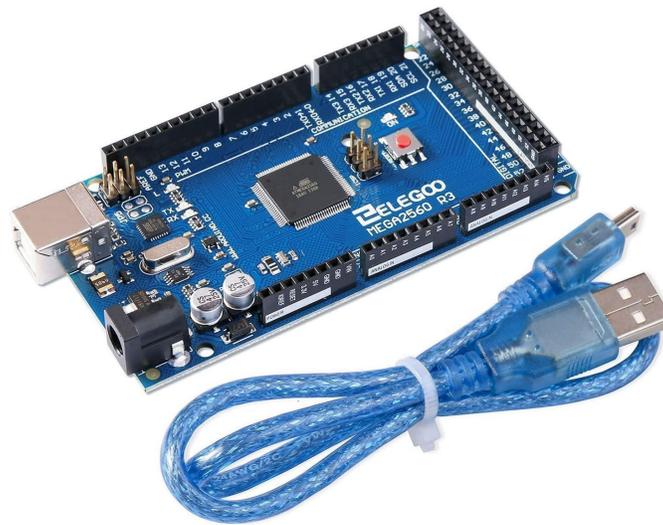


Imagen 7. Microprocesador Arduino ELEGOO Mega 2560 r3

Microcontrolador: ATmega2560
Voltaje Operativo: 5V
Voltaje de Entrada: 7-12V
Voltaje de Entrada(límites): 6-20V
Pines digitales de Entrada/Salida: 54 (de los cuales 15 proveen salida PWM)
Pines analógicos de entrada: 16
Corriente DC por cada Pin Entrada/Salida: 40 mA
Corriente DC entregada en el Pin 3.3V: 50 mA

Memoria Flash: 256 KB (8KB usados por el bootloader)
 SRAM: 8KB
 EEPROM: 4KB
 Clock Speed: 16 MHz

Raspberry Pi Pico

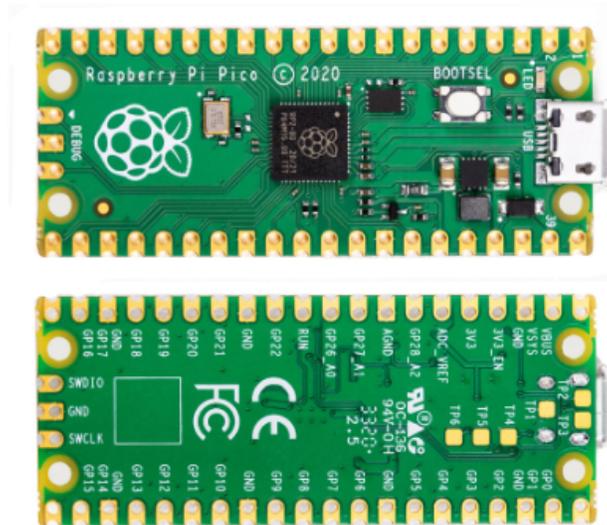


Imagen 8. Microprocesador Raspberry Pi Pico

Microcontrolador RP2040 Dual-Core ARM Cortex M0+
 Frecuencia de reloj flexible corriendo hasta 133MHz (overclock posible)
 264KB de SRAM

2MB de memoria flash integrada

Formato almenado permite soldar directamente en placa portadora

20 orificios a cada lado para soldar tiras de 20 pines

3 orificios para soldar el conector de depuración

USB 1.1 para programación y soporte (dispositivo o host)

Modo bajo consumo (low-power sleep) y modo inactivo (dormant)

Programación sencilla drag&drop usando almacenamiento masivo USB

26x GPIO multifunción

2xSPI, 2xI2C, 2xUART, 3xADC 12-bit (500ksps), 16x canales PWM

Reloj muy preciso y temporizador integrado

Sensor de temperatura

Librerías punto flotante acelerados on-chip

8x E/S programables (PIO) para soporte de periféricos personalizados como tarjetas SD y VGA

Las antes mencionadas son plataformas de hardware con diferentes propósitos y capacidades:

Arduino es una plataforma basada en microcontroladores diseñada para ser fácil de usar y programar para principiantes y aficionados. Es capaz de realizar tareas simples y controlar sensores y actuadores con un bajo consumo de energía. Las placas de Arduino se suelen utilizar para prototipos.

FPGA es un chip programable que permite al usuario crear circuitos digitales personalizados. A diferencia de los microcontroladores, las FPGAs son altamente configurables y se pueden programar para realizar operaciones complejas en tiempo real. Las FPGAs se utilizan comúnmente en procesamiento de señales digitales, procesamiento de imágenes y aplicaciones informáticas de alto rendimiento.

ESP32 es un sistema en un chip (SoC) de bajo costo y bajo consumo de energía que incluye un módulo Wi-Fi y Bluetooth, lo que lo convierte en una opción popular para aplicaciones de IoT. El Esp32 tiene más potencia de procesamiento que un Arduino y es capaz de ejecutar múltiples tareas simultáneamente.

Raspberry Pi es una computadora de placa única (SBC) utilizada para tareas informáticas generales, capaz de ejecutar sistemas operativos completos y se utiliza para muchas cosas, es muy popular debido a su bajo costo, bajo consumo de energía y gran comunidad de usuarios y desarrolladores

Basándonos en todas estas características, en nuestro caso utilizaremos ESP32 por su capacidad de conexión, multitarea y facilidad a la hora de programar en comparación a las FPGAs y su diversidad y la ampliación de posibilidades que ofrece frente a Arduino. Preferimos el ESP32 respecto a Raspberry Pi puesto que es ideal para proyectos de informática en general y servidores que necesiten una gran capacidad de procesamiento, lo que no encaja del todo bien con nuestra idea.

2.1.3 Sensores y periféricos

Transductor de Presión HK1100C

Los sensores de presión o transductores de presión son muy habituales en muchos procesos industriales. Su objetivo es transformar una magnitud física en una eléctrica, en este caso transforman una fuerza por unidad de superficie (presión) en un voltaje proporcional a la presión ejercida.

Entre las principales aplicaciones del sensor tenemos la medición de presión de agua en tuberías, medición de nivel de columna de agua por presión hidrostática, sistemas de presión constante (para dosificación). Fabricado en acero inoxidable para asegurar robustez, sanidad y larga duración.



Imagen 9. Transductor de presión HK1100C

Voltaje de funcionamiento: 5 V CC.
Voltaje de salida: 0,5 – 4,5 V CC.
Material del sensor: aleación de acero al carbono.
Corriente de funcionamiento: ≤ 10 mA.
Rango de presión de funcionamiento: 0 – 1,2 MPa.
La mayor presión: 2.4 MPa
Longitud del cable: 7.5 in.
Presión de destrucción: 3,0 MPa.
Temperatura de trabajo. Rango: 32.0-185.0 °F.
Rango de temperatura de almacenamiento: 32.0-212.0 °F.
Error de medición: $\pm 1,5$ % FS.
Error de rango de temperatura: $\pm 3,5$ % FS.
Tiempo de respuesta: ≤ 2.0 ms
Ciclo de vida: 500.000 piezas.
Aplicación: medición de líquido de gas no corrosivo.

Lector de tarjetas Micro SD

El pinout es directamente compatible con Arduino, pero también se puede utilizar con otros microcontroladores.

Ideal para proyectos con ARDUINO, PIC, AVR,STM32.El módulo lector tarjeta micro SD es una solución sencilla para la transferencia de datos.

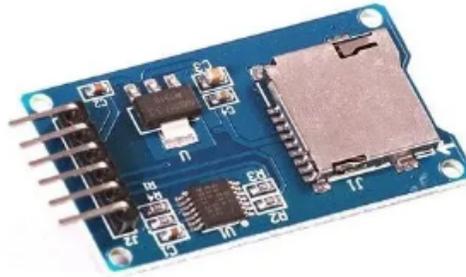


Imagen 10. Módulo lector para tarjetas Micro SD

- Voltaje de alimentación: 4,5V -5.5V
 - Regulador de tensión incorporado 3,3V para pines SPI
 - Consumo <200mA
 - Interfaz: SPI
 - Dimensiones: 24mmx 42mm x 12mm
- Soporta tarjetas microSD (hasta 2G) y micro SDHC (hasta 32G)

DHT22 Sensor Temperatura y Humedad PCB:

El sensor de humedad DHT22 se presenta en un cuerpo de plástico ideal para montar sobre un panel o similar. Utiliza un sensor capacitivo que devuelve una señal digital con la medición. Este sensor a diferencia de DHT11, es capaz de medir valores negativos de temperatura y mide décimas de grado



Imagen 11. Sensor de temperatura y humedad DHT22

- Tensión de operación: 3 a 5,5 VDC
- Rango de medición de humedad: 0-100% $\pm 2\%$ de exactitud
- Rango de medición de temperatura: -40°C a 80°C $\pm 0.5^{\circ}$ de exactitud
- Máxima velocidad de muestreo: 2 segundos

Sensor Temperatura del agua sumergible DS18B20

El DS18B20 es un sensor digital de temperatura que utiliza el protocolo 1-Wire para comunicarse, este protocolo necesita solo un pin de datos para comunicarse y permite conectar más de un sensor en el mismo bus.

Es fabricado por Maxim Integrated, el encapsulado de fábrica es tipo TO-92 similar al empleado en transistores pequeños. La presentación comercial más utilizada por conveniencia y robustez es la del sensor dentro de un tubo de acero inoxidable resistente al agua.

Con este sensor podemos medir temperatura desde los -55°C hasta los 125°C y con una resolución programable desde 9 bits hasta 12 bits.

Cada sensor tiene una dirección única de 64bits establecida de fábrica, esta dirección sirve para identificar al dispositivo con el que se está comunicando, puesto que en un bus 1-wire pueden existir más de un dispositivo. El sensor tiene dos métodos de alimentación.



Imagen 12. Sensor de temperatura del agua DS18B20

Basado en el chip DS18B20
 Tensión Alimentación: 3 ~ 5,5V
 Consumo < 5mA
 Rango de Medición: -55°C do 125°C
 Precisión: +/- $0,5^{\circ}\text{C}$
 Salidas : - GND negro
 - Vcc Rojo
 - Salida digital amarillo
 Acero inoxidable
 Longitud de cable: 1m

Pantalla LCD 20x4 + M3dulo I2C

La pantalla LCD 20x4 alfanumérica monocromática de cristal líquido, conocida también como display LCD 20x4 o 4x20, es una pantalla visual electrónica que utiliza las propiedades de modulación de la luz de los cristales líquidos, los cuales no emiten luz directamente. Es un dispositivo que permite la presentación de caracteres alfanuméricos y otros símbolos en un formato de 20 caracteres por línea, en 4 líneas. Es totalmente compatible con las pantallas basadas en el controlador Hitachi HD44780, por lo tanto la programación y las conexiones son exactamente las mismas. 20x4 significa que se pueden mostrar 20 caracteres en cada una de las 4 filas del display LCD, por lo que se puede mostrar un total de 80 caracteres en cualquier momento. Este display es muy útil para proyectos con microcontroladores PIC, AVR y Arduino ya que es una excelente manera de mostrar textos y datos de los sensores en forma visual.

Una pantalla LCD de 20x4 es un módulo muy básico y se usa comúnmente en varios dispositivos y circuitos. Estos módulos son preferibles a los de siete segmentos y otros dispositivos LED de segmentos múltiples por las siguientes razones: son económicas, fácilmente programables, no tienen limitaciones para mostrar caracteres especiales e incluso personalizados, animaciones, etc. En esta pantalla cada carácter se muestra en una matriz de 5x7 píxeles. Este display LCD tiene dos registros: Comando (para el control) y Datos (la información que el usuario quiere visualizar). No se requiere modificación del código ni de las conexiones entre un LCD estándar de 16x2 o 20x4 (ya que ambos se basan en el controlador Hitachi HD44780 o compatible).



Imagen 13. Pantalla LCD 20x04 + M3dulo I2C

Modelo: 2004A

Formato de presentación: 20 caracteres X 4 líneas

Interfaz de entrada: 4-Bits / 8-Bits

Retroiluminación: LED blanco

Controlador: SPLC780D (compatible con Hitachi HD44780)

Modo de presentación: Fondo azul / Caracteres blancos
 Caracteres: 5X8 puntos
 Tamaño de cada punto: 0.55X0.55 mm
 Separación entre puntos: 0.60X0.60 mm
 Medidas del módulo: 98.0(ancho) × 60.0(alto) × 12(espesor) mm
 Medidas del display (Area de presentación): 76.0(ancho) × 27.0(alto) mm
 Temperatura de funcionamiento: -10 a +60 grados Celsius
 Temperatura de almacenamiento: -20 a +70 grados Celsius
 Voltaje de funcionamiento: 5V
 Consumo de corriente del LCD: aproximadamente 2 mA
 Consumo de la luz de fondo (retroiluminación): 40 mA

BAYROL Redox-Elektrode Automatic Salt + Salt Relax

Este sensor es el primero que utilizamos para descubrir cómo funcionaba la lectura del ORP, son los que me ofrecía la empresa para trabajar con ellos pero no tenía información técnica de su funcionamiento por lo que decidí comprar el siguiente.



Imagen 14. Sensor de ORP Bayrol: Automatic Salt - Salt relax

Gravity: Analog ORP Sensor Meter for Arduino

Este sensor es la clave para hacer mediciones de calidad de agua con el fin de calcular estimaciones de salud y seguridad para el consumo o procesos. Este sensor brinda una configuración sencilla para monitorizar los niveles de ORP (Oxidation-Reduction Potential), en un dispositivo que incluye toda la electrónica integrada para facilitar la medición.

ORP es una medida que indica el potencial de oxidación y reducción de una solución acuosa, A diferencia de las medidas como el PH que siguen curvas logarítmicas difíciles de caracterizar, el sensor de ORP tiende a una curva lineal característica con un bajo mantenimiento al electrodo.



Imagen 15. Sensor de ORP Bayrol: Automatic Salt - Salt relax

Voltaje de trabajo 5.0VDC
 Tamaño del módulo: 40mm X 27mm
 Rango de medición -2000mV-2000mV
 Temperatura de trabajo 5-70°C
 Precisión $\pm 10\text{mV}$ (25°C)
 Tiempo de respuesta <20 seg
 Sonda ORP con conector BNC
 Boton de calibración de cero
 Indicador LED de energía

Gravity: Medidor de PH Analógico V2

Medidor de PH análogo especialmente diseñado para la medición de nivel de PH en una solución y reflejar el nivel de acidez y alcalinidad muy útil en acuaponia, acuicultura y pruebas en agua.

La versión 2 del sensor integra un regulador en la misma board que permite energizarlo con fuentes de 3.3~5.5VDC. La señal de salida es filtrada por hardware con un pasa-bajos y la librería integra el método de calibración de dos puntos para identificar soluciones estándar (4.0 y 7.0).

Con este sensor, una board de control y la librería se puede implementar rápidamente para construir el sensor de PH sin necesidad de soldaduras u otros.

La sonda mide el valor de pH como registro de acidez o alcalinidad conocido también como índice de concentración de iones de hidrógeno. El pH generalmente se registra en valores de 0 a 14 donde bajo condiciones termodinámicas estándar, un $\text{pH}=7$, implica que la solución es neutral; $\text{pH}<7$, indica que la solución tiende a la acidez; $\text{pH}>7$, indica que la solución tiende a ser alcalina.

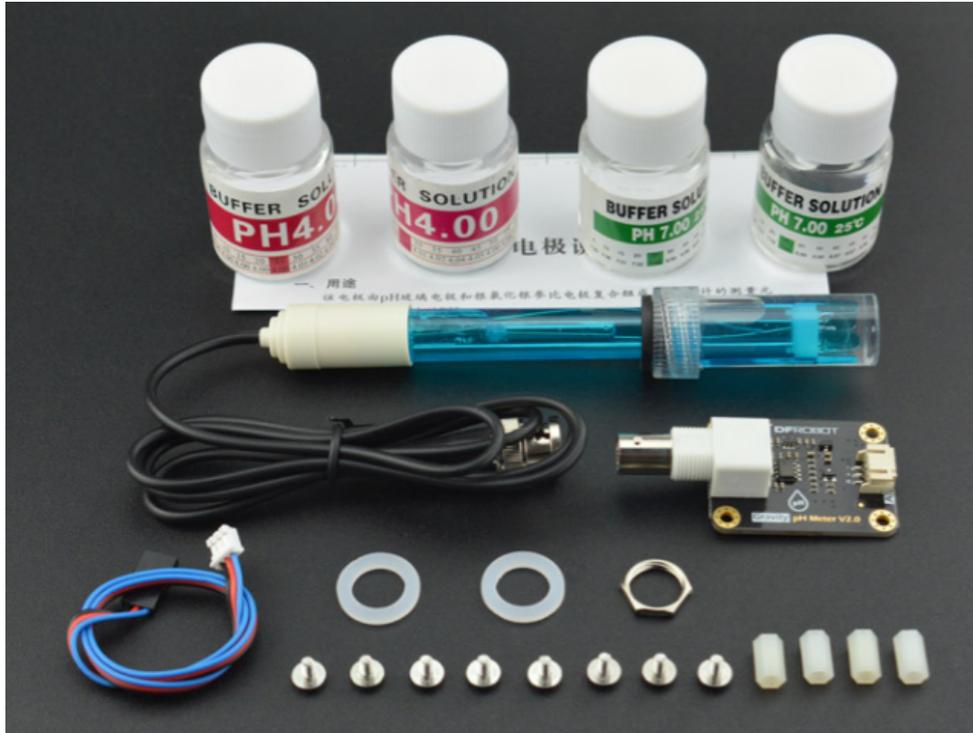


Imagen 16. Gravity: Medidor de PH analógico V2

Voltaje de trabajo 3.3~5.5V

Filtro integrado en el Hardware tipo pasa-bajo

Conector tipo BNC plug and play

Librería por software que soporta la calibración de dos puntos e identifica la solución de ajuste.

Especificaciones:

Board de conversión Version2.

Voltaje de Suministro 3.3~5.5V

salida en voltaje 0~3V

Conector de prueba : BNC

Conector de señal PH2.0-3P

Precisión de medición $\pm 0.1 @ 25^\circ\text{C}$

Dimensiones 42mm x 32mm

Tipo de sonda: Para uso en laboratorio

Rango de detección 0~14

Rango de temperatura 5~60°C

Punto Zero: 7 ± 0.5

Tiempo de respuesta <2min

Resistencia interna <250M Ω

Vida útil de la sonda: >0.5 Años (dependiendo de la frecuencia de uso).

Longitud del cable: 100cm

Gravity: Aislador analógico de señal

El aislador de señal analógica puede aislar la señal analógica y proporcionar energía aislada a los sensores. Puede prevenir eficazmente interferencias de señal del sensor y aplicar a sensores de monitoreo de calidad de agua multiparámetro. Debido a que generalmente los sensores serán interferidos por los otros sensores durante la medición, especialmente aquellos sensores en el sistema de monitoreo de calidad del agua. Por ejemplo, el sensor de pH (medidor de pH) se verá afectado por el sensor de conductividad eléctrica (medidor EC).

La solución común es aislar la fuente de alimentación. Una vez los coloques en el mismo contenedor y compartes la misma fuente de alimentación, su valor no será interferido entre sí. Este aislador de señal analógica está diseñado para esta situación, aislará la señal analógica y proporcionará energía aislada a los sensores. Tiene un IC de potencia aislado de 5V integrado y es compatible con la interfaz de Gravity-3Pin, plug and play, sin necesidad de soldar. La tasa de transformación es 1:1, lo que significa que los valores analógicos en dos lados deben permanecer iguales.

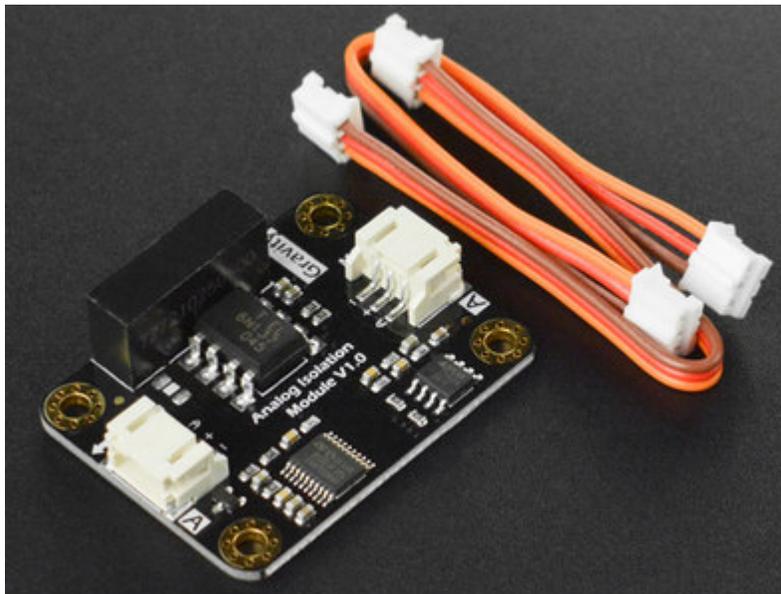


Imagen 17. Gravity: Aislador analógico de señal

Voltaje de funcionamiento: 5,0 V

Tiempo de respuesta: 4 segundos (Typ)

Error analógico (ambos extremos): $<\pm 20\text{mV}$ (Typ).

Tipo de interfaz: PH2.0-3P

Sin corriente de carga: 75ma

Dimensiones: 42x32mm/1,65x1,26 pulgadas

Terminal no aislado (lado Arduino)

Tensión de alimentación: $5,0 \pm 0,1 \text{ V}$

Salida analógica: $0 \sim 5,0 \text{ V}$

Terminal aislado (lado del Sensor)

Voltaje de salida: $5,0 \pm 0,2 \text{ V}$

Entrada analógica: $0 \sim 5,0 \text{ V}$

J-OPTION-FLOW (Flujostato)

Sensor de caudal



Imagen 18. Flujostato J-Option-Flow

Sistema de paleta - Rosca 3/4"
Incluye collarín y machón

Modulo Relé 5V 10A de 8 Canales

Módulo relé de 8 canales (protegido con Optoacoplador). Tiene tensión de Alimentación de 5V y corriente de Salida de 10A. Este módulo puede ser controlado directamente por el microcontrolador (Arduino, 8051, AVR, PIC, DSP, ARM, ARM, MSP430, de la TTL)

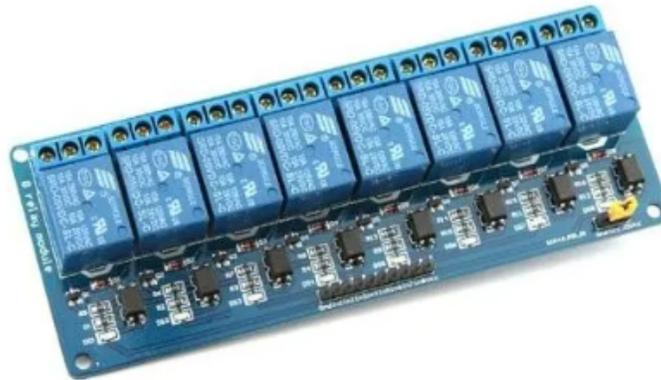


Imagen 19. Módulo 8 relés

Canales: 8 (independientes protegidos con Optoacopladores)

Tensión de Alimentación: 5V

Corriente de Salida: 10A

Corriente de activación por relé: 15mA~20mA

Aislamiento: Si

LED indicador: Para cada canal

Dimensiones: 54x 135mm

Adaptador de sensor BNC de PH PH4502c

Para hidroponía, acuario, procesos de fermentación, laboratorios, ..., este sensor permite medir sencillamente el PH de un líquido

Este módulo PH-4502C es una tarjeta interface que ayuda y mejora la medición del PH, conectado a electrodo para medición de PH como el E201-BNC para el cual está especialmente diseñado.

Voltaje de alimentación de 5V (cuanto más exacto es el voltaje de 5V más exacta será la medición del PH), lo cual lo hace compatible con tecnologías TTL como lo es Arduino, PIC, AVR, DSP, Raspberry entre otros siempre y cuando se tenga una entrada analógica (ADC) disponible.

Generalmente es utilizado con una Tarjeta Arduino por lo que podemos encontrar en la web varios proyectos utilizándolo. Cuenta con 2 potenciómetros, uno de offset para la calibración de la medición de PH (el situado al lado del conector BNC) mientras que el otro potenciómetro controla la salida Do, la cual se activa cuando el PH haya alcanzado el límite que hayamos configurado a través del potenciómetro, cabe mencionar que la salida Do es de 3.7V como salida alta.

El pin Po puede conectarse directamente a un pin analógico de un Arduino, PIC o cualquier tecnología TTL, por lo que no es necesario alguna librería específica para que podamos utilizarlo. Como dato adicional cuenta con dos indicadores LED uno para señalar la alimentación del circuito y otro para señalar la salida Do.



Imagen 20. Modulo adaptador BNC PH4502c

Voltaje de Alimentación: 5VDC +/- 0.2V

Corriente: 5 a 10mA

Rango de PH: 0 a 14

Rango de temperatura: 0 a 80°C

Tiempo de Respuesta: 5seg

Tiempo de estabilización: 60Sg

Temperatura de trabajo: 10 a 50°C Ideal 20°C

Humedad de trabajo: 95 RH sin condensación

Dimensiones de la tarjeta: 42.5 X 32.6 X 20 mm
Mediciones: Temperatura y PH
Disposición de los pines
To: Salida Analógica de Temperatura.
Do: Salida Digital de PH limite.
Po: Salida Analógica de PH.
G: Tierra
Vcc: 5V
Incluye juego de 4 tornillos, 3 tuercas y 2 arandelas.

Tenda A9 Extensor de Alcance Universal Inalámbrico N300

Extensor WiFi para ampliar la cobertura de internet inalámbrico.



Imagen 21. Extensor Wifi Tenda A9

Tipo: Transmisor y receptor de red
Velocidad de transferencia de datos: 300 Mbit/s
Cantidad de antenas: 2
Ganancia de la antena (max): 3 dBi
Tipo de antena: Externo
Estándares de red: IEEE 802.11b, IEEE 802.11g, IEEE 802.11n
Wifi: Si
Wi-Fi estándares: 802.11b, 802.11g, Wi-Fi 4 (802.11n)
Wi-Fi velocidad de transferencia de datos (max): 300 Mbit/s
Algoritmos de seguridad soportados: WPA-PSK, WPA-TKIP, WPA2-PSK
Frecuencia de banda: 2.412 - 2.484 GHz
Color del producto: Gris, Blanco
Voltaje de entrada AC: 100 - 240 V

Frecuencia de entrada AC: 50 - 60 Hz
Intervalo de temperatura operativa: 0 - 40 °C
Intervalo de temperatura de almacenaje: -40 - 70 °C
Intervalo de humedad relativa para funcionamiento: 10 - 90%
Intervalo de humedad relativa durante almacenaje: 5 - 90%
Ancho: 56 mm
Profundidad: 47,7 mm
Altura: 111 mm

Cuadro eléctrico común para conexión de nuestro sistema con relés a corriente de 220V.

antes:

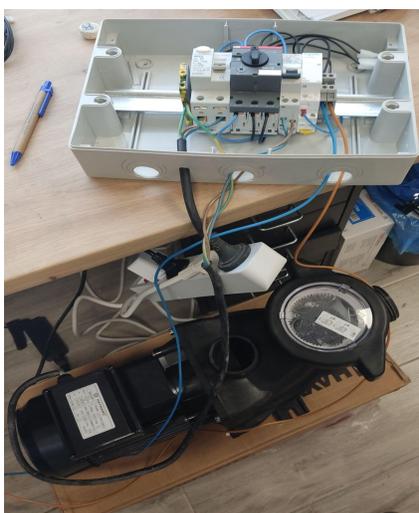


Imagen 22. Cuadro eléctrico sin terminar

después:



Imagen 23. Cuadro eléctrico montado

Componentes del cuadro eléctrico

Interruptor Diferencial KRC3

El interruptor de Tipo AC detecta fallos en componentes de corriente alterna, garantiza una eficaz protección de una fuga de corriente.



Imagen 24. Diferencial

Tensión de funcionamiento nominal 230/415

VCA 50/60 Hz

Conforme con normativas IEC 61008-1: 2010 +AMD1:2020+ AMD2 :2013 IEC
61008-2-1:1990

Tensión nominal soportada a impulso 6 kV

Número de polos: 1P+N, 3P+N

Corriente nominal: 63/80/100 A

Guardamotor 4 a 6 amp

Disyuntor guardamotor trifásico para protección magnetotérmica de motores eléctricos. La regulación del disyuntor guardamotor está comprendida entre 4 a 6 Amp. La instalación de un disyuntor guardamotor permite prevenir la avería de motores eléctricos, tanto monofásicos como trifásicos, por sobrecarga, por fallo de una fase o cortocircuito.



Imagen 25. Guardamotor 4 a 6 amp

Contactor 220V

Dispositivo eléctrico que cumple la función de apertura y cierre de circuitos eléctricos, mediante la conexión y desconexión de sus contactos a través de una señal externa.



Imagen 26. Contactor

Bomba Davey Silensor SLL200E-1.5HP Pump single phase

Las bombas de filtración para piscina son las encargadas de poner en movimiento el agua a través del sistema de filtración. El agua es absorbida mediante los skimmers, toma de aspiración de limpiafondos y el sumidero, hacia al filtro. El filtro (de arena, de cartucho, diatomeas, etc) será posteriormente el encargado de ir reteniendo todas las impurezas para poder devolver el agua limpia a la piscina a través de las boquillas de impulsión.



Imagen 27. Bomba Davey Silensor SLL200E-1

Tensión (V): 220/240
Frecuencia de alimentación (Hz): 50
Potencia de entrada del motor (W/hP): 1130/1.5
Potencia de salida del motor (W/hP): 830/1.1
RPM: 2850
Clase de la caja (IP): 56
Clase de aislamiento: F
Aprobación n.º: A/10EA
Cable eléctrico 3m
Máx. temperatura del agua: 40°C
Máx. temperatura ambiente: 40°C
Nivel de ruido (AS 5102.2): 54 to 56dBa
Dimensiones:
Diámetro de los agujeros de montaje 10x34
Entrada PVC: 50/Salida PVC: 50/Peso neto: 12,5(kg/lbs)

Filtro de arena/vidrio 600 AstralPool Volcano

Son los elementos más utilizados para filtración de aguas con cargas bajas o medianas de contaminantes, que requieran una retención de partículas de hasta veinte micras de tamaño. Las partículas en suspensión que lleva el agua son retenidas durante su paso a través de un lecho filtrante de arena.



Imagen 28. AstralPool Volcano - Filtro de arena/vidrio 600

Filtro de arena Vesubio de Ø 600 mm
Válvula selectora lateral de 6 vías
Bomba Victoria Plus Silent 1 CV monofásica
Cargas necesarias para el filtro de 600 según el medio filtrante empleado:
Carga de arena: 148 kg
Carga de vidrio:
Grado 1: 125 kg

2.1.4 Buses de comunicación

USB

Es un protocolo de comunicación serial utilizado sobre todo para la comunicación entre dispositivos electrónicos.

Este protocolo utiliza una conexión punto a punto, lo que significa que dos dispositivos están conectados directamente uno al otro.

USB admite una amplia variedad de dispositivos y admite una gran variedad de velocidades de transferencia de datos, además utiliza un sistema de jerarquía de dispositivos para controlar la comunicación en el bus USB.

I2C

Protocolo de comunicación serial síncrono que utiliza dos líneas principales de comunicación: SDA como línea de datos de serie y SCL como línea de reloj de sincronización. Los dispositivos conectados a un bus I2C comparten estas líneas de comunicación para la transferencia de datos, este bus puede incluir líneas adicionales de control como SS para la línea de selección de dispositivo y INT para líneas de interrupción.

Lo útil de este protocolo es su bidireccionalidad y puede transferir tanto datos como comandos, cada dispositivo tiene su dirección única de identificación en este bus y aunque es más fácil de implementar tenemos que tener en cuenta que es algo más lento.

SPI

Protocolo serial síncrono utilizado para aplicaciones donde se requiere una comunicación rápida y eficiente de datos como por ejemplo en dispositivos de almacenamiento de datos, algunos tipos de sensores, pantallas LCD, controladores de motores y otros dispositivos periféricos.

Utiliza cuatro líneas principales de comunicación: SCK como línea de reloj de sincronización, MISO como línea de entrada de datos, MOSI como línea de salida de datos y SS/CS como línea de selección de dispositivos. El dispositivo maestro controla la línea de reloj SCK y la línea

de selección de dispositivo SS y transmite datos a través de la línea MOSI al esclavo, mientras que el dispositivo esclavo envía datos a través de la línea MISO.

Tanto el maestro como el esclavo tienen que estar sincronizados con una señal de reloj para enviar y recibir datos, y, por esto SPI es más rápido que otros protocolos de comunicación como I2C.

UART

Es un protocolo responsable de convertir datos paralelos en serie y viceversa, lo que permite la transmisión de datos entre dispositivos a través de un canal serial, es una comunicación asíncrona por lo que no se utiliza una señal de reloj.

Tiene tres líneas de comunicación: TX para la transmisión de datos, RX para la recepción de datos y CTS para la línea de control, aunque la línea de control puede variar dependiendo del dispositivo y puede incluir líneas adicionales como la línea de control de flujo RTS

OneWire

Este protocolo utiliza un solo cable para la transmisión de datos que funciona en serie y sincronizado y permite la comunicación entre dispositivos que utilizan un solo cable y un suministro de energía compartido.

Los datos se transmiten en paquetes de bits sincronizados con un reloj interno del maestro, los dispositivos esclavos reciben datos y responden con un reconocimiento.

Es muy eficiente en cuanto a costos y uso de pines entrada/salida.

2.1.5 Protocolos de comunicación

MQTT

En nuestro proyecto es fundamental conocer todos los protocolos de comunicación que existen y poder elegir el más adecuado, para ello después de investigar llegamos a la conclusión de hacer uno de uno de los protocolos más importantes y actuales que existen en el mundo del IoT. Hablamos del protocolo MQTT de mensajería basado en estándares utilizado para la comunicación de un equipo a otro. Necesitamos transmitir y recibir datos a través de una red con recursos restringidos y ancho de banda limitado, es fácil de implementar y muy eficiente, admite mensajería entre dispositivos a la nube y la nube al dispositivo.

Requiere recursos mínimos, ideal para microcontroladores. Los encabezados son muy pequeños y optimizan el ancho de banda, tanto que un mensaje de control puede tener tan solo dos bytes de datos. Posibilita la opción de conectarse con millones de dispositivos.

Tiene tres niveles de calidad de servicio a fin de garantizar la fiabilidad para los casos de uso IoT, como máximo una vez(0), al menos una vez(1) y exactamente una vez(2). Admite varios lenguajes como Python o C++, nosotros nos basaremos en Arduino IDE y programaremos con C++ en mayor medida lo que hace que podamos desarrollar e implementar muy rápido el sencillo código.

Funciona según los principios del modelo de publicación o suscripción. En la comunicación de red tradicional, los clientes y servidores se comunican directamente entre sí, los clientes solicitan recursos o datos del servidor y a continuación el servidor procesa y envía una respuesta. Sin embargo, MQTT utiliza un patrón de publicación o suscripción para desacoplar el remitente del mensaje del receptor del mensaje, un tercer componente (agente de mensajes) controla la comunicación entre ellos. El trabajo consiste en filtrar los mensajes de editores y distribuirlos correctamente a los suscriptores.

Un cliente MQTT es cualquier dispositivo, desde un servidor hasta un microcontrolador que ejecuta una biblioteca MQTT, si el cliente envía mensajes entonces actúa como editor y si recibe mensajes actúa como receptor.

El agente MQTT es el sistema back-end que coordina los mensajes entre clientes, recibe, filtra e identifica los clientes suscritos a cada mensaje y los envía.

1. Un cliente MQTT establece una conexión con el agente MQTT.
2. Una vez conectado, el cliente puede publicar mensajes, suscribirse a mensajes específicos o hacer ambas cosas.
3. Cuando el agente MQTT recibe un mensaje, lo reenvía a los suscriptores que están interesados.

Wi-Fi

Es un protocolo de comunicación inalámbrico utilizado para la conexión de dispositivos electrónicos a internet y a otros dispositivos dentro de una red inalámbrica local (LAN).

Utiliza ondas de radio de alta frecuencia para transmitir datos entre dispositivos. Los dispositivos WI-FI están equipados con un transceptor que permite la comunicación inalámbrica mediante la transmisión y recepción de señales de radio. Este transceptor utiliza un conjunto de estándares IEEE 802.11 que define las especificaciones técnicas para la comunicación inalámbrica.

Estos dispositivos se conectan a través de un punto de acceso (AP) el cual es típicamente un router inalámbrico que se conecta a una red de área amplia (WAN) para proporcionar acceso a Internet a los dispositivos.

Esta conexión puede ser segura o no segura dependiendo de la configuración del punto de acceso y del dispositivo del cliente. La seguridad de esta conexión se puede mejorar mediante protocolos de cifrado como WPA o alguno mejorado como WPA2.

LoRa

Esta tecnología la hemos tenido muy en cuenta a la hora del estudio de desarrollo del proyecto, en muchas ocasiones ha surgido la idea de poder utilizar este protocolo por sus características, las definiremos a continuación.

Es una tecnología de comunicación inalámbrica de larga distancia y bajo consumo de energía diseñada para IoT y la comunicación máquina a máquina (M2M), utiliza una modulación de espectro ensanchado y se basa en el estándar LoRaWAN.

Utiliza un sistema de transmisión de radio de larga distancia que puede transmitir datos a distancias de varios kilómetros con baja potencia y bajo consumo de energía, se caracterizan

en parte por utilizar una modulación de espectro ensanchado que permite una mayor sensibilidad de recepción y una mayor inmunidad al ruido en comparación con otros sistemas de transmisión de radio. Todo esto permite que los dispositivos puedan comunicarse a larga distancia y en entornos de interferencia de radio frecuentes.

LoRa es especialmente útil para aplicaciones en las que los dispositivos tienen batería de larga duración y se encuentran aislados o en lugares complicados de acceder, como por ejemplo nuestro dispositivo si está en algún campo alejado de un pueblo o ciudad donde la cobertura en general es bastante escasa.

Es especialmente útil para utilizarla en sistemas inteligentes, sensores de calidad del aire y monitoreo de parámetros que necesitan conexión pero ancho de banda limitado.

Bluetooth

Estándar de comunicación inalámbrica utilizado para la transmisión de datos entre dispositivos electrónicos, entre los más utilizados tenemos los teléfonos móviles, auriculares, altavoces, teclados, ratones y otros dispositivos portátiles.

Es una tecnología basada en el uso de ondas de radio de corto alcance para la transmisión de datos. Utiliza un sistema de frecuencia de salto para evitar interferencias y mejorar la seguridad de la transmisión y su alcance es aproximadamente 10 metros.

Durante los últimos años se ha desarrollado y mejorado en términos como velocidad de transmisión, seguridad y eficiencia energética, incluso las últimas versiones ofrecen velocidades de transmisión más rápidas, un mayor rango de transmisión, mayor seguridad y una menor demanda de energía.

En cuanto al IoT se utiliza en muchas aplicaciones como el monitoreo de la salud, control inteligente del hogar y automatización industrial.

Vamos a realizar una comparativa entre los distintos protocolos de comunicación mencionados teniendo en cuenta que MQTT es un protocolo que necesita internet para comunicarse:

El alcance es uno de los factores más importantes a tener en cuenta, y, dado que tenemos la necesidad de utilizar un rango bastante alto descartamos Bluetooth que quedaría en los 10 metros (aunque podríamos activar el acceso para mantenimiento o algo similar en determinadas ocasiones) y nos quedamos con LoRa y WiFi, los cuales tienen un largo alcance. LoRa tiene un alcance muy largo, un consumo de energía muy bajo, una velocidad de transmisión muy baja y es ideal para proyectos como el nuestro, por otro lado WiFi tiene un alcance decentemente alto dependiendo del hardware que utilicemos, un consumo de energía muy alto y es ideal para transferencias de alta velocidad y comunicaciones de corta distancia dentro de una WLAN.

Por simplicidad y situación nosotros utilizaremos WiFi ya que estamos acostumbrados, no necesitamos añadir hardware extra a nuestro sistema como en el caso de LoRa, casi todos los clientes tienen WiFi en casa accesible a su piscina, además sabemos manejarlo y conocemos el funcionamiento del protocolo. En base a nuestras necesidades es más que suficiente y nos ofrece la posibilidad de poder ampliar más adelante nuestro sistema con velocidades más

elevadas o la incorporación de hardware particular que necesite mejor transmisión de datos como por ejemplo una cámara.

En otras situaciones particulares y más difíciles de acceder se estudiará la manera de utilizar LoRa pero se propone para un futuro desarrollo.

2.1.6 Lenguajes de programación

C++

C++ es un lenguaje de programación de propósito general. La intención de su creación fue extender el lenguaje de programación C y añadir mecanismos que permitan la manipulación de objetos. En ese sentido, desde el punto de vista de los lenguajes orientados a objetos, C++ es un lenguaje híbrido. Posteriormente se añadieron facilidades de programación genérica, que se sumaron a los paradigmas de programación estructurada y programación orientada a objetos. Por eso se suele decir que C++ es un lenguaje de programación multiparadigma.

MicroPython

MicroPython es una implementación eficiente y ligera del lenguaje de programación Python 3 que incluye un subconjunto pequeño de la biblioteca estándar de Python y está optimizada para ejecutarse en microcontroladores y entornos restringidos. Está repleto de características avanzadas como un prompt interactivo, enteros de precisión arbitraria, cierres, comprensión de listas, generadores, manejo de excepciones y más. Sin embargo, es lo suficientemente compacto como para caber y ejecutarse dentro de solo 256k de espacio de código y 16k de RAM. Tiene como objetivo ser lo más compatible posible con Python normal para permitirte transferir código con facilidad desde el escritorio a un microcontrolador o sistema embebido.

Lua

Lua es un lenguaje de programación multiparadigma, imperativo, estructurado y bastante ligero, que fue diseñado como un lenguaje interpretado con una semántica extendible. Está diseñado principalmente para ser utilizado de manera incorporada en aplicaciones. Lua es un lenguaje multiplataforma y su intérprete está escrito en el estándar ANSI C.

Lua es un lenguaje interpretado y no está diseñado para ser compilado en código de máquina, por lo que no hay nada que pueda cargarse en un Arduino como un archivo "hex". Sin embargo, hay una biblioteca de Arduino llamada ESP-Arduino-Lua que proporciona el motor de scripting Lua 5.3.5 para bocetos ESP8266/ESP32. Esto permite la ejecución dinámica del código Lua en el Arduino sin tener que compilar y cargar un nuevo firmware.

Para la elección del lenguaje en el que vamos a programar hemos tenido varias cosas en cuenta, la idea principal es utilizar un lenguaje que ya sabemos utilizar, ¿Merece la pena aprender un lenguaje nuevo de programación que no conocemos? En mi caso la decisión ha sido continuar con C++ dado que no sé si en un futuro se desarrollará más el proyecto. Por el

momento C++ parece ideal para nosotros, podemos desarrollar casi cualquier cosa si sabemos programar bien en este lenguaje. Tanto MicroPython como Lua son lenguajes interpretados y dinámicos que ofrecen facilidad de uso y flexibilidad pero C++ es un lenguaje compilado y estático que ofrece un alto rendimiento y control sobre la memoria.

2.1.7 Plataformas IOT online

ThingSpeak

ThingSpeak es una plataforma de análisis de IoT que te permite agregar, visualizar y analizar flujos de datos en vivo en la nube. Puedes enviar datos a ThingSpeak desde tus dispositivos, crear visualizaciones instantáneas de datos en vivo y enviar alertas.

Para usar ThingSpeak, debes iniciar sesión con tu cuenta MathWorks existente o crear una nueva. Los usuarios no comerciales pueden usar ThingSpeak de forma gratuita, aunque las cuentas gratuitas ofrecen límites en ciertas funcionalidades. Los usuarios comerciales son elegibles para una evaluación gratuita por tiempo limitado.

Ubidots

Ubidots es un software de IoT que las empresas industriales utilizan para lanzar aplicaciones web y móviles para el monitoreo de condiciones, control de procesos en la nube (SCADAs), análisis de datos y más. Con las herramientas intuitivas y probadas en la industria de Ubidots puedes centrarte en tu negocio y llegar más rápido al mercado.

Ubidots también ofrece una versión STEM para makers, estudiantes e investigadores interesados en probar, aprender o enseñar IoT. Es perfecto para unirte a miles de usuarios de todo el mundo en la creación de un futuro impulsado por los datos.

Tanto Thingspeak como Ubidots son plataformas de IoT que habilitan la toma de decisiones a empresas de integración de sistemas a nivel global, al inicio del proyecto decidí usar Thingspeak por la facilidad de integración en el código y que las vistas del dashboard donde se recogen los datos parecía más vistoso, después me di cuenta que Ubidots ofrecía también la posibilidad de integración bastante fácil, un dashboard que no está nada mal y la oportunidad de utilizar MQTT con suscribe y publish, en Thingspeak solo podía consultar los datos que estaba recibiendo de los sensores y no podía enviar un dato al microcontrolador.

2.2 Estudio de propuestas y estimación de recursos

En base a la información recopilada en el estado del arte y los requerimientos especificados por el cliente, se proponen los siguientes escenarios de desarrollo en el que expondremos una breve justificación de cada uno de ellos y una recopilación del hardware y software a utilizar:

2.2.1 Escenario 1

Arduino Mega
LCD Display (16x2)
Sensor ultrasonidos HC-SR04 o Sensor de nivel de agua Water Level Sensor
Sensor de temperatura y humedad DHT11
Sensor temperatura agua DS18B20
Reloj DS1307 Module V03
Transductor de presión HK1100C
MicroSD Card Adapter
Módulo Wifi ESP01-8266
PH4502C módulo de sensor regulador de PH
Sonda Bayrol de PH-Einstabmesskette
Sonda Bayrol de ORP
Módulo de relé 1 canal

Por un lado en el escenario 1 vemos como tenemos sensores con modelos algo básicos y electrónica bastante barata que nos serviría para desarrollar un proyecto bastante bueno aunque con la placa de arduino no tenemos conexión a internet, habría que comprar un módulo WiFi y eso conlleva la integración de más hardware en nuestro sistema aparte de que nos reduce mucho el uso general de IoT, pese a que la placa arduino es más cara la electrónica es muy similar ya que no hay apenas diferencia entre ellas, incluido utilizamos algunos sensores iguales. La capacidad de conexión de la placa mega con el resto de sensores es muy grande, tenemos una gran cantidad de pines disponibles y varios puertos de comunicación de varios protocolos distintos lo que proporciona bastante libertad a la hora de organizar el cableado y poder separar mejor unos sensores de otros. Tenemos por otro lado, sensores de PH y ORP más caros y que no nos ofrecen una mejora en nuestras lecturas ni en nuestra durabilidad por lo que no nos interesa comprar el sensor más caro si vamos a conseguir lo mismo. En cuanto al software, utilizamos la plataforma Thingspeak que nos ofrece una visualización rápida del estado de nuestras variables de nuestro sistema.

2.2.2 Escenario 2

ESP32 Wemos D1 R32
LCD Display (20x4)
Sensor de temperatura y humedad DHT22
Sensor temperatura agua DS18B20
MicroSD Card Adapter
Transductor de presión HK1100C
Módulo de relé de 8 canales
Sonda Gravity PH
Sonda Gravity ORP
Circuito aislador DFRobot

J-OPTION-FLOW (Flujostato)

Por otro lado en cuanto al escenario 2 tenemos sensores muy similares pero el cambio más importante es que cambiamos la placa por una más ajustada a las necesidades de nuestro proyecto en el sentido de la conexión WiFi y Bluetooth (que no usaremos en este momento), en particular nos viene muy bien la conectividad inalámbrica sin necesidad de una placa externa para poder realizar la conexión y subir los datos a la nube. La elección de sensores es quizá un poco más meticulosa, usaremos prácticamente los mismos sensores y si podemos alguna versión mejorada que no sea mucho más cara para poder mejorar la precisión de la medición.

En cuanto a la estimación de recursos vamos a ser generosos y vamos a comprar todo lo necesario sin tener que pensar en ahorrar en detalles de sensores que puedan producir cualquier lectura errónea, o problemas como el exceso de calor o cualquier detalle que pueda ser perjudicial para nuestro sistema, buscamos eficiencia a la hora de medir, que el sistema pueda permanecer activo el mayor tiempo posible y que se caliente lo menos posible, proporcionando activamente los datos recogidos a la tarjeta SD y a la aplicación web por medio de conexión WiFi. En nuestro caso no nos vamos a meter en la preocupación de diseñar una caja acorde a las medidas del proyecto y su ventilación porque daría lugar a un proyecto mucho más grande.

En cuanto al software realizamos una mejora hacia el uso de la plataforma Ubidots que nos ofrece la visualización de los parámetros de nuestro sistema en la web y una interacción por MQTT con nuestro sistema.

2.3 Justificación

Una vez vistas las propuestas anteriores, se trata de estudiar el impacto de cada propuesta, suponiendo las horas de desarrollo, una estimación del presupuesto, lo que necesitamos para desarrollar la propuesta, tanto software como hardware, ya que se tiene que ver que proyecto elegir, por eso a continuación se estudiarán las propuestas anteriores.

2.3.1 Impacto económico

En cuanto a los costes, siendo sinceros y como he explicado anteriormente el impacto económico será medio ya que para obtener buena información necesitamos sensores fiables, pero teniendo en cuenta la cantidad de valores que vamos a recoger, el tiempo que vamos a ahorrar, la calidad del servicio, los empleados que no vamos a precisar, etc... podemos llegar a la conclusión de que el coste a la larga es más bien bajo.

En el estudio de mercado hemos visto sistemas similares ya montados y su variación en el precio en función de sus utilidades, por ejemplo, si utilizamos WiFi tendría un coste adicional a si solo compramos el sistema y lo utilizamos tan solo por Bluetooth. Hemos comprobado también la calidad de la pantalla y es muy similar al nuestro aunque los productos más modernos (versiones nuevas) ofrecen una pantalla táctil, esta mejora le hemos propuesto para una futura ampliación del proyecto y en este caso instalaremos una de mejor calidad que todas las que he visto (esto está desarrollado en el apartado de mejoras).

Los precios son bastante elevados y en ningún caso he visto la inclusión de un mantenimiento o ayudas a la hora de saber cómo operar con el sistema, tendremos esto en cuenta en el análisis de costes y beneficios para incluir una opción de mantenimiento dado que la explicación de cómo funciona el sistema se proporciona con la compra del producto (he pensado en la opción de añadir un código QR con nuestro producto para la lectura de un manual de uso).

En este apartado voy a proponer una tabla con los valores medios de los gastos aproximados de la obtención de todo el hardware, la posibilidad de utilizar un software para los datos más allá de uno gratuito y el salario que debería cobrar un ingeniero por desarrollar este sistema:

Costes Escenario 1:

Costes Escenario 1	Precio	Horas	Comentarios
Personal			
Ingeniero en prácticas	4,40 €	450	3 meses
	1.980,00 €		
Hardware			
Arduino Mega 2560 R3	50,03 €		
TempAgua DS18B20	3,00 €		
Presión HK1100C	16,48 €		
DHT11	6,49 €		
Sonda ORP Bayrol	81,83 €		
Sonda PH Bayrol	36,32 €		
Tabla 8 relés	12,74 €		
Protoboard 5pc	7,99 €		
Repetidor Tenda A9	14,90 €		
Lector/Grabador Micro SD	2,49 €		
Pantalla I2C LCD 16x2	4,84 €		
	237,11 €		
Software			
ThingSpeak	54,16 €		3 meses, aunque podríamos usar versión limitada hasta crecer
	162,48 €		
Costes Totales:	2.379,59 €		
Posibles Beneficios Escenario 1			
Producto y Servicios			
Venta del producto	900,00 €		
Mantenimiento Software	50,00 €		mensual opcional
Mantenimiento Hardware	50,00 €		mensual opcional
Mantenimiento y Control 24h	100,00 €		mensual opcional
Beneficios Totales:	1.100,00 €		

Imagen 29. Análisis de costes escenario 1

Costes Escenario 2:

Costes Escenario 2	Precio	Horas	Comentarios
Personal			
Ingeniero en prácticas	4,40 €	450	3 meses
	1.980,00 €		
Hardware			
Wemos D1 R32	9,95 €		
TempAgua DS18B20	3,00 €		
Presión HK1100C	16,48 €		
DHT22	6,49 €		
Gravity aislador	18,30 €		
Gravity PH	81,83 €		
Gravity ORP	36,32 €		
Tabla 8 relés	12,74 €		
Protoboard 5pc	7,99 €		
Repetidor Tenda A9	14,90 €		
Lector/Grabador Micro SD	2,49 €		
J-OPTION-FLOW	134,00 €		
Pantalla I2C LCD 20x4	4,84 €		
	349,33 €		
Software			
Ubidots	50,00 €		3 meses, aunque podríamos usar versión limitada hasta crecer
	150,00 €		
Costes Totales:	2.479,33 €		
Posibles Beneficios Escenario 2	Precio	Horas	Comentarios
Producto y Servicios			
Venta del producto	900,00 €		
Mantenimiento Software	50,00 €		mensual opcional
Mantenimiento Hardware	50,00 €		mensual opcional
Mantenimiento y Control 24h	100,00 €		mensual opcional
Beneficios Totales:	1.100,00 €		

Imagen 30. Análisis de costes escenario 1

2.3.2 Propuesta final

En cuanto a la elección de la propuesta nos quedaremos con el escenario 2 por razones obvias, la necesidad de conexión a Internet hace que el ESP32 sea ideal para nuestro proyecto puesto que la misma placa tiene WiFi y Bluetooth integrado, al contrario que el Arduino Mega, el cual necesitaría recurrir a hardware externo para poder comunicarse mediante un módulo ESP8266 a internet además de tener un precio más elevado. Incluimos también mejoras en la calidad de los productos que vamos a adquirir, asegurando así una larga duración.

El segundo cambio más importante respecto a estas dos placas es el cambio en el voltaje de funcionamiento, independientemente de la velocidad de los procesadores y su memoria es importante notar que a pesar de que ambos tienen los prácticamente los mismos protocolos de comunicación a excepción de WiFi y Bluetooth, el ADC del ESP32 tiene 12 bits de resolución y trabaja a 3.3v máximo. Este detalle hace que tengamos que modificar varias fórmulas para ajustarnos al voltaje de trabajo así como la posible necesidad de crear divisores resistivos para poder leer de forma adecuada los valores proporcionados por los sensores.

Para nuestro entorno de desarrollo necesitamos contar con varias herramientas e instrumentos que nos permitan realizar pruebas y estar preparados para cualquier imprevisto.

Hardware:

Necesitamos tener una conexión WiFi estable y esto lo vamos a conseguir con la instalación de un extensor WiFi de una red cercana a nuestra piscina de pruebas. Claro está que necesitamos una piscina cercana también.

En nuestro caso vamos a utilizar un vaso de agua que rellenaremos de nuestra piscina de pruebas que tenemos cercana (la nuestra está a unos 10 metros) con el que realizaremos las pruebas.

Utilizaremos un manómetro para poder medir la presión de manera real y poder ajustar bien la presión que estamos midiendo mediante el ajuste idóneo por software.

Necesitamos un multímetro para medir los voltajes que tenemos de entrada y salida de nuestro sistema en función de los sensores que estamos utilizando, de este modo podemos realizar los ajustes de calibración mediante software y la certificación de que estamos midiendo lo que tenemos que medir sin fallos.

Tenemos todos los sensores, líquidos de prueba, herramientas como destornilladores, cables, enchufes, alargadores, resistencias, etc.. disponibles cerca de nuestro lugar de pruebas.

Nos ayudamos de una caja abierta para poder transportar nuestro sistema y aislarlo del exterior dejándolo así un poco más recogido.

Todo esto no sería posible sin un portátil desde el que realizaremos todo, tanto la documentación como la programación, y, al ser un portátil tenemos la ventaja de poder por ejemplo desplazarnos al entorno de producción u otro lugar distinto y realizar pruebas en otros lugares bajo otras condiciones aunque para esto también utilizaremos un Smartphone porque Ubidots aparte de ofrecer un servicio web también tiene aplicación (no vamos a entrar en este detalle).

Software:

En cuanto al software utilizado será muy amplio y surgen necesidades constantemente por lo que resumimos las más básicas y que más he utilizado.

Lógicamente necesitamos un IDE para la programación y utilizaremos Arduino IDE, un entorno de desarrollo integrado utilizado para programar microcontroladores y proporciona una interfaz gráfica de usuario para escribir, compilar y cargar código en una placa. El IDE de Arduino está disponible para Windows, Mac y Linux y puede descargarse gratuitamente desde el sitio web de Arduino.

En la siguiente imagen podemos ver junto a una leyenda los diferentes apartados del IDE de arduino.

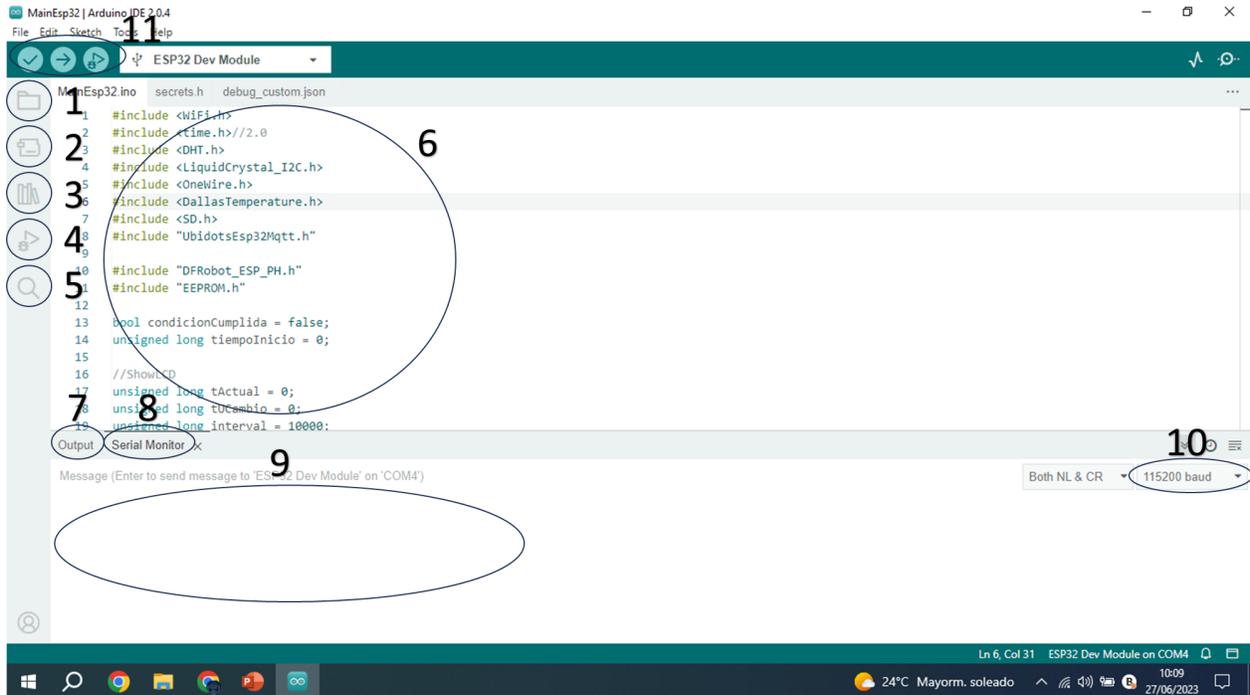


Imagen 32. Interfaz Arduino IDE

1. Sketchbook: Acceso directo a tus sketch.
2. Boards Manager: Acceso directo desde donde ver el software para tus tarjetas, instalar nuevo software o actualizar/desactualizar la versión del mismo.
3. Library Manager: Acceso directo a la visualización de las versiones de las librerías instaladas o por instalar.
4. Debug: Herramienta para debugear situaciones, breakpoints, threads, memoria, variables, etc..
5. Search: Un lugar desde donde podemos buscar cualquier palabra o bloque incluida en nuestro código.
6. Área de código: Aquí observamos el código que estamos creando.
7. Output: Consola que nos proporciona los datos de ejecución.
8. Serial Monitor: Abre la caja de texto.
9. Caja de texto: Datos recibidos del puerto serie.
10. Baudios: Velocidad de transmisión de datos por comunicación serial.
11. Compilar, cargar y verificar código.

En la siguiente imagen podemos ver la selección de la tarjeta que vamos a utilizar así como su configuración particular:

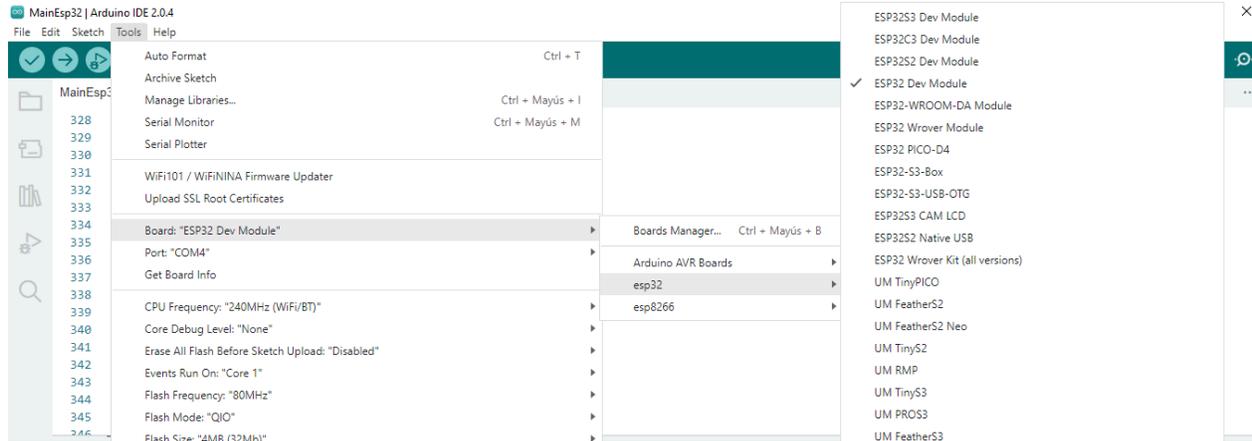


Imagen 33. Interfaz de herramientas Arduino IDE

Utilizaremos Microsoft Word para poder realizar la documentación de todo el proceso realizado y la toma de datos necesarios.

Microsoft Paint para las ediciones de imágenes o textos más sencillos y de forma rápida.

Fritzing para la creación de las maquetas y diseños que nos permitirán visualizar el posible futuro montaje de nuestro sistema.

Claro que necesitamos un navegador para toda la búsqueda de información relevante, consulta de datos, visualización de nuestros datos y para ello utilizaremos Google Chrome.

Poco más que añadir en general puesto que el equipo de producción se reduce solo a mí realizando todas las tareas.

3.2 Implementación práctica

Este proceso creativo y técnico para desarrollar nuestro nuevo producto implica identificación de necesidades del mercado y del consumidor, generación de ideas, selección de opciones, creación de prototipos, pruebas y producción.

Necesitamos analizar meticulosamente las opciones que tenemos, haremos uso de nuestra habilidad de ingeniería, diseño gráfico, marketing y gestión de proyectos para poder llevar a cabo este proceso.

Nos basaremos principalmente en la funcionalidad, estética, ergonomía, durabilidad, seguridad, sostenibilidad y costo.

3.2.1 Diseño del producto

Desde el principio de la creación de la idea de nuestro producto tenemos muchos factores claros que queremos poner en funcionamiento.

Hardware:

Sabemos que queremos un producto lo más pequeño posible, que no genere demasiado calor, y si genera calor que sea capaz de no influir en el sistema, poder disipar y expulsarlo fuera de la “caja” donde iría todo montado. Todo esto montado en una pantalla táctil para poder ver todos los datos y hacer uso de las funcionalidades del sistema, todo montado en una caja con un diseño atractivo para todos los públicos aunque es posible que se realicen varias opciones de diseño de producto para distintos públicos (diferencias en color, tamaño y adaptable al lugar donde sería colocado).

Después de investigar un poco también tenemos en cuenta que los clientes piden un sistema robusto que consuma poco y lo haga “todo”, es por eso que necesitamos varias opciones y por ellos se crearán placas adaptables a distintos sensores y funcionalidades. No todos los diseños tendrán todo incluido y algunos serán más completos que otros y por esto variaría también el precio en general.

Software:

En cuanto al diseño de una aplicación software necesitamos una conexión sencilla pero estable y fiable, un punto de acceso muy cercano o un rango muy amplio que nos llegue al lugar donde queremos instalar el sistema, en nuestro caso suelen ser campos o lugares alejados del núcleo urbano por lo que tendríamos que implementar una conexión LoRa, WiFi cercano que podamos utilizar o una conexión GPRS.

Necesitamos una base de datos para los datos que envía el sistema de los clientes. Este envío sería cifrado por lo que necesitamos seguridad en el servidor que vamos a utilizar, por lo que utilizaremos el sistema de Token Auth.

El cliente tendría acceso a la visualización de todos sus datos y la interacción de lo que le demos acceso, por ejemplo accionar o apagar la bomba.

Tenemos por lo tanto una aplicación con un dispositivo por cliente con todas las métricas mostradas en un dashboard de manera muy visual, personalizable y disponible en todo momento.

Para la accesibilidad el software debe ser sencillo y claro, no nos excederemos en opciones básicas que no sean importantes y trataremos de no sobrecargar las pantallas y poder visualizar mejor los datos de una manera sencilla sin tener que entrar en menús, desplegables y más opciones. Esto hará que sea accesible para todos los públicos y con pocas instrucciones sepamos utilizarlo de manera fácil.

3.2.2 Diagrama de funcionamiento

Un diagrama de flujo es una representación gráfica de un proceso o algoritmo que se utiliza para ilustrar visualmente las etapas o pasos necesarios para completar una tarea o resolver un problema y en cuanto a nuestro diagrama de la siguiente imagen podremos ver todo el proceso que lleva a cabo nuestro sistema de forma breve puesto que si tuviéramos que crear un diseño de todos los detalles que tenemos programados sería inviable, por lo que resumimos de forma breve su funcionamiento.

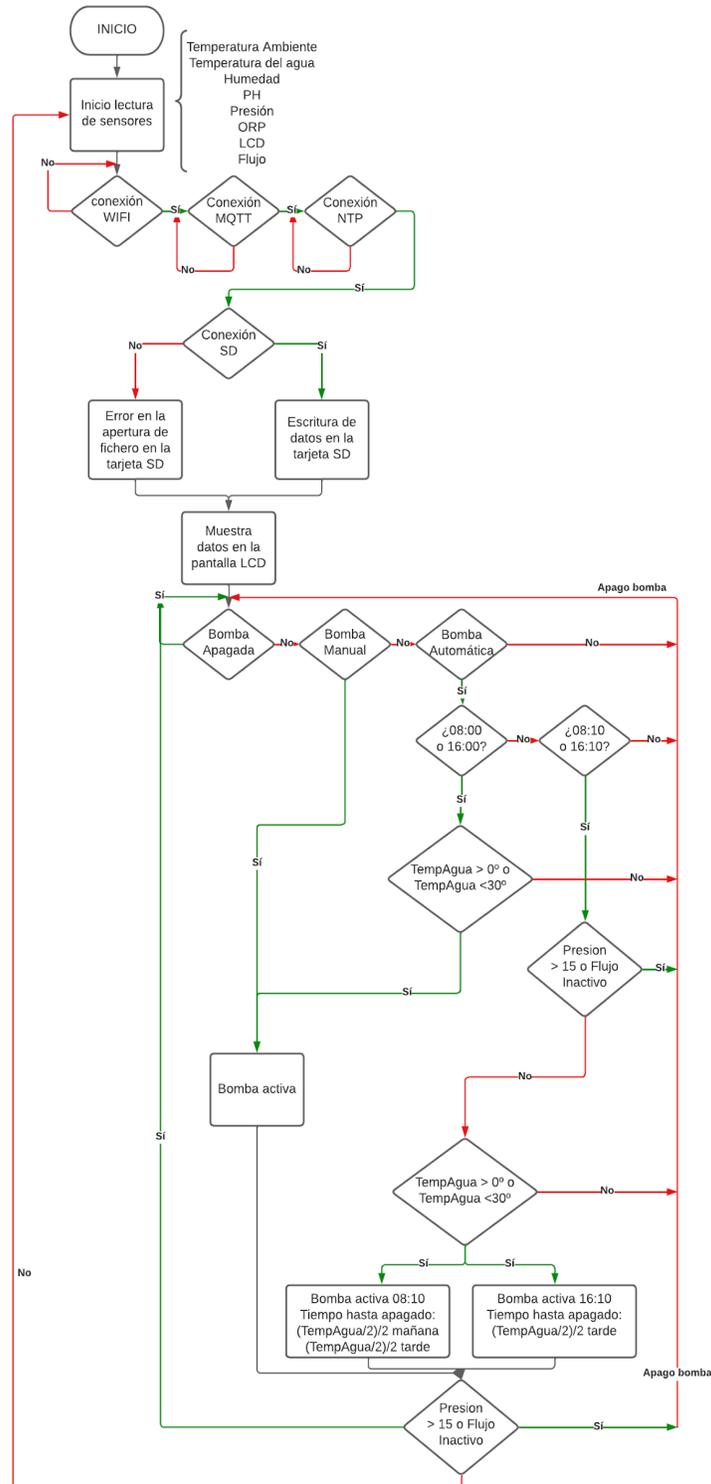


Imagen 34. Diagrama de funcionamiento

Dado que el Esp32 está en un bucle constante podemos ver que el diagrama tiene un inicio pero no un fin, con lo cual nos mantenemos siempre conectados y en constante recopilación de información de los sensores.

3.2.3 Maquetas

Con la maqueta representaremos visualmente el software en fase de diseño, con esto mostraremos cómo será la interfaz de usuario y cómo funcionará el software en primera instancia (sujeta a cambios). Por otro lado también mostraremos la arquitectura del proyecto y su esquemático donde podemos ver todas las conexiones de cada sensor

Hardware:

En la arquitectura de la siguiente imagen podemos ver como están todos los sensores conectados entre sí para formar el sistema completo:

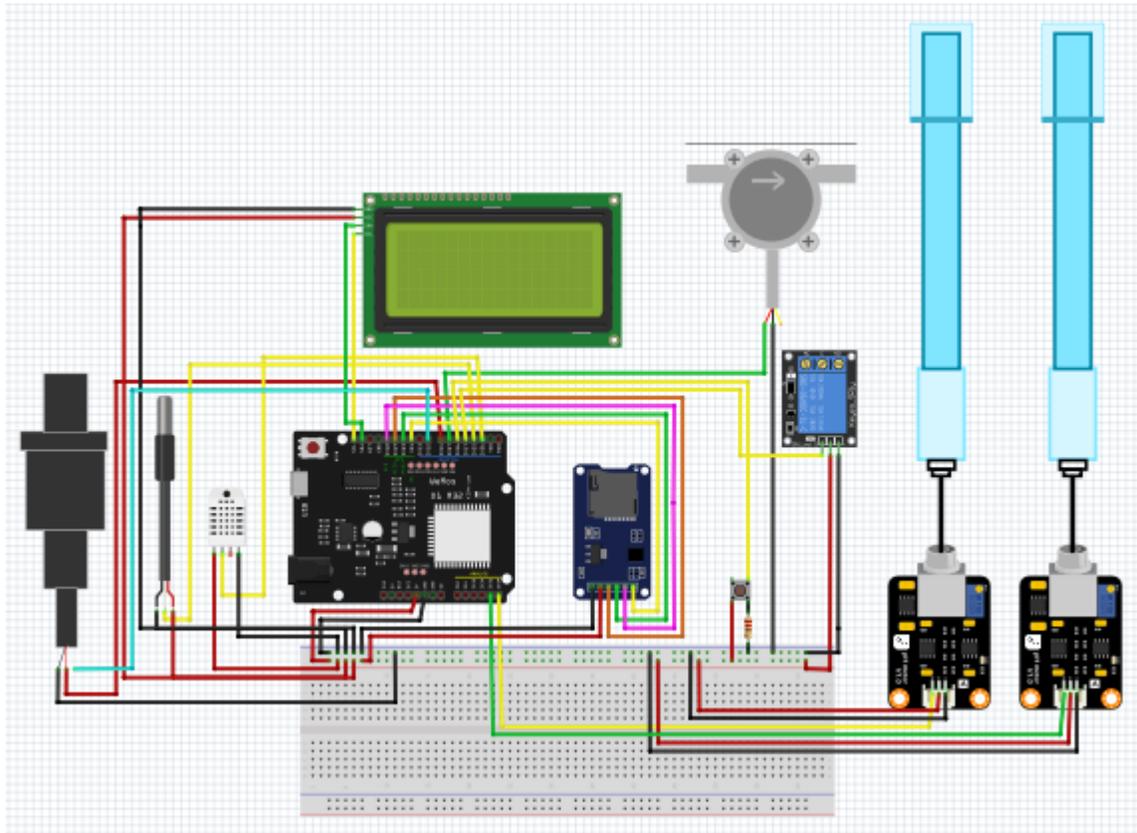


Imagen 35. Diagrama de arquitectura hardware

En cuanto al esquemático de la siguiente imagen podemos ver a bajo nivel todas las conexiones de los sensores y cómo se comunican entre ellos con mayor detalle:

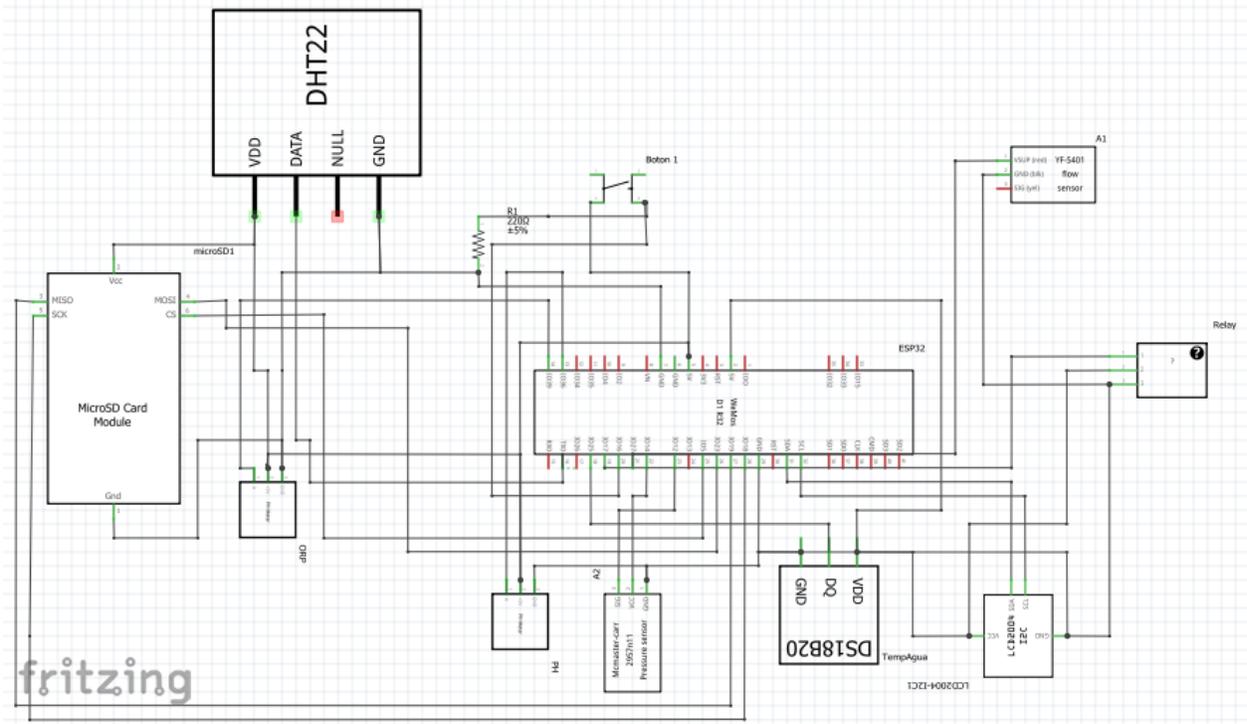


Imagen 36. Esquemático de arquitectura hardware

Software:

Proponemos 3 vistas de usuario en las que podemos distinguir Login, Dashboard y Datos, para las cuales vamos a explicar brevemente la idea de cada una y su interacción conjunta.

En cuanto a la vista del login tenemos algo muy sencillo, un campo como nombre de usuario y otro campo de contraseña, una vez tengamos estos dos campos completados con nuestro usuario proporcionado por el administrador del sistema ya le damos Enter o hacemos un click en el botón azul de Login para entrar a la siguiente vista de usuario.

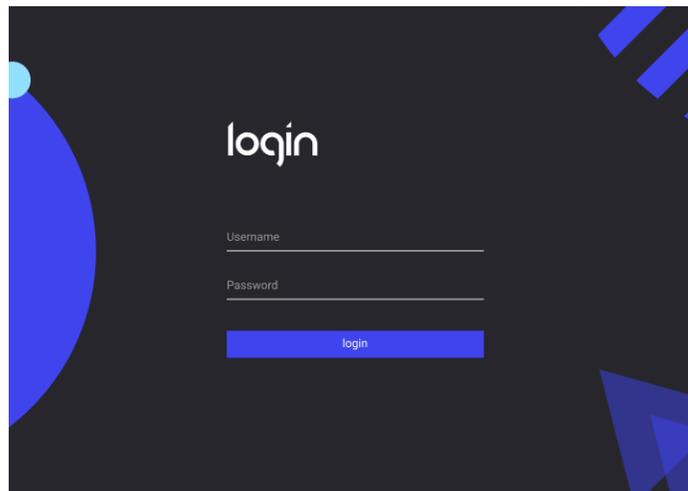


Imagen 37. Maqueta de interfaz de login de usuario para aplicación software

Una vez entramos en la aplicación nos dirigiremos al dashboard donde podemos ver el nombre del usuario conectado, el nombre del dispositivo que estamos visualizando, el dashboard que estamos utilizando con sus estadísticos y botones abajo y en la parte derecha tenemos un par de botones que nos dirigirán tanto a la vista de usuario de datos como a la vista de usuario de dashboard si no estamos ya en ella.

Los botones de la bomba automática y manual serán interactivos y controlarán la activación de un modo u otro de la bomba, en el caso de que la bomba se activara manualmente, el botón de automático pasaría a modo OFF y el de manual pasaría a modo ON, todo esto está diseñado en el código y tiene su propio sistema inteligente para un mejor control de la piscina.



Imagen 38. Maqueta de dashboard para aplicación software

Tanto esta vista de usuario como la anterior de dashboard son accesibles entre ellas, dando la oportunidad de poder visualizar por un lado Widgets interactivos desde los que podemos controlar nuestra piscina y tener una visualización más bonita y por otro lado en la vista de datos tendremos acceso a todo un registro de estadísticas de todos los datos que están recogiendo los sensores. Desde aquí podemos ver claramente un histórico de todo lo que ha estado sucediendo en nuestra piscina desde hace tiempo, si queremos ver una fecha en particular se proporcionará un desplegable con la fecha a seleccionar o una forma de mover el estadístico hasta la fecha solicitada.



Imagen 39. Maqueta de vista de variables para aplicación software

Con todo esto tenemos las vistas suficientes y sencillas para que un cliente pueda tener el control de todo lo que está sucediendo en su piscina. No queremos añadir muchas cosas más para hacer la aplicación más intuitiva aunque a medida que crezca se podrían solicitar opiniones a los clientes para satisfacer las necesidades que sean propuestas.

3.2.4 Desarrollo

El uso de librerías para el ESP32 es una práctica común que permite a los programadores aprovechar el trabajo de otros desarrolladores y utilizar código ya probado y optimizado en otros proyectos.

Pueden proporcionar funciones para realizar operaciones complejas como comunicarse con sensores o controlar dispositivos sin tener que escribir todo el código desde cero, teniendo así la posibilidad de tan sólo algunos parámetros o realizar algún cambio sencillo. Puede ahorrar tiempo y reducir errores de programación y son compatibles con una amplia variedad de placas y periféricos.

A pesar de todos los beneficios tenemos que tener en cuenta que las actualizaciones de las librerías pueden ocasionar problemas en el código y errores que no existían en otro momento, ya sea porque se actualiza o porque se ha quedado obsoleto, por lo que hay que tener cuidado con las actualizaciones automáticas, las actualizaciones de las librerías de las placas y de los periféricos.

A continuación listamos las librerías y la correspondiente versión que vamos a utilizar en el desarrollo de este proyecto.

Librería para conectar la placa ESP32 y aprovechar al máximo sus características:

esp32 by Espressif Systems
Version 2.0.6 **INSTALLED**

Boards included in this package:
WiPy 3.0, WEMOS D1 MINI ESP32, UM TinyPICO, UM PROS3, TTGO T7 V1.4 Mini32, Adafruit QT Py ESP32, UM FeatherS2 Neo, Adafruit ESP32 Feather, BPI-Leaf-S3, Adafruit FunHouse, MH ET LIVE ESP32MiniKit, OLIMEX ESP32-EVB, WEMOS D1 R32, ESP32-S3-Box, ESP32-S3-USB-OTG, TTGO T-OI PLUS RISC-V ESP32-C3, LOLIN D32 PRO, ATmegaZero ESP32-S2, SparkFun LoRa Gateway 1-Channel, Heltec WiFi Kit 32(V3), ProtoCentral HealthyPi 4, LOLIN S2 Mini, Adafruit QT Py ESP32-S2, WT32-ETH01 Ethernet Module, Frog Board ESP32, DOIT ESP32 DEVKIT V1, AI Thinker ESP32-CAM, Department of Alchemy MiniMain ESP32-S2, ESP32-WROOM-DA Module, microS2, LOLIN S2 PICO, NodeMCU-32S, OLIMEX ESP32-PoE, ESP32 FM DevKit, Labplus mPython, ESP32S3 Dev Module, Dongsen Tech Pocket 32, Adafruit Feather ESP32-S2 TFT, OLIMEX ESP32-GATEWAY, ESP32 Wrover Kit (all versions), Deneyap Mini, Bee Motion S3, Heltec WiFi Kit 32, ESP32S2 Native USB, TTGO T7 V1.3 Mini32, SparkFun ESP32 Thing, SparkFun ESP32 Thing Plus C, unPhone 8, S, ODI Ultra v1, XinaBox CW02, LOLIN D32, M5Stack-ATOM, KITS ESP32 EDU, ESP32 PICO-D4, MagicBit, Adafruit Feather ESP32-S3 No PSRAM, MH ET LIVE ESP32DevKIT, WEMOS LOLIN32, Heltec WiFi LoRa 32, ALKS ESP32, Bee Motion Mini, INEX OpenKB, AirM2M CORE_ESP32C3, RedPill(+) ESP32-S3, LOLIN C3 Mini, M5Stack-Core2, Heltec WiFi LoRa 32(V2), BPI-BIT, Watchy, ESP-C3-M1-I-Kit, LoPy, MGBOT IOTIK 32B, ET-Board, Deneyap Kart, Trueverit ESP32 Universal IoT Driver, unPhone 7, Node32s, OLIMEX ESP32-DevKit-LiPo, ESPectro32, TTGO T-Watch, ESP32S3 CAM LCD, Noduino Quantum, WiFiduinoV2, IMBRIOS LOGSENS_V1P1, OLIMEX ESP32-PoE-ISO, uPesy ESP32 Wroom DevKit, Trueverit ESP32 Universal IoT Driver MK II, Bee Motion, TTGO T1, SparkFun ESP32 MicroMod, Onehorse ESP32 Dev Module, Adafruit Feather ESP32-S3 2MB PSRAM, ESP32S2 Dev Module, Nano32, M5Stack-Core-ESP32, OROCA EduBot, ESPea32, Adafruit ItsyBitsy ESP32, ESP32vn IoT Uno, T-Beam, ESP32 Dev Module, ESP32 Wrover Module, UM FeatherS2, Widora AIR, Senses's WEIZEN, Heltec WiFi LoRa 32(V3) / Wireless shell(V3) / Wireless stick lite (V3), Electronic SweetPeas - ESP320, Hornbill ESP32 Dev, M5Stack-FIRE, OROCA ESP32, unPhone 9, UM RMP, Adafruit Feather ESP32 V2, HONEYLemon, TAMC Termod S3, Heltec Wireless Stick Lite, VintLabs ESP32 Devkit, Denky, Deneyap Kart 1A, Turta IoT Node, SparkFun ESP32-S2 Thing Plus, u-blox NINA-W10 series (ESP32), LOLIN S3, Cyttron Maker Feather ALoT S3, WeMos WiFi&Bluetooth Battery, Hornbill ESP32 Minima, Deneyap Mini v2, Franzininho WiFi, IntoRobot Fig, M5Stick-C, Heltec Wireless Stick, ESP32C3 Dev Module, SparkFun ESP32 Thing Plus, DFRobot Firebeetle 2 ESP32-S3, FireBeetle-ESP32, M5Stack-Station, M5Stack-Timer-CAM, uPesy ESP32 Wrover DevKit, KB32-FT, WEMOS LOLIN32 Lite, Microduino-CoreESP32, Piranha ESP-32, Metro ESP-32, Deneyap Kart G, Lion.Bit Dev Board, Connaxio's Espoir, SparkFun ESP32 IoT RedBoard, Adafruit Metro ESP32-S2, Adafruit Feather ESP32-S2, Adafruit QT Py ESP32-C3, TTGO LoRa32-OLED, Adafruit Feather ESP32-S3 TFT, ThaiEasyElec's ESPino32, Franzininho WiFi MSC, LoPy4, Pycorn GPY, MGBOT IOTIK 32A, Deneyap Kart 1A v2, Silicognition wESP32, WiFiduino32, WiFiduino32S3, UM TinyS2, DFRobot Beetle ESP32-C3, Adafruit QT Py ESP32-S3 No PSRAM, M5Stack-CoreInk, D-duino-32, Sonoff DUALR3, XIAO_ESP32C3, CNRS AW2ETH, UM FeatherS3, u-blox NORA-W10 series (ESP32-S3), Adafruit MagTag 2.9", DOIT ESPduino32, Bee S3, UM TinyS3, DPU ESP32

[More info](#)

2.0.7 INSTALL

Imagen 40. Librería ESP32

Librería para conectar la placa Arduino Mega y aprovechar al máximo sus características (aunque no la usaremos para este proyecto, la hemos utilizado en la versión anterior):

Arduino AVR Boards by Arduino
Version 1.8.6 **INSTALLED**

Boards included in this package:
LilyPad Arduino, Arduino NG or older, Linino One, Arduino Uno WiFi, Arduino Uno, Arduino Nano, Arduino Robot Control, Arduino Esplora, Arduino Yún Mini, Arduino Uno Mini, Arduino Mega ADK, Arduino Pro or Pro Mini, Arduino Industrial 101, Arduino Leonardo ETH, Arduino Micro, Arduino Robot Motor, Adafruit Circuit Playground, LilyPad Arduino USB, Arduino Gemma, Arduino Duemilanove or Diecimila, Arduino Mega or Mega 2560, Arduino Leonardo, Arduino Mini, Arduino Fio, Arduino BT, Arduino Yún, Arduino Ethernet

[More info](#)

1.8.5 INSTALL

Imagen 41. Librería Arduino mega

Librería para utilizar el sensor de temperatura del agua DS18B20:

DallasTemperature by Miles Burton <miles@mnetcs.com>, Tim Newsome <nuisance@casualhacker.net>, Guil Barros <gfbarrros@bappos.com>, Rob Tillaart <rob.tillaart@gmail.com>
Version 3.9.0 **INSTALLED**

Supports DS18B20, DS18S20, DS1822, DS1820
Arduino Library for Dallas Temperature ICs

Imagen 42. Librería sensor DS18B20

Librería para utilizar el sensor de temperatura y humedad DHT22:

DHT sensor library by Adafruit

Version 1.4.4

INSTALLED

Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors
 Arduino library for DHT11, DHT22, etc Temp & Humidity Sensors

Imagen 43. Librería sensor DHT22

Librería para utilizar el protocolo de comunicación OneWire:

OneWire by Jim Studt, Tom Pollard, Robin James, Glenn Trewitt,
 Jason Dangel, Guillermo Lovato, Paul Stoffregen, Scott Roberts,
 Bertrik Sikken, Mark Tillotson, Ken Butcher, Roger Clark, Love
 Nystrom

Version 2.3.7

INSTALLED

Access 1-wire temperature sensors, memory and other chips.

Imagen 44. Librería sensor OneWire

Librería para utilizar la tarjeta SD como dispositivo de almacenamiento físico:

SD by Arduino, SparkFun

Version 1.2.4

INSTALLED

Once an SD memory card is connected to the SPI interface of the
 Arduino or Genuino board you can create files and read/write on them.
 You can also move through directories on the SD card.
 Enables reading and writing on SD cards.

[More info](#)*Imagen 45. Librería tarjeta SD*

Librería para poder realizar la conexión al servidor NTP y sincronizar la hora de internet con la placa ESP32:

NTPClient by Fabrice Weinberg

Version 3.2.1

INSTALLED

Get time from a NTP server and keep it in sync.
 An NTPClient to connect to a time server

Imagen 46. Librería NTPClient para ESP32

Librería incluida por defecto en nuestro IDE de Arduino y proporciona las funciones necesarias para conectarse a una red WiFi y enviar/recibir datos a través de ella:

WiFi by Arduino

Version 1.2.7

INSTALLED

With this library you can instantiate Servers, Clients and send/receive
 UDP packets through WiFi. The shield can connect either to open or
 encrypted networks (WEP, WPA). The IP address can be assigned
 statically or through a DHCP. The library can also manage DNS.
 Enables network connection (local and Internet) using the Arduino
 WiFi shield.

Imagen 47. Librería WiFi

Librería para utilizar la memoria EEPROM de la placa:

Esta librería no hay que instalarla ya que viene instalada por defecto cuando instalas Arduino IDE.

Librería para utilizar el sensor de PH de DFrobot en un ESP32 con 3.3V por parte de un usuario de Github pero modificada por nuestra parte para poder calibrar con los buffers cada cierto tiempo y poder ofrecer lecturas correctas con cualquier sonda:

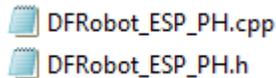


Imagen 48. Librerías sensor PH

La necesidad de tener control del tiempo en el ESP32 también conforma una parte fundamental de nuestro sistema, no podemos depender del manejo del tiempo con el uso de los delay puesto que nos bloquearía el programa hasta que las tareas terminen ese periodo, por lo tanto optamos por el uso de la función millis(), la cual consiste en realizar un cálculo del tiempo que ha pasado respecto al tiempo en el que empiezas a contar, es decir si empiezas a contar desde el segundo 5 y dejas pasar 10 segundos, en el segundo 15 habrán pasado 10 segundos y empezaría a contar de nuevo. Es una manera de solucionar el bloqueo del programa y poder usar los tiempos para medir el uso de variables o funciones, la única desventaja es que millis() como máximo puede tener un valor de 4.294.967.295 o 49 días, y después de esto causará un overflow y se reiniciará la cuenta a 0.

Otra necesidad respecto al tiempo es saber la hora exacta en todo momento y para eso vamos a utilizar el protocolo NTP. En este proyecto requerimos de un manejo del tiempo muy preciso ya que debemos activar la bomba en un determinado instante y saber el momento exacto en el que surja un problema para poder registrarlo y/o actuar inmediatamente, de este modo el uso de un RTC interno es importante. En el ESP32 tenemos un RTC preciso y confiable pero tener que actualizarlo manualmente es un trabajo innecesario y costoso, para esto sincronizamos servidores NTP aprovechando la conexión WiFi y así saber la fecha real actualizada.

NTP es un protocolo de Internet para sincronizar los relojes de los sistemas informáticos, la ventaja de estos servidores viene dada en que son gratuitos y solo tenemos que preocuparnos de tener conexión a Internet.

Mediante la librería Time.h podemos acceder a todas las herramientas para sacar provecho a nuestro RTC interno. Podemos ver su funcionamiento en el código del programa.

3.3. Pruebas

3.3.1 De sistema

Hemos hecho pruebas de cada sensor implementado distintos códigos para comprobar el funcionamiento de los mismos y de qué manera podremos más adelante instalar los sensores con la menor cantidad de cableado posible, lo más optimizado posible (hasta que diseñemos

una placa), con el menor consumo de memoria, menor número de líneas de código posible y lo mejor calibrado que se puede por software.

Para la conexión con Ubidots probamos individualmente cada una de las funcionalidades del sistema como por ejemplo que estamos suscritos a un tema y que podemos publicar también, veremos el ejemplo con la puesta en marcha de la bomba en modo automático y como podemos desactivarla activando el modo manual.

En esta parte podemos ver la conexión MQTT y la suscripción al dispositivo (en este caso se llama esp32) y a las variables (en este caso llamadas “apagado”, “man” y “autom”):

```
Attempting MQTT connection...connected
Subscribing to/v2.0/devices/esp32/apagado/lv
Subscribing to/v2.0/devices/esp32/man/lv
Subscribing to/v2.0/devices/esp32/autom/lv
```

Imagen 49. Muestra conexión de suscripción MQTT

Una vez que nos hemos suscrito nos llega el mensaje del estado de las variables:

```
Message arrived [/v2.0/devices/esp32/apagado/lv] 0.0
apagadoTotal = false
Message arrived [/v2.0/devices/esp32/man/lv] 0.0
manual = false
Message arrived [/v2.0/devices/esp32/autom/lv] 1.0
automatico = true
```

Imagen 50. Muestra conexión de estado de variables MQTT

Y probamos la siguiente secuencia para ver que funciona:

autom encendido -> autom apagado -> autom encendido -> manual encendido

```
Message arrived [/v2.0/devices/esp32/autom/lv] 1.0
automatico = true
Message arrived [/v2.0/devices/esp32/autom/lv] 0.0
automatico = false
Message arrived [/v2.0/devices/esp32/autom/lv] 1.0
automatico = true
Message arrived [/v2.0/devices/esp32/man/lv] 1.0
manual = true
Message arrived [/v2.0/devices/esp32/autom/lv] 0.0
automatico = false
```

Imagen 51. Muestra conexión correcta MQTT



Imagen 52. Muestra tres variables para accionar relés desde aplicación

Podemos ver que cuando activamos el botón manual se nos apaga el automático y así lo haría también con los otros dos si pulsamos el botón de apagado. Todo esto lo explicaremos más adelante con más detalle.

PH: Para el ph calibramos con 2 buffers, buffer de 4 y buffer de 7, en este caso ponemos el ejemplo para ver como se calibra con un buffer de 7 y la salida que produce. Para calibrar insertamos por serial el comando “enterph”, una vez pasados unos segundos insertamos “calph” lo que calculara el valor del buffer y se ajustará, después insertamos “exitph” y ya tendremos nuestra sonda calibrada.

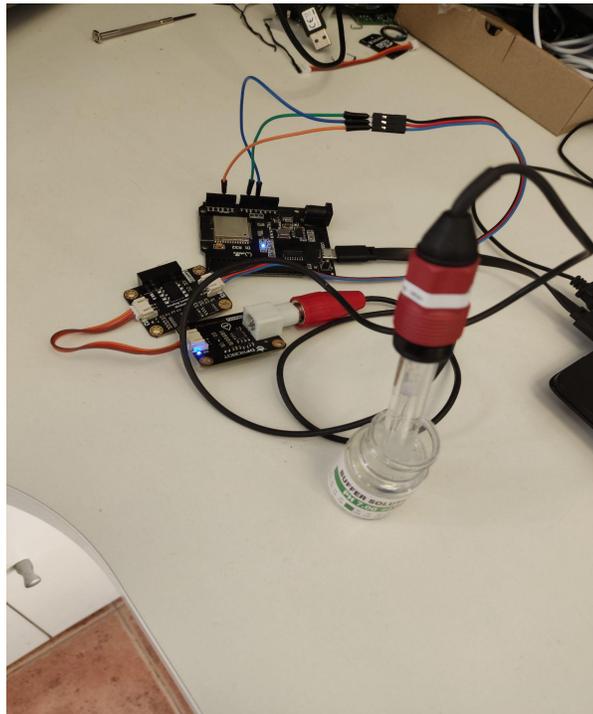


Imagen 53. Hardware y muestra sensor PH

```
voltage:1430.0537
temperature:25.0^C
pH:7.0055
voltage:1430.0537
temperature:25.0^C
pH:7.0055
voltage:1430.0537
temperature:25.0^C
pH:7.0055
voltage:1430.0537
temperature:25.0^C
pH:7.0055
voltage:1435.6934
temperature:25.0^C
pH:6.9669
voltage:1432.4707
temperature:25.0^C
pH:6.9890
voltage:1430.0537
temperature:25.0^C
pH:7.0055
```

Imagen 54. Respuesta serial sensor PH

ORP: Para el Redox no usaremos más que un buffer de calibración, es posible que en un futuro implementemos mejoras en el algoritmo y las mediciones pero por el momento usaremos solo el buffer de 650.

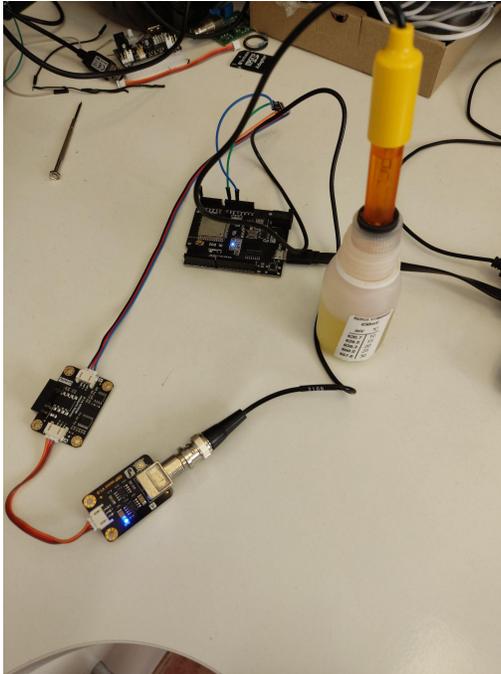


Imagen 55. Hardware y muestra sensor ORP

```
ORP value: 634.99
ORP value: 637.44
ORP value: 647.20
ORP value: 652.09
ORP value: 647.20
ORP value: 650.87
ORP value: 649.65
```

Imagen 56. Respuesta serial sensor ORP

Presión: En cuanto a la medición de la presión podemos apreciar como cuando soplamos suben los psi, en principio la fórmula medía en bar pero hemos ido mejorándola, hemos implementado la conversión a psi y para la versión final añadiremos valor absoluto para que no muestre valores negativos.

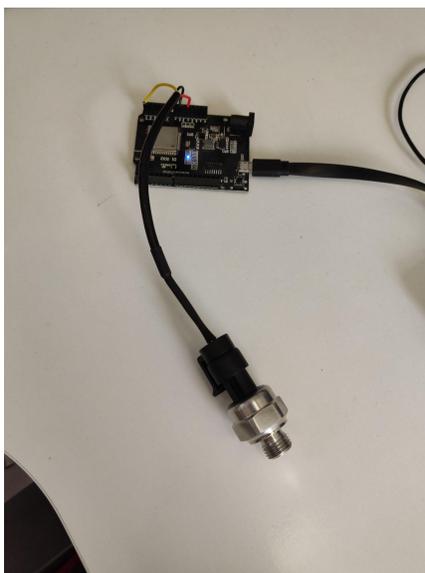
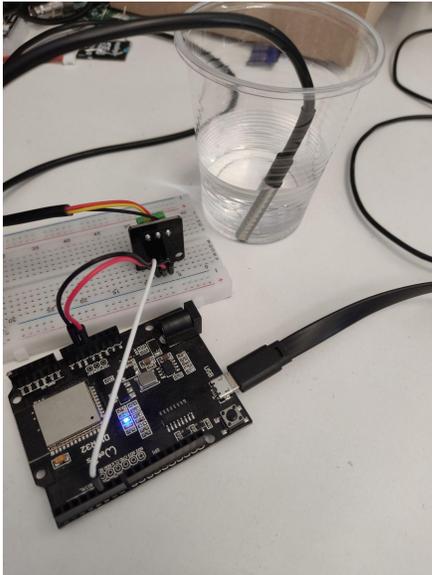


Imagen 57. Hardware sensor de Presión

```
voltageRead: 0.55 | PressureBAR: -0.01 bars | PressurePSI: 0.00 psi
voltageRead: 0.55 | PressureBAR: -0.01 bars | PressurePSI: 0.00 psi
voltageRead: 0.55 | PressureBAR: 0.01 bars | PressurePSI: 0.00 psi
voltageRead: 0.59 | PressureBAR: 0.13 bars | PressurePSI: 1.88 psi
voltageRead: 0.60 | PressureBAR: 0.16 bars | PressurePSI: 2.31 psi
voltageRead: 0.59 | PressureBAR: 0.13 bars | PressurePSI: 1.88 psi
voltageRead: 0.60 | PressureBAR: 0.16 bars | PressurePSI: 2.36 psi
voltageRead: 0.61 | PressureBAR: 0.17 bars | PressurePSI: 2.41 psi
```

Imagen 58. Respuesta serial sensor de presión

Temperatura del agua: Para esta medición tenemos una librería que nos proporciona la resolución que tenemos que utilizar y obtiene la medida mediante el protocolo 1-Wire. Podemos apreciar como baja la temperatura cuando insertamos la sonda en el vaso de agua dado que respecto a la temperatura ambiente, esta está más fría.

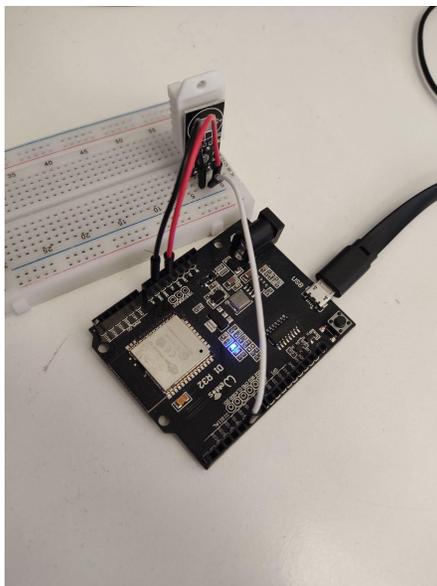


```

Temperatura= 26.31 C
Temperatura= 26.25 C
Temperatura= 24.75 C
Temperatura= 22.87 C
Temperatura= 22.25 C
Temperatura= 22.00 C
Temperatura= 21.94 C
Temperatura= 21.87 C
    
```

Imagen 59. Hardware sensor de temperatura del agua Imagen 60. Respuesta serial sensor de temperatura del agua

DHT22: Este sensor tiene su propia librería y recibe los datos de temperatura y humedad, como podemos ver después de programarlo hemos exhalado un poco sobre el sensor y podemos apreciar cómo aumenta la temperatura y la humedad.



```

Temperatura = 27.00C
Humedad = 68.00%
Temperatura = 27.30C
Humedad = 75.10%
Temperatura = 27.50C
Humedad = 77.80%
Temperatura = 27.70C
Humedad = 78.00%
Temperatura = 27.80C
Humedad = 76.10%
Temperatura = 27.80C
Humedad = 68.70%
    
```

Imagen 61. Hardware sensor temperatura y humedad Imagen 62. Respuesta serial sensor temperatura y humedad

J-OPTION-FLOW (Flujostato): Este flujostato tiene una conexión muy fácil, básicamente es como un botón y tan solo necesitamos una señal si está circulando el agua o si no, de este modo podemos apagar la bomba cuando está en marcha pero el agua no está circulando, esto quiere decir que hay algún problema y tanto con la presión como con el flujostato podemos determinar que la presión sube o que el agua no circula y va a reventar por algún sitio. Con esto podemos ver que seguramente tengamos alguna llave cerrada o nuestro circuito se encuentra obstruido. Enviará 0 cuando está en circulación y 1 cuando está en reposo.

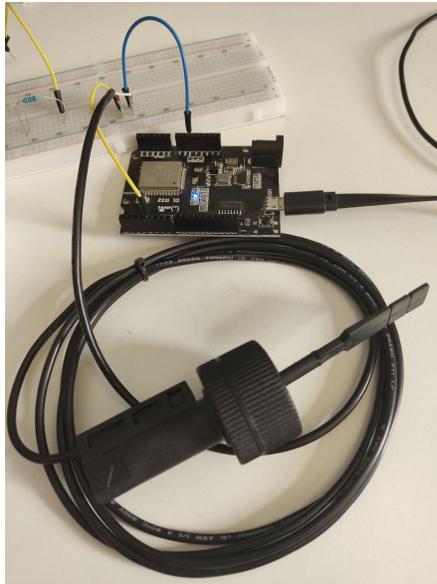


Imagen 63. Hardware flujostato

```
Circulando: 0
Reposo: 1
Circulando: 0
Reposo: 1
Circulando: 0
Reposo: 1
Circulando: 0
Reposo: 1
```

Imagen 64. Respuesta serial flujostato

En cuanto al resto de periféricos o sensores también han sido sometidos a varias pruebas aunque ya forman parte como medio de la muestra de información que hemos recogido, como medio para conseguir la información o como intermediario para utilizar el sistema y poder comunicarnos entre nuestro sistema y nuestra aplicación web. Ejemplos de estos son los relés, la pantalla LCD, el adaptador microSD, entre otras cosas.

3.3.2 De integración de sistemas

Para la integración de sistemas necesitamos que todos los sensores funcionen bien, que los relés funcionen como deben, que no tenemos picos de tensión, que las lecturas son correctas, que el dispositivo está bien conectado al WiFi y que por MQTT estamos realizando tanto publish como subscribe, con todo esto ya tendríamos todo lo necesario para el siguiente paso del proyecto.

Las pruebas necesarias se basan en incluir los sensores poco a poco en la protoboard y a su vez implementar el código por funciones independientes en las que tan solo añadimos la función del sensor y el código para mostrar la información en una función común para mostrar y/o enviar datos a la tarjeta SD y a la aplicación web, aparte obviamente de todas las

constantes, librerías, callbacks, y obtención de tiempo en línea necesario que va surgiendo a lo largo del desarrollo.

Una vez hemos integrado todas las partes en nuestro sistema quedaría algo como muestra la siguiente imagen:

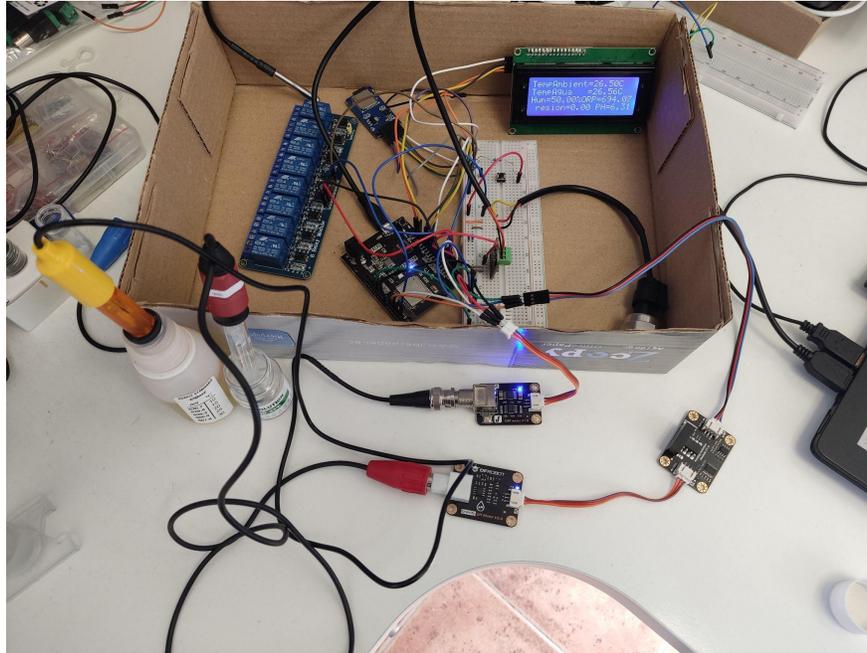


Imagen 65. Sistema integrado en pruebas

En el dashboard de Ubidots podemos comprobar que obtenemos los datos de cada sensor actualizados cada 3 minutos porque es lo que hemos programado previamente para no sobrecargar la aplicación de datos, esto nos indica que la integración se ha realizado bien y todo funciona como es debido.

Esta vista desde la aplicación móvil nos muestra cuándo fue la última actualización de la variable y en qué estado se encuentra, de esta manera tenemos otra opción de vista resumida para todo nuestro sistema.



Imagen 66. Muestra de variables en aplicación 1



Imagen 67. Muestra de variables en aplicación 2

Tanto para el sensor de PH como para el sensor de ORP hay que tener en cuenta que no podemos medir sobre el mismo líquido con ambos sensores al mismo tiempo porque se producen interferencias de señal, este detalle fue el motivo por el que adquirimos la placa aisladora de DFRobot, antes de esto teníamos una variación del PH usando la placa PH4502C de aproximadamente 1.5 hacia arriba y hacia abajo según la hora del día, a las 07:00 estaba 1.5 por debajo y a las 15:00 estaba 1.5 por arriba. Al principio intuía que sería por la temperatura pero al no poder calibrar bien tuve que investigar un poco más, hasta que descubrí este detalle y compré la placa en un intento de solucionar el problema, el cual funcionó. Estuvimos probando el algoritmo que se desarrolló para el funcionamiento de la bomba. Principalmente se encendía en automático por la mañana y por la tarde pero al trabajar en la empresa por la mañana, las lluvias y las bajas temperaturas decidimos modificar el código para solo encenderla por la tarde, más adelante ya probado pondremos resultados del funcionamiento completo pero por el momento parece que funcionaba bastante bien:

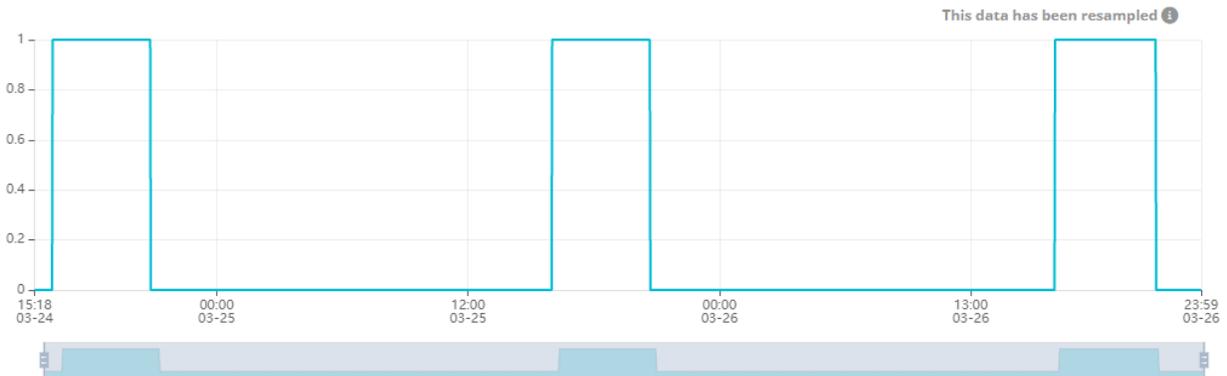


Imagen 68. Muestra variable bomba de la aplicación

Sabiendo que nuestro ESP32 tiene un ADC de resolución 12 y que admite como máximo una entrada de 3.3V diseñamos un divisor de voltaje para poder reducir el voltaje que sale de nuestros sensores y ajustarlo a la entrada.

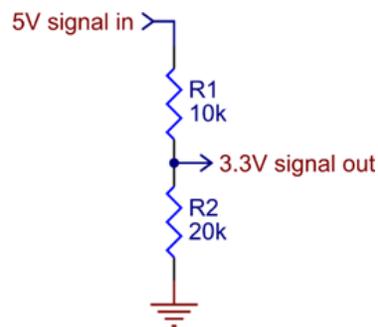


Imagen 69. Divisor de tensión

Durante las pruebas vimos que esto hacía efecto en algunas ocasiones pero normalmente las placas pueden ser alimentadas a 3.3V en lugar de 5V y tenemos el mismo resultado si ajustamos bien la calibración por software, con lo cual decidimos dejar de utilizar el divisor de voltaje y utilizar 5V desde la misma placa para ahorrar en hardware y alimentar algunos sensores como por ejemplo el de ORP.

Al final de las pruebas tenemos una aplicación básica sin personalizar donde podemos ver un resumen de la integración de todos los sensores que vamos a utilizar mejor o peor calibrados pero útil para poder pasar al entorno donde vamos a instalar el sistema final.

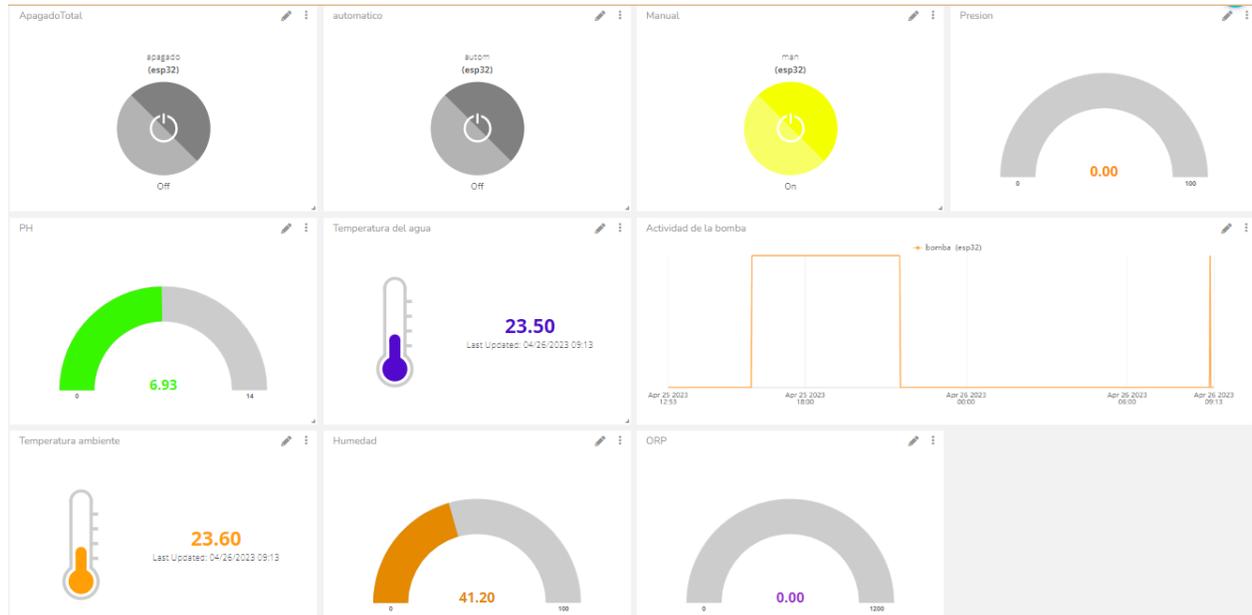


Imagen 70. Dashboard de la aplicación

3.3.3 De volumen

Somos capaces de almacenar todos los datos de los clientes de forma eficiente

Para las pruebas de volumen hemos tratado de optimizar varias cosas.

Por un lado los tiempos de subida de datos a la aplicación web, dado que vamos a usar una aplicación de terceros y nos conectaremos por MQTT a un broker (en este caso Ubidots) en el que tenemos que reducir la cantidad de datos que enviamos al servidor porque la versión gratuita está limitada a 4000 datos por día, y, teniendo en cuenta que enviamos aproximadamente entre 7 y 10 datos de los sensores por cada conexión, vamos a subir a Ubidots aproximadamente la actualización de los datos cada 3 minutos porque el uso de los relés no se actualiza a menos que se pulse el botón de encendido o apagado desde la aplicación. En esta fase de prueba nos conformamos con este volumen de datos para el servidor, pero más adelante habría que estudiar la manera de crear nuestro propio servidor y no limitar en la medida de lo posible la subida de estos datos.

Por otro lado el almacenamiento de los datos en la tarjeta SD no nos preocupa en lo más mínimo porque hemos comprado una tarjeta SD de 32GB, los datos están formateados en un .txt y tan solo vamos a escribir una línea de datos por cada vez que guardemos la información. Este tipo de información no ocupa casi nada de espacio y tendríamos que estar años guardando datos hasta poder agotar su capacidad.

A continuación vemos un ejemplo del manejo de volumen de datos en la siguiente imagen (El dato de la temperatura del agua no coincide porque estamos en el entorno de pruebas y la sonda no estaba sumergida en agua sino encima de la mesa y probablemente estaba siendo calentada por el aire acondicionado).

En la primera imagen vemos el volcado de datos en la tarjeta SD y en la segunda imagen vemos el volcado de datos en la web de Ubidots:

```

-----
Iniciando tarjeta ...
Iniciacion correcta
Fecha: Monday, February 27 2023 11:59:19
Temperatura: 22.70C || Humedad: 26.90% || Presion: 0.72psi || Temperatura agua: 27.62C || MV: 2.57 || PH: 7.19
Bomba: Apagada
Escritura correcta
-----

Enviando los datos a Ubidots:
Temperatura: 22.80
Humedad: 26.70
Temperatura del agua: 27.50
presion: 0.45
MV: 2.57
PH: 7.15
Bomba activa: 0
-----

```

Imagen 71. Apertura y escritura en la SD y envío de datos a Ubidots

```

FECHA: Monday, February 27 2023 13:14:05 Temperatura: 25.00C || Humedad: 24.60% || Presion: 0.00psi Temperatura agua: 27.62C ||MV: 2.56 || PH: 7.22
Bomba: Apagada
FECHA: Monday, February 27 2023 13:16:00 Temperatura: 25.40C || Humedad: 23.80% || Presion: 0.72psi Temperatura agua: 27.75C ||MV: 2.61 || PH: 6.94
Bomba: Apagada
FECHA: Monday, February 27 2023 13:17:32 Temperatura: 24.70C || Humedad: 25.50% || Presion: 0.08psi Temperatura agua: 14.19C ||MV: 2.58 || PH: 7.13
Bomba: Apagada
FECHA: Monday, February 27 2023 13:19:27 Temperatura: 23.90C || Humedad: 24.70% || Presion: 0.14psi Temperatura agua: 13.69C ||MV: 2.58 || PH: 7.13
Bomba: Apagada
FECHA: Monday, February 27 2023 13:21:27 Temperatura: 23.70C || Humedad: 23.30% || Presion: 0.40psi Temperatura agua: 13.75C ||MV: 2.57 || PH: 7.17
Bomba: Apagada
FECHA: Monday, February 27 2023 13:23:27 Temperatura: 24.20C || Humedad: 23.60% || Presion: 0.00psi Temperatura agua: 13.94C ||MV: 2.57 || PH: 7.16
Bomba: Apagada
FECHA: Monday, February 27 2023 13:25:26 Temperatura: 23.50C || Humedad: 22.40% || Presion: 0.14psi Temperatura agua: 14.19C ||MV: 2.58 || PH: 7.13
Bomba: Apagada
FECHA: Monday, February 27 2023 13:27:27 Temperatura: 22.60C || Humedad: 22.60% || Presion: 0.35psi Temperatura agua: 14.38C ||MV: 2.59 || PH: 7.06
Bomba: Apagada
FECHA: Monday, February 27 2023 13:29:27 Temperatura: 22.80C || Humedad: 23.80% || Presion: 0.45psi Temperatura agua: 14.50C ||MV: 2.59 || PH: 7.09
Bomba: Apagada
FECHA: Monday, February 27 2023 13:31:26 Temperatura: 23.00C || Humedad: 23.90% || Presion: 0.14psi Temperatura agua: 14.75C ||MV: 2.58 || PH: 7.09
Bomba: Apagada
FECHA: Monday, February 27 2023 13:33:27 Temperatura: 22.50C || Humedad: 22.80% || Presion: 0.14psi Temperatura agua: 14.88C ||MV: 2.58 || PH: 7.13
Bomba: Apagada
FECHA: Monday, February 27 2023 13:35:27 Temperatura: 22.80C || Humedad: 23.50% || Presion: 0.14psi Temperatura agua: 15.06C ||MV: 2.59 || PH: 7.08
Bomba: Apagada
FECHA: Monday, February 27 2023 13:37:26 Temperatura: 24.00C || Humedad: 23.10% || Presion: 0.45psi Temperatura agua: 15.38C ||MV: 2.59 || PH: 7.08
Bomba: Apagada
FECHA: Monday, February 27 2023 13:39:27 Temperatura: 23.60C || Humedad: 22.10% || Presion: 0.51psi Temperatura agua: 15.50C ||MV: 2.58 || PH: 7.10
Bomba: Apagada
FECHA: Monday, February 27 2023 13:41:27 Temperatura: 22.80C || Humedad: 22.90% || Presion: 0.30psi Temperatura agua: 15.69C ||MV: 2.57 || PH: 7.16
Bomba: Apagada
FECHA: Monday, February 27 2023 13:43:27 Temperatura: 22.70C || Humedad: 23.70% || Presion: 0.88psi Temperatura agua: 15.81C ||MV: 2.59 || PH: 7.06
Bomba: Apagada
FECHA: Monday, February 27 2023 13:45:27 Temperatura: 23.00C || Humedad: 24.30% || Presion: 0.67psi Temperatura agua: 16.00C ||MV: 2.59 || PH: 7.09
Bomba: Apagada
FECHA: Monday, February 27 2023 13:47:26 Temperatura: 23.60C || Humedad: 24.50% || Presion: 0.51psi Temperatura agua: 16.12C ||MV: 2.58 || PH: 7.10
Bomba: Apagada

```

Imagen 72. Muestra de escritura en el fichero de la SD

4 Resultados

4.1 Migración al entorno de producción

Una vez montado el sistema en el entorno de pruebas tenemos que probar como funciona todo en el entorno de producción y esto es uno de los cambios más importantes en el desarrollo de nuestro producto ya que los valores de los sensores varían bastante en función de las condiciones meteorológicas, la señal recibida, el control de los picos de corriente, la presión

dentro de las tuberías, la conexión de todas las tuberías e instalación de todos los sensores dentro de ellas y la disponibilidad del tiempo para estar atento a cualquier cambio que se pueda producir entre otras cosas.

Hay que tener en cuenta que estamos manejando corriente, presión y situaciones que pueden ser peligrosas si no medimos bien nuestros pasos, así que lo importante de todo esto es tener bien claro lo que estamos haciendo e ir paso a paso de forma segura.

Pongamos un ejemplo: Si instalamos un sensor de presión para comprobar si el filtro está obstruido, la única forma de comprobar si el sensor está midiendo bien es obstruir la tubería o el filtro y estar atento a la situación que se puede dar, o esperar a que esto ocurra de manera natural y ver que pasa. Para todo esto existen las pruebas y más de una vez he tenido que parar el sistema o reemplazar alguna cosa por otra por problemas que han surgido, y, pese a que esto pueda parecer un problema, en ocasiones es muy útil para averiguar cosas que no habías tenido en cuenta o tener la ocasión de investigar un poco más y obtener alguna solución más adecuada ante determinados problemas

Procedemos a la instalación de nuestro sistema de manera completa aunque hemos ido paso a paso incorporando sensores y comprobando tuberías, enchufes, conexión WiFi, cableado y presión.

4.2 Manual de usuario

Hardware:

La utilización de la parte hardware de nuestro sistema es muy sencillo, dado que necesitamos que un profesional o personal cualificado instale nuestros sensores y sistemas a la electricidad, así como los medios que vamos a utilizar (bomba, filtro de arena, tuberías, etc..), no hay que hacer mucho más ya que va todo automatizado por software y manejado desde ahí, tan solo tenemos la opción de apagar el sistema de la bomba desde un botón insertado en la protoboard por si tenemos que hacer una parada de emergencia o mantenimiento.

Software:

En nuestro caso tenemos una cuenta gratuita del software Ubidots de terceros que nos da acceso hasta 10 variables y una cantidad limitada de datos que podemos guardar como hemos explicado anteriormente, escogimos por 1 mes la opción de 50€ mensuales gratuita de prueba para ver el potencial de esta aplicación y la verdad es que es bastante buena, no obstante puesto que no es el objetivo principal de este trabajo, nos limitaremos a recoger la cantidad que podamos y optimizar lo máximo posible.

Indicaremos los pasos que hemos seguido para tener nuestro dispositivo conectado y utilizado como administrador, cambiando que el usuario final no tendría el acceso a cambiar ciertas cosas que podemos limitar, ya sea el nombre del dispositivo, nombre de las variables y funcionamiento de las variables que le permitamos ver. Tenemos que otorgar permisos si queremos que el cliente sea capaz de activar, desactivar y modificar variables para la bomba por ejemplo.

4.2.1 Instalación de la aplicación

Hardware:

Como hemos comentado en la descripción, necesitaremos un profesional cualificado o una persona responsable y capacitada para realizar la instalación de todos los periféricos sobre los que vamos a utilizar nuestros sensores, por no hablar de que la piscina tiene que estar funcional.

Por lo que vamos a recopilar una serie de imágenes con las que podemos ver de forma más clara cómo vamos a ir finalizando la instalación de todo nuestro sistema en general:

- Imagen de la piscina:



Imagen 73. Piscina Azulena donde se realizan las pruebas

- Imagen de sala de depuración:



Imagen 74. Sala depuradora donde se instala el sistema

- Imagen de la alimentación del sistema y adaptador WiFi:



Imagen 75. Adaptador WiFi

- Imagen de la bomba:



Imagen 76. Bomba en sala de depuración

- Imagen de los sensores PH y ORP(de este lado usaremos solo ORP)



Imagen 77. Sensores de PH y ORP instalados en tubería

- Imagen del sensor de presión:

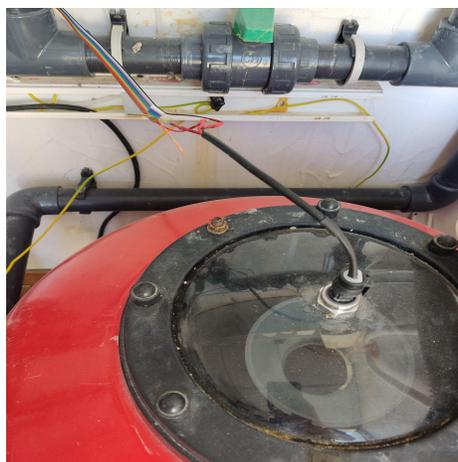


Imagen 78. Sensor de presión instalado en filtro de arena

- Imagen de los sensores de PH, Temperatura del agua y Flujostato:



Imagen 79. Sensores de PH, Vaina con sensor de temperatura del agua y flujostato

- Imagen del sistema final:



Imagen 80. Sistema final

Software:

En nuestro caso tenemos una cuenta gratuita que nos da acceso hasta 10 variables y una cantidad limitada de datos que podemos guardar como hemos explicado anteriormente, escogimos por 1 mes la opción de 50€ mensuales gratuita de prueba para ver el potencial de esta aplicación y la verdad es que es bastante buena, no obstante puesto que no es el objetivo principal de este trabajo, nos limitaremos a recoger la cantidad que podamos y optimizar lo máximo posible

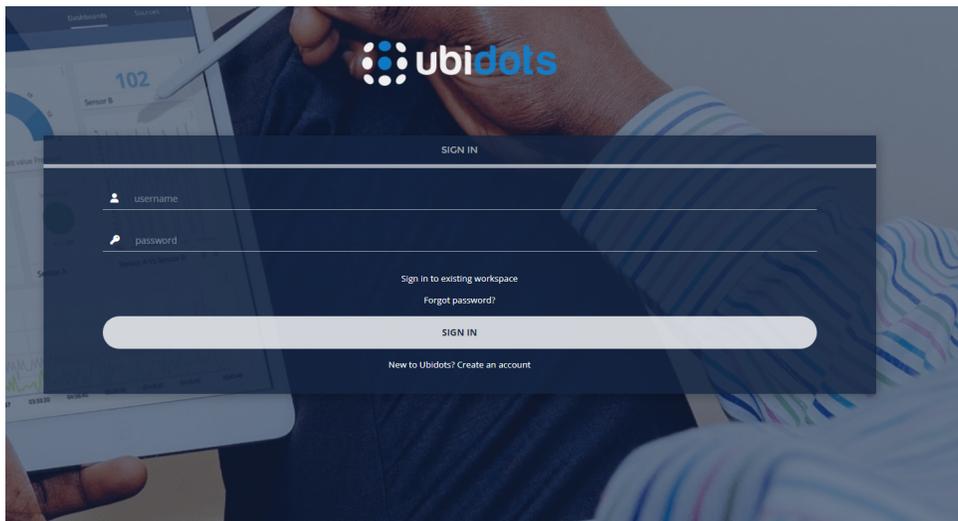


Imagen 81. Inicio de sesión en aplicación

Independientemente del cliente, nosotros utilizaremos los token para conectarnos con el dispositivo que hemos instalado o vamos a instalar, el token se configurará en el software de nuestro sistema y será administrada por nosotros, el cliente no se va a enterar de todo esto, tan solo le proporcionaremos un usuario y una contraseña para poder acceder a la aplicación y poder gestionar su sistema.

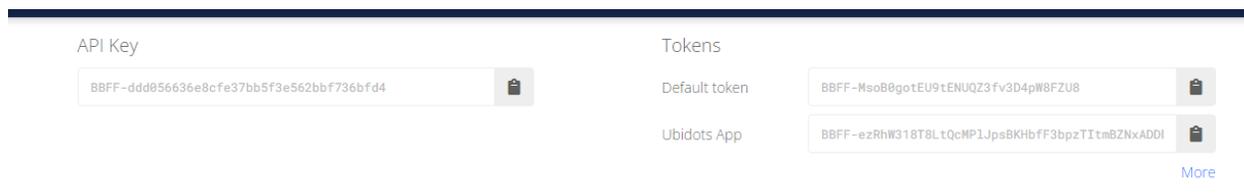


Imagen 82. Muestra de variables en aplicación

La sencillez de la instalación del software hace posible que sea accesible a todo tipo de públicos, prácticamente el cliente tan solo tiene que entrar en su cuenta y disfrutar del servicio. Después de esto, desde la pantalla de login el cliente ya podrá acceder a su aplicación y ver los parámetros de su piscina como veremos en el siguiente apartado.

4.2.2 Explotación

En cuanto a la explotación de la aplicación ya teniendo el hardware montado y el software desarrollado tenemos que dedicarnos poco más que a observar los parámetros de nuestra piscina, todo está programado para ser automatizado excepto los botones que podremos observar en nuestro dashboard, tenemos la opción de bomba apagada para poder realizar tareas de mantenimiento y apagar la bomba sin oportunidad de que esta se encienda, un botón de bomba manual para que se mantenga encendida siempre y cuando este botón lo esté también y la opción de automático que encenderá la bomba a las 8:00 para que circule el agua y a las 8:10 realizar el autoprogramado del tiempo necesario que la bomba de la piscina estará activa en función de la temperatura del agua. Explicándolo un poco mejor hay que tener en cuenta que en el código existe una funcionalidad añadida y es que, por prioridad si el botón de la bomba apagada está encendido nada podrá activarse, ni el botón de manual ni el automático, si dicho botón está apagado entonces tendremos la opción de encender manual o automático, pero pasa igual con estos dos, si el botón de automático está encendido y se pulsa el manual o apagado, éste se apagará y se quedará el otro encendido, por orden de prioridad.

Damos por hecho que el cliente ya tiene su aplicación y los datos correctamente verificados para poder acceder.

En la siguiente imagen veremos cómo tenemos la bomba funcionando completamente las horas que necesita en función de la temperatura y las horas que le hemos programado anteriormente:

monitorPEP



Imagen 83. Muestra de variable actividad de la bomba en aplicación

Cabe añadir que toda la recogida de datos está automatizada pero las labores de mantenimiento se tienen que seguir realizando, la limpieza del filtro, la inyección del cloro, la inyección del ph, la limpieza de la piscina, la recogida de la tarjeta SD, etc... Teniendo en cuenta que en nuestro caso ya tenemos instalada la barrendera, el skimmer, el fondo y la impulsión que se activarán manualmente con llave pero que entrarán en funcionamiento cuando la bomba se encienda.

Ubidots nos ofrece una aplicación móvil desde la cual podemos acceder a nuestro dashboard pero no nos entretendremos en explicar este apartado ya que pese a ser una herramienta muy útil para el desarrollo nos hemos adaptado al sistema vía web.

Las vistas de usuario final quedarían como en las siguientes imágenes:

Así quedaría la vista de la selección del dispositivo:

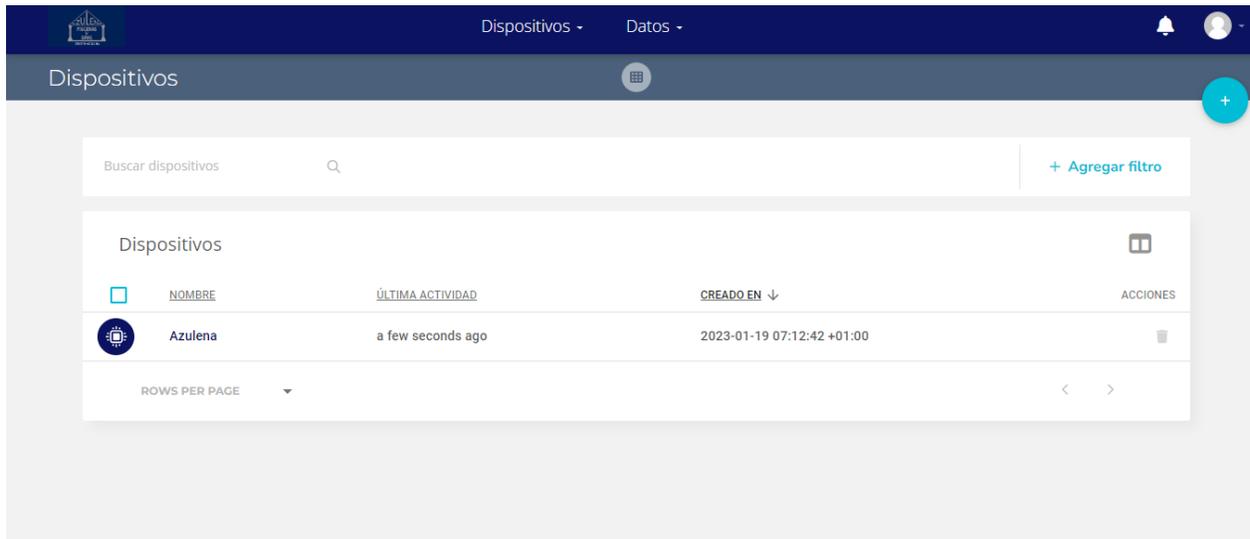


Imagen 84. Muestra de dispositivo a elegir en aplicación

Así quedaría la vista del dashboard del dispositivo seleccionado, en nuestro caso “Piscina Azulena”:

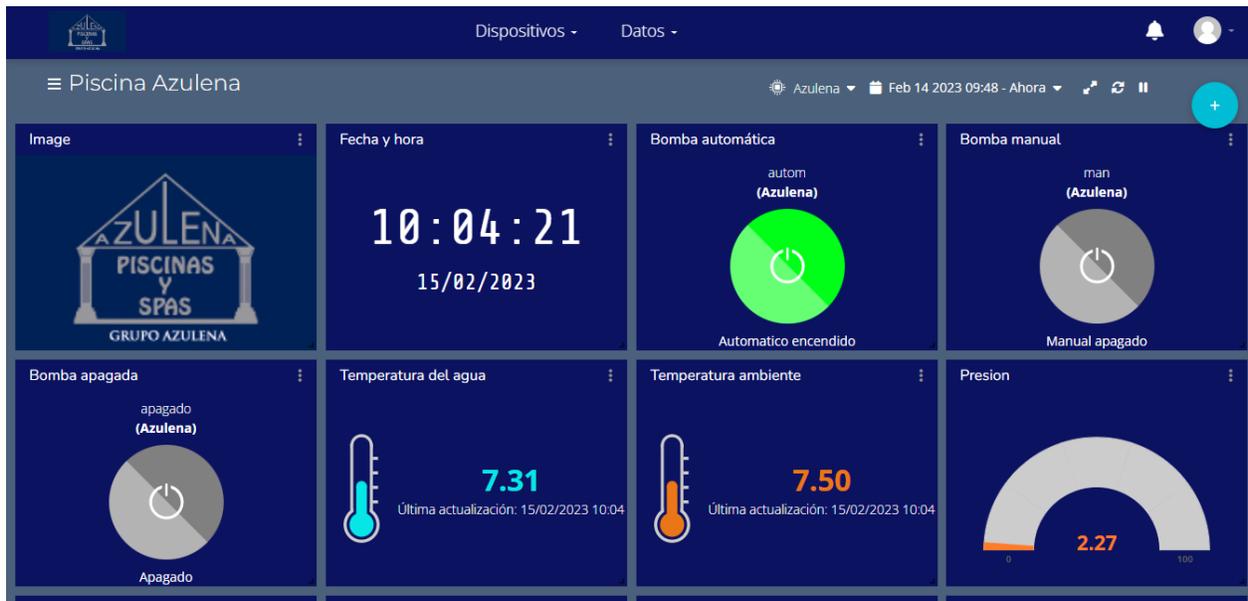


Imagen 85. Muestra del dashboard de la aplicación 1



Imagen 86. Muestra del dashboard de la aplicación 2

Vista de selección de la variable a explorar:

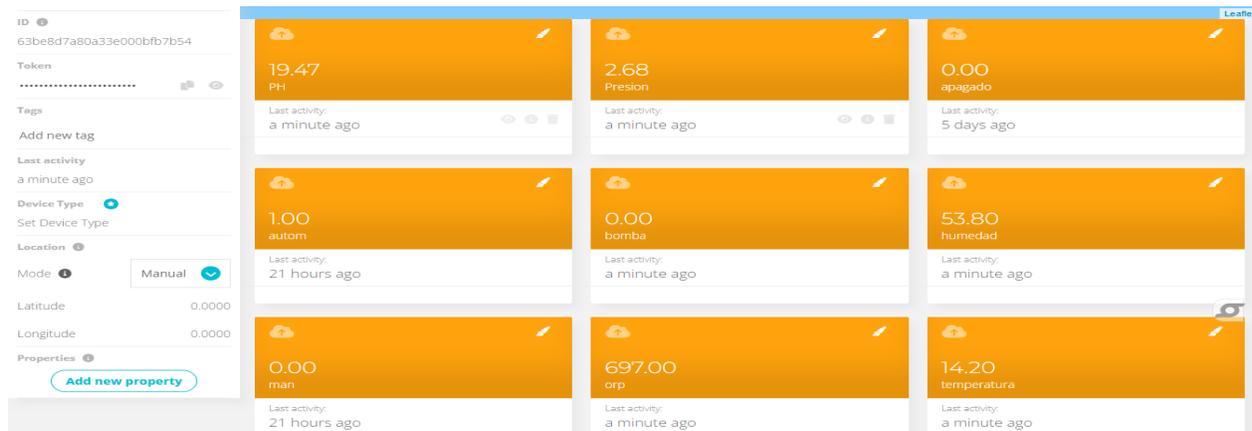


Imagen 87. Muestra de selección de variable en aplicación

Vista de usuario de variable elegida:



Imagen 88. Muestra de variables elegida en aplicación

Ya hemos visto todos los detalles del sistema final, la recogida y muestra de datos, y el sistema final instalado en todo su conjunto, por lo demás no queda más que disfrutar de su aplicación.

5 Conclusiones

5.1 Conclusiones personales

A modo de cierre de este trabajo, podemos afirmar que hemos cumplido notoriamente con los objetivos establecidos dado que hemos podido desarrollar el sistema planteado con la finalidad de recoger datos y tanto guardarlos como poder verlos en “casi” tiempo real.

Hemos aprendido muchísimas cosas en el transcurso de este proyecto, una de las más importantes es que el trabajo y la constancia hacen que mejores día a día y si no te rindes al final eres capaz de conseguir prácticamente lo que te propongas, tecnologías nuevas que no resultaban familiares, sensores con los que hemos tenido que estudiar los parámetros leídos para poder entender cómo recoger los datos y de qué dependen sus variaciones y, en nuestro caso, dado que este proyecto a sido probado en una empresa real, hemos sido capaces de entender, comentar y hablar con profesionales sobre el tema, los cuales me han ayudado a entender lo que es importante y centrarme mejor en ello.

En nuestro caso dos de los parámetros más importantes serían el PH y Redox, dos variables muy importantes para mantener una piscina limpia y equilibrada, descubrir esto me llevó tiempo y poder ajustar los parámetros a su calibración ideal es algo costoso, por lo que con esto podemos decir que también hemos cometido algunos tipos de errores. Los errores más notorios serían el comienzo de nuestro proyecto con el uso del microcontrolador Arduino, el cual como hemos explicado anteriormente, es más caro, no tiene conexión inalámbrica por sí mismo y no necesitamos tantos puertos, por lo que la opción del ESP32 era más adecuada, otros como la compra de la placa PH4502C, durante días, incluso semanas, estuvimos intentando calibrar y analizar las lecturas de esta placa cuando en realidad era una placa defectuosa y que en realidad sus lecturas eran erróneas. Referente al código, hay muchas cosas que depurar y problemas que han ido surgiendo a lo largo del tiempo, puesto que con el cambio de estaciones, las condiciones climáticas cambian, hay apagones de luz y diversos factores que te hacen valorar situaciones que no tenías previstas.

Tanto profesional como personalmente considero que he aprendido a ser autosuficiente y capaz de trabajar en un entorno competente en el mundo laboral, todo con ayuda de la carrera que he estudiado (Grado en Ingeniería Informática), el progreso durante estos años ha generado un importante cambio en mi manera de ver las cosas respecto a los trabajos importantes y las capacidades que tenemos de afrontar retos que nos puedan surgir. No me arrepiento de haber estudiado esta carrera, te prepara para situaciones muy complicadas donde tienes que salir airoso por tus propios medios, pensar como ingeniero, innovar, anticipar problemas que puedan surgir y una gran variedad de cualidades más que te definen como alguien confiable en cuestiones relacionadas con la ingeniería.

Si bien es verdad que en mi carrera he estudiado tanto parte software como hardware, nunca puedes saberlo todo, pero es verdad que a nivel práctico han sido un poco escasos los conocimientos que tenía en relación con el trabajo que he realizado, no estaría de más invertir

más horas en prácticas y un poco menos en teoría, sobre todo en aquellos que nos especializamos más en el entorno del hardware.

Seguiría desarrollando este proyecto durante mucho tiempo, si bien es verdad que entra en juego la necesidad de trabajar en equipo junto con personas que se especialicen en otros campos, podría decir varios pero los dos que considero más importantes serían un Ingeniero Químico para poder entender mejor el estudio de las variables y los cambios que realizan los productos químicos que debemos utilizar y un Ingeniero Electrónico que fuera capaz de realizar tanto el análisis de las corrientes que estamos utilizando y su estabilidad, el montaje de una placa y la supervisión de los montajes en general. De esta manera consideraría una buena opción el poder desarrollar junto a una buena empresa, un producto de calidad que sea funcional y óptimo para su propósito.

Si una cosa me ha enseñado la empresa con la que he colaborado, es que un sistema como este está poco investigado dada la alta dificultad de investigación y desarrollo que esta conlleva, si bien es verdad que los clientes querrían un sistema como este, la realidad es que al final sería un producto caro que no mucha gente se puede permitir, y de poder permitírselo, hay demasiados factores que analizar y un mantenimiento que llevar a cabo. Al final son muchas cosas por las que imagino que las empresas grandes han desistido en desarrollar un sistema bastante más complejo en este campo.

5.2 Futuras líneas de desarrollo

- Incremento del número de parámetros (sensores) a monitorizar: TDS (Total de sólidos Disueltos), TAC (alcalinidad total), turbidez (floculación), fosfatos, salinidad, metales (Co, Fe), estabilizador, oxígeno disuelto, conductividad eléctrica.
- Conexiones adicionales inalámbricas del sistema (BT, LoRa).
- Implementación de un sistema de alarmas por e-mail.
- Integración de una cámara ESP32 cam.
- Cambio de interfaz de pantalla a TFT Nextion.
- Realización de un manual de uso para la aplicación que se acceda mediante lectura de etiqueta QR en nuestro producto final, por enlace proporcionado al cliente o desde el área de cliente dentro de la web o aplicación.
- Digitalizar y automatizar con nuestro sistema el uso de la barrendera, la impulsión, el fondo y el skimmer.
- Creación de una placa integrada para realizar todas las conexiones soldadas y sin cables.
- Diseño y creación de una caja para aislar el sistema de la intemperie a medida.
- Añadir un lector de huellas para la manipulación del sistema físico tan solo por parte de los profesionales de mantenimiento.
- Sustituir entrada serial por botones para la calibración de PH y ORP.
- Añadir control de una bomba de calor en remoto.

- Añadir un sensor detector de presencia para activar y apagar el LCD cuando detecte o no la presencia de alguien enfrente del sistema.

6 Bibliografía

- [1] Cuesta Frau, David. (2021). Apuntes de la asignatura Internet de las cosas. Escuela Politécnica Superior de Alcoy. Grado en ingeniería informática.
- [2] Miró Orozco, Ignacio. (2017). Apuntes de la asignatura Tecnología de Computadores. Escuela Politécnica Superior de Alcoy. Grado en ingeniería informática.
- [3] Lucid Software. (2023). Lucidchart [Software para diagramas de flujo], consulta: [online](#)
- [4] Isabel Lopez. (2023). Connect an ESP32-DevKitC to Ubidots over MQTT. consulta: [online](#)
- [5] ESP32 NTP Client-Server: Get Date and Time (Arduino IDE). (2020). Sincronizar tiempo desde NTP server. consulta: [online](#)
- [6] Ubidots. (2023). Ubidots [Aplicación web IoT y Aplicación móvil v1.1.1 IoT]. consulta: [online](#)
- [7] Gravity: Analog ORP Sensor Meter for Arduino. (2023). Product Wiki. consulta: [online](#)
- [8] Gravity: Analog pH Sensor/Meter Kit V2. (2023). Product Wiki. consulta: [online](#)
- [9] Gravity: Analog Signal Isolator. (2023). Wiki. consulta: [online](#)
- [10] GreenPonik/DFRobot_ESP_PH_BY_GREENPONIK. (2019). Librería calibración para PH. consulta: [online](#)
- [11] Fritzing Software. (2023). Fritzing v0.9.3 [Software para creación de arquitectura hardware y esquemas eléctricos. consulta: [online](#)
- [12] Figma Software. (2023). Figma [Software para creación de maquetas]. consulta: [online](#)
- [13] Gastón C. Hillar. (abr 2017). MQTT Essentials - A Lightweight IoT Protocol. Packt Publishing Ltd. consulta (muestra gratuita): [online](#)
- [14] Juan A. Villalpando. (2021). Wemos D1 R32 ESP32. Instalación. IDE Arduino. consulta: [online](#)

- [15] ORP de una piscina: qué es y cuáles son los niveles óptimos. (Mar 25, 2023). Cloración salina. Mantenimiento. consulta: [online](#)
- [16] Todo lo que necesitas saber sobre el pH en la piscina. (2023). Trucos, Consejos e Ideas. consulta: [online](#)
- [17] Soloelectronicos. (Ago 21, 2022). Como gestionar una SD desde un ESP32. consulta: [online](#)
- [18] Eloscoscopio. (2023). Tutorial de relés para Arduino, ESP8266 y ESP32. consulta: [online](#)
- [19] Rafael Lozano. (Jun 27, 2021). Esp32 y pantalla lcd i2c. consulta: [online](#)
- [20] Ernesto Tolocka. (Jul 4, 2022). Tutorial: Cómo utilizar el DHT22 con ESP8266 y ESP32 en Arduino. consulta: [online](#)
- [21] Solectroshop. (2023). Tutorial del sensor de temperatura DS18B20 con ESP32 - servidor web. consulta: [online](#)

7 Acrónimos

- SBC: Single Board Computer
- SPL: Sound Pressure Level
- MQTT: Message Queue Telemetry Transport
- IDE: Integrated Development Environment
- NTP: Network Time Protocol
- FPGA: Field Programmable Gate Array
- UART: Universal Asynchronous Receiver-Transmitter
- SPI: Serial Peripheral Interface
- I2C: Inter-Integrated Circuit
- IoT: Internet Of Things
- LoRa: Long Range
- SoC: System on a Chip
- ORP: Oxidation-Reduction Potential
- PH: Potencial de hidrógeno
- SBC: Single Board Computer

8 Anexos

8.1 Código

En este apartado vamos a adjuntar un documento en el que incluiremos todo el código referente a este proyecto, no incluiremos el código de las librerías ya que hemos mostrado antes la gran mayoría y son públicas.

Se trata de código escrito en lenguaje C++, usado en Arduino IDE, compilado y ejecutado en el mismo:

<https://pastebin.com/21pn3g0N>