

Proyecto final

Detección de Fugas y Monitoreo de Consumo en Sistemas de Líquidos mediante IoT

Alumno: Fagundez Eduardo
Docente: Santiago Chopitea
División 4to año
27ABR2025

El agua es un recurso único

El agua es nuestro recurso máspreciado, un “oro azul” al que más de 2.000 millones de personas no tienen acceso. No solo es esencial para la supervivencia, sino que también desempeña un papel sanitario, social y cultural en el seno de las sociedades humanas.

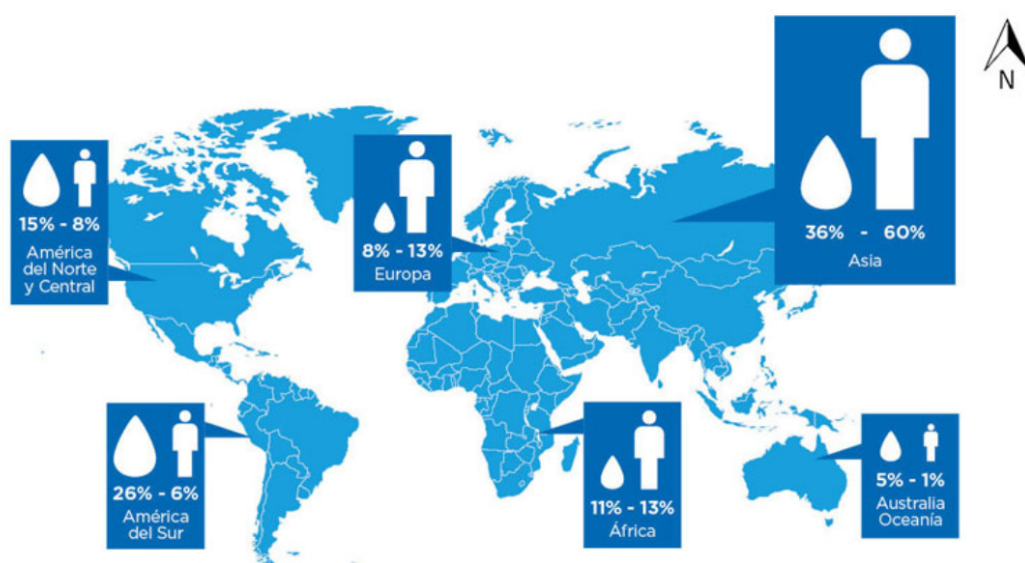


fig. 1.0 Relación entre disponibilidad de agua y población por continente

El agua es un recurso único e insustituible cuya cantidad es limitada. Al ser fundamental para la vida, las sociedades y las economías, presenta múltiples valores y beneficios, pero, al contrario que con la mayoría de los recursos valiosos, se ha comprobado que su valor “real” resulta muy difícil de determinar.

El Informe Mundial sobre el Desarrollo de los Recursos Hídricos de 2021 titulado “El valor del agua” examina el estado actual y los retos de la valoración del agua en distintos sectores y desde diferentes perspectivas e identifica las formas en las que se puede fomentar dicha valoración como una herramienta que contribuya a alcanzar la sostenibilidad.

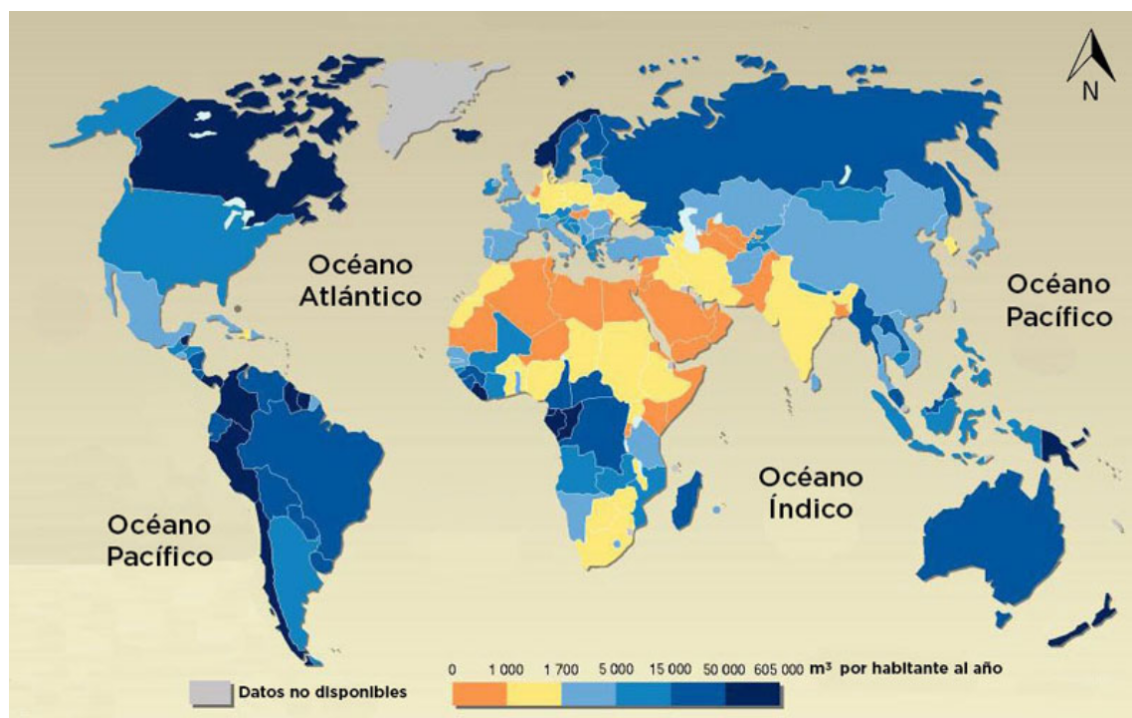


fig. 1.1 consumo de agua potable por continente

Disponibilidad de agua en el mundo por país

Se estima que una canilla que gotea pierde en promedio 46 litros de agua en un día; si la pérdida no es por gota, sino que es por hilo de agua la pérdida es mucho mayor y ni hablar si es un caño roto bajo piso. Tanto los usuarios residenciales como las instituciones públicas que disponen de facturación del servicio empleando medidores de agua en tiempo real, no pueden calcular con facilidad cuánto dinero pierden por pérdidas de agua. Sin duda, el mecanismo conductual del usuario es un factor determinante en el ahorro del agua potable. El uso del agua debe ser racional, justa y la tecnología debería apuntar a este

ÍNDICE GENERAL

RESUMEN	2
El agua es un recurso único	2
Disponibilidad de agua en el mundo por país	3
ÍNDICE GENERAL	4
INTRODUCCIÓN	6
¿Qué es una fuga de agua?	6
Fugas que se reportan a nivel internacional	6
Contaminación del agua potable en China	6
Brote de E. coli en Canadá	6
Caso de Aguas de La Habana, Cuba	6
Caso de la minería en Chile	7
Experiencia Laboral: Gestión de Incidentes en el Consumo de Recursos Hídricos	8
OBJETIVO	9
DISEÑO DEL PROTOTIPO	10
DESARROLLO Y CONSTRUCCIÓN	11
Diseño de las fuentes reguladas 5vdc	11
Transformador 220v-12v	11
Fuente de potencia y control	11
Puente de diodos	11
Capacitores de filtrado	13
Regulador LM7805	15
Etapa de potencia y circuito de control	16
Optoacoplador PC817	16
Transistor 2N3904	17
Relé SRD-05VDC-SL-C	19
Diodo 1N4007	21
Fuente de lógica	23
NodeMCU ESP32S	23
Sensor ultrasónico hc-sr04	24
Funcionamiento	25
Sensor de flujo YF-S401	26
Display tft 3.5	27
Diagrama de conexión con esp32	28
Conclusión	31
Consideraciones Finales	31
Bibliografía	32
Anexo I: Software	33
Anexo II: esquemático, modelo 3D, gerber y PCB	34
I. Esquemáticos	34

II. Modelo 3D	35
V. Gerber	37
IV. PCB	40
Anexo III: Código para control y monitoreo, envío a la nube y uso de periféricos	42
Anexo IV: Alta en plataforma IoT	65
Internet de las cosas (IoT)	65
¿Por qué es tan importante el Internet de las Cosas?	65
Pasos en Thingspeak	65
Anexo V: Presupuesto	72
Anexo VI: Armado	73

INTRODUCCIÓN

¿Qué es una fuga de agua?

Es la pérdida no deseada de agua de una tubería, grifo, cisterna u otro sistema de distribución. Puede ocurrir por diversas razones, como el desgaste de materiales, roturas, presión excesiva o defectos en las conexiones. Las fugas pueden ser pequeñas, como un goteo constante en un grifo, o grandes, como una tubería rota que causa inundaciones. Detectarlas a tiempo es clave para evitar desperdicio de agua, daños materiales y costos elevados en la factura del servicio.

Fugas que se reportan a nivel internacional

Podría enumerar infinitos casos, pero cito cuales creo exponen cómo la falta de detección temprana de pérdidas de agua puede generar impactos económicos y sociales significativos.

Contaminación del agua potable en China

En abril de 2014, la ciudad de Lanzhou enfrentó una grave crisis de contaminación del agua potable. Se descubrió que el agua del grifo contiene hasta 20 veces el límite aceptable de benceno, un químico altamente tóxico utilizado en la industria petroquímica. La fuente de la contaminación fue un escape en un oleoducto operado por la Corporación Nacional de Petróleo de China, lo que llevó a una escasez de agua embotellada en la ciudad y generó preocupaciones sobre la salud pública.

Brote de E. coli en Canadá

En septiembre de 2023, un brote de E. coli en Calgary, Canadá, afectó a 264 niños, muchos de los cuales fueron hospitalizados con síntomas graves. La bacteria se propagó desde una cocina que abastecía a 11 guarderías, donde se encontraron condiciones insalubres, como alimentos sin refrigeración y presencia de plagas. Las autoridades sanitarias confirmaron 310 casos y diagnosticaron a 20 niños con síndrome urémico hemolítico (SUH), una complicación grave que afecta los riñones.

Caso de Aguas de La Habana, Cuba

En septiembre de 2024, más de un millón de cubanos enfrentaron dificultades para acceder al agua potable debido a problemas en la infraestructura y pérdidas de agua en la red de distribución. A pesar de inversiones millonarias, como los 102 millones de dólares aportados por Kuwait para mejorar el sistema hidráulico, la crisis persistió debido a la falta de mantenimiento y materiales constructivos. En abril de 2025, la empresa Aguas de La Habana reportó que entre 30,000 y 40,000 personas fueron afectadas diariamente por averías en las redes y cortes eléctricos, lo que incrementó el uso de camiones cisterna para abastecer a la población.

Además, la infraestructura deficiente ha provocado grandes pérdidas de agua antes de que llegue a los hogares, lo que agrava la escasez hídrica en la ciudad. La

situación ha generado costos adicionales en combustible y productos químicos para el tratamiento del agua, además de afectar la calidad del servicio.

Caso de la minería en Chile

Chile enfrenta una crisis hídrica que ha impactado fuertemente a la industria minera. En 2024, la minería representó aproximadamente el 12% del PIB del país, pero la escasez de agua ha generado costos adicionales y conflictos con comunidades locales. La minería chilena depende de diversas fuentes de agua: 28% de fuentes superficiales, 33% de fuentes subterráneas, 30% del mar y 9% de terceros. Para enfrentar la crisis, la industria ha incrementado el uso de desalación de agua de mar, que pasó de representar el 30% del consumo de agua en minería en 2024 a una proyección del 50% para 2030.

Un ejemplo clave es el proyecto conjunto entre ABB y Minera Los Pelambres, que busca que el 90% del agua utilizada en sus procesos provenga del mar o de fuentes recirculadas. Este proyecto ha inyectado a la economía de la región de Coquimbo 1,977 millones de dólares y en su punto de máxima actividad generará más de 2,000 empleos. Sin embargo, la crisis hídrica sigue afectando la productividad y aumentando los costos operativos en el sector.

Estos casos muestran cómo la falta de detección temprana de pérdidas de agua puede generar impactos económicos y sociales significativos

En Argentina, varias industrias han enfrentado problemas por pérdidas de agua no detectadas a tiempo, lo que ha generado impactos económicos y ambientales significativos. Aquí algunos casos:

- Sector minero en la provincia de San Juan: En 2021, una minera sufrió una fuga de agua en su sistema de procesamiento que no fue detectada durante semanas. Esto provocó una pérdida de miles de metros cúbicos de agua y afectó la producción, generando costos adicionales en insumos y energía.
- Industria petrolera en Vaca Muerta: En 2020, se reportaron pérdidas de agua en los procesos de fractura hidráulica. La falta de monitoreo adecuado llevó a un desperdicio de millones de litros de agua, lo que incrementó los costos operativos y generó conflictos con comunidades locales.
- Plantas industriales en Buenos Aires: En 2024, varias fábricas enfrentaron problemas por fugas en sus sistemas de refrigeración y producción. La falta de mantenimiento adecuado provocó pérdidas de agua que afectaron la eficiencia operativa y aumentaron los costos de producción.

Daños en las estructuras edilicias

Además de las consecuencias anteriormente expuestas existen daños comunes a todas las fugas hídricas a nivel arquitectónico nos referimos a la estructura edilicia. Estos ejemplos muestran la importancia de la detección temprana de fugas para evitar daños estructurales graves

Edificios con filtraciones graves: Según un análisis de patologías en construcciones, las infiltraciones de agua pueden debilitar la estructura de los edificios, causando daños en cimientos y paredes.

En viviendas de segunda mano, las fugas de agua no detectadas con el tiempo han generado debilitamiento en la estructura, provocando grietas, filtraciones y crecimiento de moho. Muchas de estas pérdidas de agua fueron invisibles durante años, causando daños que requieren costosas reparaciones.

Colapsos estructurales: En algunos casos, la acumulación de humedad y el deterioro de materiales han llevado al colapso parcial o total de edificaciones, lo que ha requerido reconstrucciones complejas.

Fugas invisibles en tuberías: Las fugas de agua ocultas pueden provocar daños progresivos en la infraestructura, afectando la estabilidad de las construcciones y generando costos elevados en reparaciones.

Experiencia Laboral: Gestión de Incidentes en el Consumo de Recursos Hídricos

En noviembre de 2024, el área de Higiene y Seguridad (HSE, por sus siglas en inglés) detectó un consumo anómalo de agua, equivalente a siete veces el volumen habitual. A través de un análisis retrospectivo de las facturas provistas por AYSA, se identificó que el incremento tuvo su origen en mayo del mismo año, como consecuencia de una pérdida no visible al momento de su ocurrencia (véase Fig. 1 y Fig. 2). La falta de detección temprana impidió una intervención oportuna, agravando el impacto operativo y económico en el sitio afectado.

Dado el carácter urgente del problema, no fue factible realizar una cotización previa ni aplicar un proceso de licitación entre proveedores. Se recurrió de manera inmediata al proveedor con mayor disponibilidad operativa, sin realizar comparaciones de precios. Este procedimiento de emergencia implicó costos significativamente superiores a los habituales, similares a situaciones cotidianas como el requerimiento de un cerrajero fuera de horario, pero con un valor aproximado de, al menos, cuatro jornales diarios sólo en concepto de mano de obra.

La pérdida de agua, además, provocó el debilitamiento de la estructura del piso en la zona afectada, lo cual fue reportado nuevamente por el área de HSE como un riesgo para la seguridad operativa. Esta situación condujo a una segunda intervención correctiva de similares características, generando un gasto adicional considerable.

Finalmente, el consumo excesivo de agua no solo implicó un impacto económico desproporcionado, sino que evidenció una debilidad crítica en términos de

sostenibilidad y gestión de recursos. La ausencia de un sistema de monitoreo en tiempo real sobre un insumo vital como el agua derivó en una situación evitable de gran perjuicio. A pesar de que el daño ya se había producido, se estableció la necesidad imperiosa de implementar medidas correctivas que impidan la repetición de este tipo de eventos en el futuro.



fig. 1.2 Pérdida debajo del piso



fig. 1.3 búsqueda de pérdida de agua

OBJETIVO

El objetivo principal de esta investigación es desarrollar un sistema de detección de fugas y análisis de consumo en tiempo real, utilizando Internet de las Cosas (IoT) como herramienta clave. La implementación de controladores electrónicos permitirá registrar el flujo de líquidos y generar alertas en caso de uso anómalo, garantizando una gestión eficiente de los recursos hídricos.

Se pretende la detección de fugas y monitoreo de consumo en sistemas de líquidos mediante IoT.

En la actualidad, la gestión eficiente del consumo de líquidos es fundamental para optimizar recursos y minimizar pérdidas. Este trabajo propone una metodología basada en (IoT) para detectar fugas en sistemas de líquidos y monitorear el consumo en tiempo real. A través de sensores y actuadores electrónicos, se busca generar una lectura precisa del flujo de agua y diferenciar entre un consumo intencionado, como el uso de grifos o sistemas de riego, y una pérdida imprevista.

Resultados esperados

La aplicación de este sistema proporcionará una herramienta eficiente para la gestión del agua, reduciendo pérdidas y optimizando su uso. Se espera que la implementación de IoT en la supervisión de sistemas de líquidos facilite el desarrollo de estrategias de ahorro y mantenimiento preventivo en diversas industrias y entornos domésticos.

DISEÑO DEL PROTOTIPO

El prototipo se puede describir como el conjunto de 5 bloques electrónicos montados en un PCB, dos de estos son alimentación, un circuito de potencia con relés y mini bombas, uno lógico y el quinto de conexión IoT. El primero de estos es una fuente regulada de 5vdc para alimentar el 3er bloque potencia y todo lo que pueda generar ruido, el segundo bloque es una fuente de 5vdc que está dedicada al microcontrolador para evitar perturbaciones en la línea. El cuarto bloque es la parte lógica un esp-32 al que se conectan varios sensores que a través de las señales que emiten, mediante técnicas de procesamientos de datos se traducen en información del monitoreo. Por último se aloja toda esta información en un servidor virtual y se muestra en una página para conseguir desde donde se podrá ver en tiempo real la información mediante IoT.

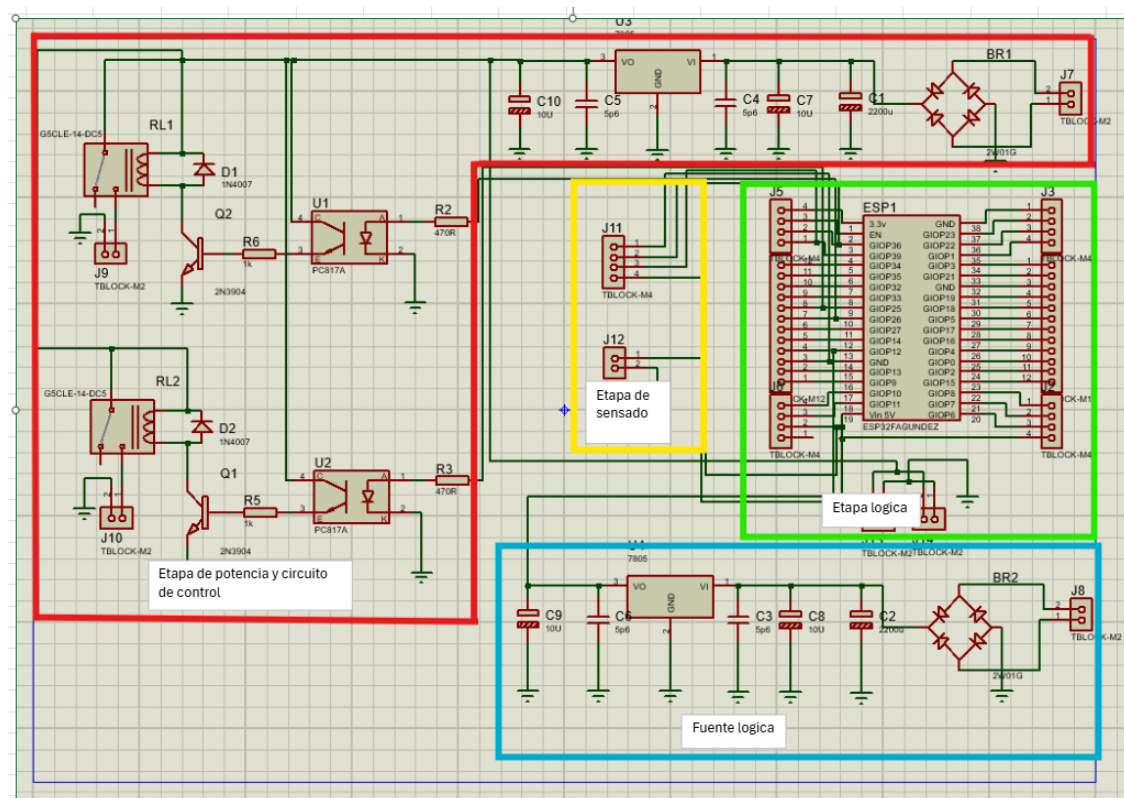


fig. 1.4 Etapas del prototipo

DESARROLLO Y CONSTRUCCIÓN

Diseño de las fuentes reguladas 5vdc

Para el desarrollo de la placa se elaboraron dos fuentes de iguales características, una para manejar la etapa de potencia y la otra para manejar la etapa lógica.

Transformador 220v-12v

El transformador 220Vac a 12 Vac, 3A es común para ambas fuentes. Se eligió esta potencia para alimentar dos circuitos rectificadores de 5vdc.

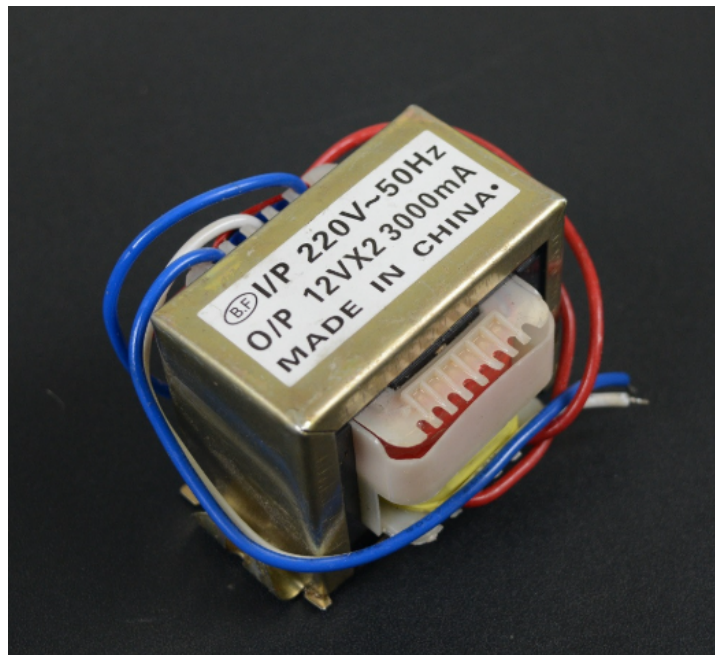


fig.1.5 Transformador reductor 220vac-12vac

Fuente de potencia y control

Puente de diodos

Una vez reducida la señal de la red eléctrica a 12Vac se conecta esta tensión al circuito rectificador que comienza con un puente de diodo.

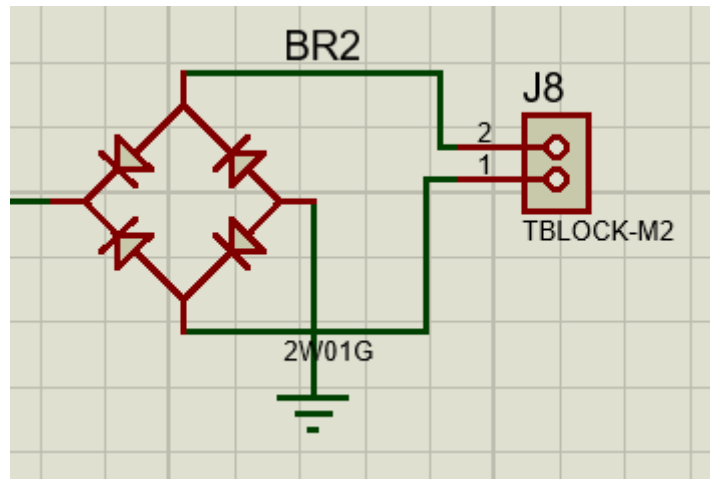


fig. 1.6 Puente de diodo diagrama electrónico

En los sistemas electrónicos de alimentación, la conversión de corriente alterna (CA) en corriente continua (CC) es una etapa fundamental. El puente de diodos, también conocido como puente rectificador o puente de Graetz, es un circuito ampliamente utilizado para lograr esta conversión de manera eficiente y confiable. Este informe detalla su estructura, principio de funcionamiento y aplicaciones prácticas.

Un puente de diodos es un circuito rectificador de onda completa compuesto por cuatro diodos dispuestos en una configuración específica que permite convertir una señal de CA en una señal de CC pulsante. Su principal ventaja es que aprovecha ambos semiciclos de la señal alterna, a diferencia del rectificador de media onda.

Se eligió en particular el puente W10m porque soporta 1.5 amperes que son suficientes para la demanda del circuito además de ser de bajo costo y tamaño reducido.

Cálculo de la tensión de salida rectificada

v_p = tensión pico

tensión pico = tensión de salida de transformador por raíz de dos

v_d = tensión de caída en los diodos

$$v_p - v_d \Rightarrow 12v \times \sqrt{2} - 1,4v = 15,57v$$

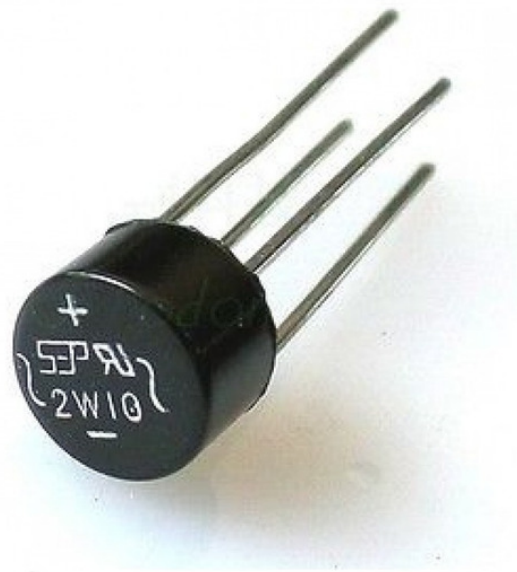


fig. 1.7 Puente de diodo

Capacitores de filtrado

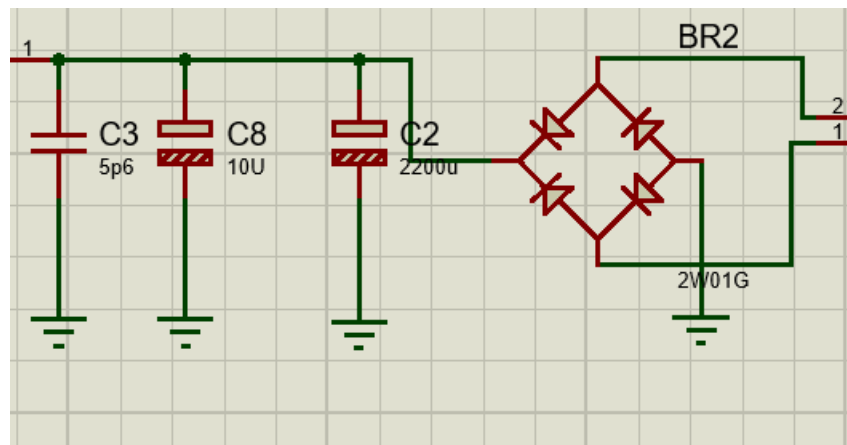


fig. 1.8 Filtrado de señal entrada al regulador

Con la tensión rectificada se pasa a la etapa de filtrado que se necesita entre el puente y el regulador. Siempre es aconsejable estabilizar esta tensión y rectificar lo mejor posible. Según lo calculado se escogió para un ripple de tensión aceptable dos capacitores electrolíticos uno de 2200 μ F 25V y 1000 μ F 25v, para el desacoplo

un capacitor cerámico de 0.1 μF . Los capacitores se conectan en paralelo y todo ruido que se genere los desacople los eliminaran enviandolos por gnd.

Rizado

capacitor= corriente de carga / (frecuencia x 2(onda completa))

$C = 0.7 / (100 \times 2) = 0.7 / 200 = 0.0035 \text{ F} = 3500 \mu\text{F}$ se elige $2200\mu\text{F} + 1000\mu\text{F}$

Desacople de ruido

0.1 μF



fig. 1.9 Capacitor electrolítico

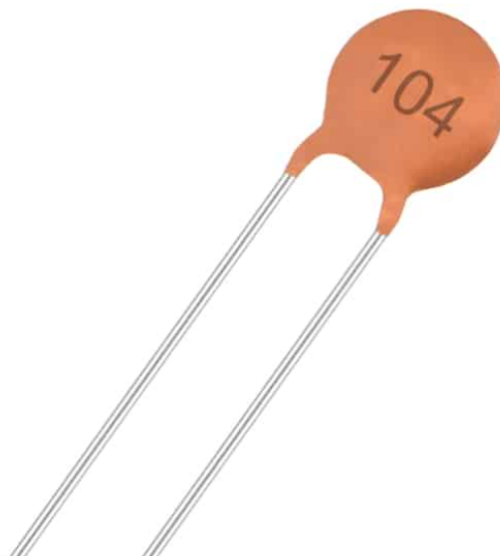


fig. 2.0 Capacitor cerámico

Regulador LM7805

Para garantizar los 5v y el funcionamiento estable y seguro de los componentes se optó por el LM7805, un circuito integrado regulador de voltaje lineal que proporciona una salida constante de +5 voltios a partir de una entrada variable, típicamente entre 7v y 35v como se mencionó anteriormente la salida del transformador alimenta 2 fuentes reguladas , por lo tanto la tensión de salida del trafo de 15.7v se reparte por igual 7,85v en cada una de dichas fuentes, valores que se encuentran dentro de los parámetros de trabajo del regulador .

El LM7805 es ideal debido a su simplicidad, bajo costo y confiabilidad. La corriente que entrega es suficiente para mantener operando el circuito sin recalentamiento y sin necesidad de una refrigeración forzada.

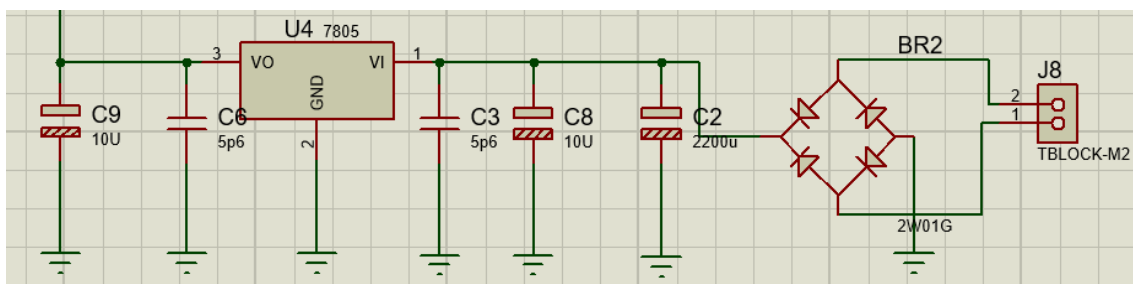


fig. 2.1 Regulador con filtros en la entrada y salida

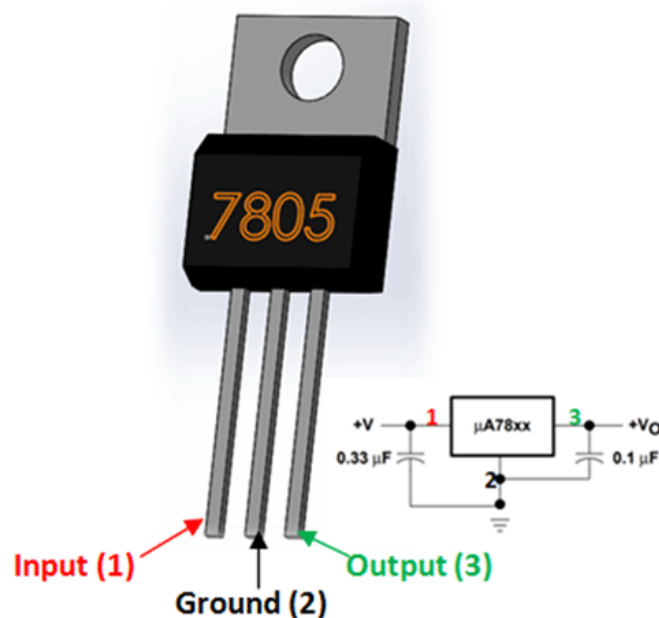


fig. 2.2 Regulador lm7805

- Pin de entrada (IN): recibe el voltaje no regulado.

- Pin de tierra (GND): referencia común.
- Pin de salida (OUT): entrega los 5V regulados.

Etapa de potencia y circuito de control

Optoacoplador PC817

A la salida de esta línea de 5vdc regulada se usó para el circuito de control un optoacoplador PC817. En sistemas electrónicos donde se requiere aislamiento galvánico entre distintas secciones de un circuito, es uno de los optoacopladores más utilizados por su bajo costo, fiabilidad y facilidad de integración.

Está compuesto internamente por un LED infrarrojo y un fototransistor NPN, encapsulados en un paquete DIP de 4 pines.

De esta manera se consigue aislar eléctricamente las dos partes de un sistema, permitiendo la comunicación entre ellas mediante luz infrarroja.

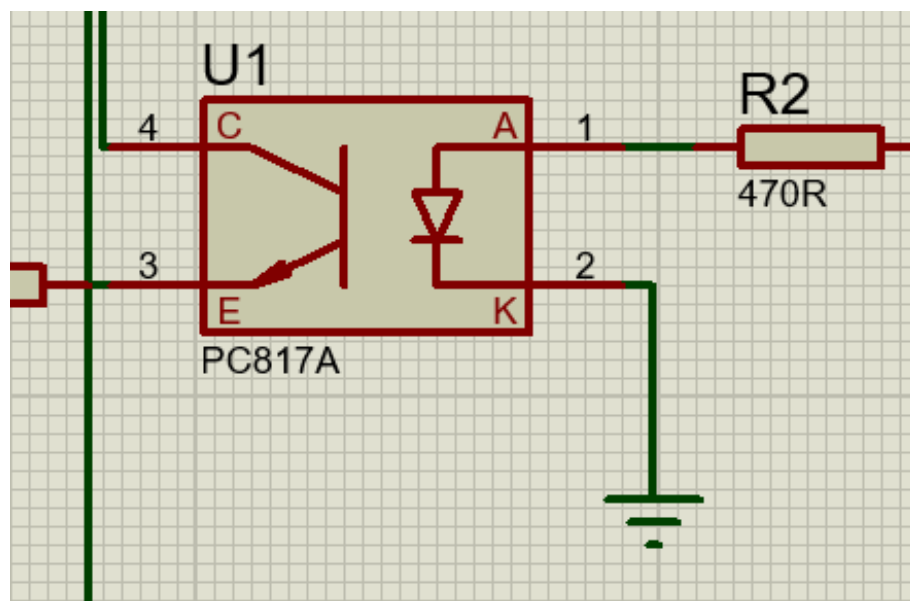


fig. 2.3 Resistencia del optoacoplador y resistor de 470Ω

La resistencia de polarizado del led se obtiene de la diferencia de la salida del esp32 (3.3V) y la caída de tensión en el diodo (1.3V) sobre una corriente segura en este caso son 4mA .

Cálculo de resistencia del led del optoacoplador

$(\text{Tensión de salida del gpio} - \text{tensión del led}) / \text{corriente segura que se decretó}$

$(3.3\text{v} - 1,3\text{ v}) / 4\text{ mA}$, se define una corriente para el led 4 mA , entonces :

$2\text{V} / 4\text{mA} = 500\Omega$, valor comercial más cercano 470Ω

PINOUT Optoacoplador PC817

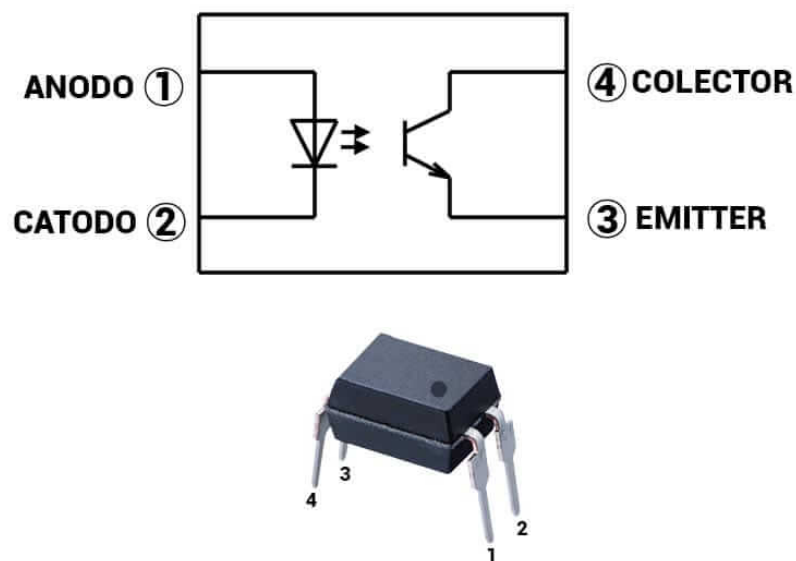


fig. 2.4 Pinout de un optoacoplador pc817

Transistor 2N3904

En la salida del optoacoplador, para manejar mayor corriente y tensión se utilizó un transistor del tipo npn 2n3904 en configuración emisor común ya que es ideal para conmutaciones o amplificación de señales muy débiles.

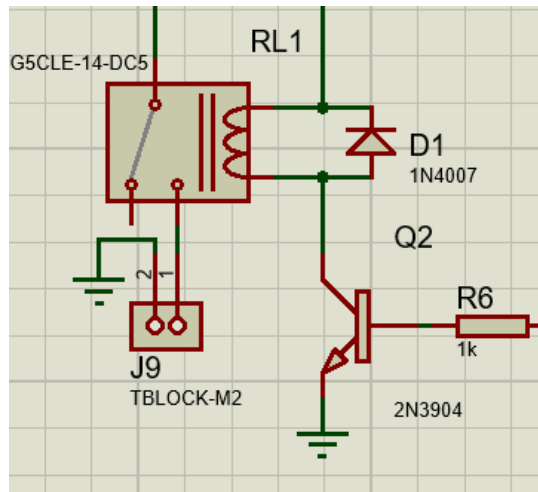


fig. 2.5 Transistor 2N3904 conectado a un relé esquemático

Especificaciones técnicas principales

- Tipo: NPN
- Encapsulado: TO-92
- Tensión máxima colector-emisor (V_{ce}): 40 V
- Corriente máxima de colector (I_c): 200 mA
- Potencia máxima de disipación: 625 mW
- Ganancia de corriente (h_{FE}): entre 30 y 300
- Frecuencia de transición (f_t): hasta 300 MHz
- Tensión de saturación colector-emisor: < 0.3 V a 10 mA
- Rango de temperatura de operación: $-55\text{ }^{\circ}\text{C}$ a $+150\text{ }^{\circ}\text{C}$

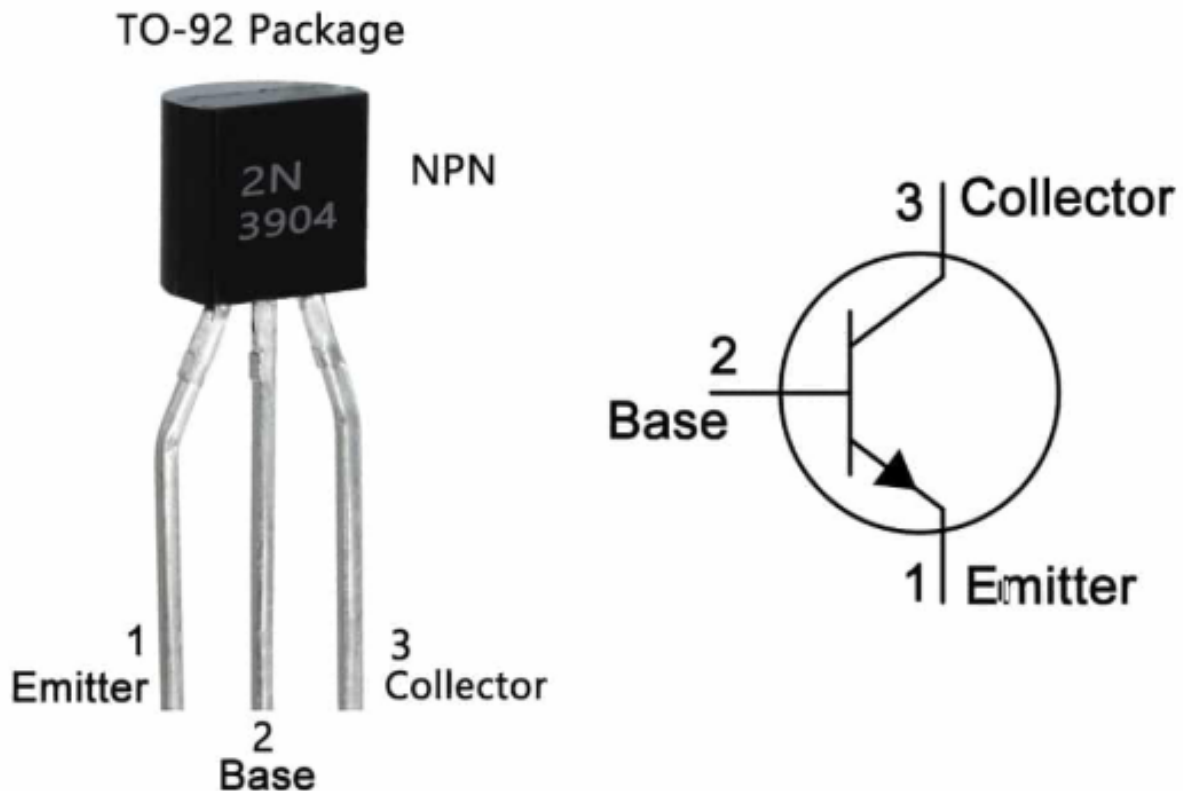


fig. 2.6 Pinout de un transistor 2N3904

El transistor no puede conectarse directamente al gpio, necesita una resistencia en la base para accionarlo para reducir la corriente de esta salida para trabajar de manera segura en saturación. Esta corriente se obtiene del datasheet del npn 3904 y se calcula de la siguiente manera cuando se accione la salida de 3.3v del saturación

Cálculo de resistencia de base

El transistor no puede conectarse directamente al optoacoplador necesita una resistencia de base que se calcula tomando la corriente de activación del relé que indica el datasheet (*fig.2.2*) que son para 5v, 71.4 mA. Una corriente de 100mA en el colector es suficiente para accionar el relé

Cálculo de resistencia de base.

$v_{cc} = 5v$

$v_{be} = 0.7v$ (caída en los diodos)

$v_{cc} - v_{be}$

i_b = corriente de base

i_c = corriente de colector se eligió 100mA

r_b = resistencia de base

β = ganancia del transistor en veces

$$i_c = i_b \times \beta \Rightarrow 100 = i_b \times 30$$

$$i_c / \beta = i_b \Rightarrow 100 \text{ mA} / 30 = 3.33 \text{ mA}$$

$$v_b / i_b = r_b \Rightarrow 4.3 \text{ V} / 3.33 \text{ mA} = 1.291,29 \Omega \text{ valor comercial menor más cercano para garantizar la saturación } 1 \text{ k}\Omega .$$

Relé SRD-05VDC-SL-C

El relé escogido para la carga de la bomba es un relé SRD-05VDC-SL-C. Se optó por este porque es uno de los más populares que se usan en 5v, la corriente que necesita en la bobina para el accionamiento son 70 mA, son fáciles de conseguir tiene muy bajo consumo 360mW y esto es muy importante no solo es menor el consumo también es menor la energía que entrega esos picos son bastantes nocivos para el circuito generan daños severos. Para esto se colocó un diodo de alta frecuencia 1N4007 que funciona como apagachispa es decir protege al circuito de la corriente que devuelve la bobina.

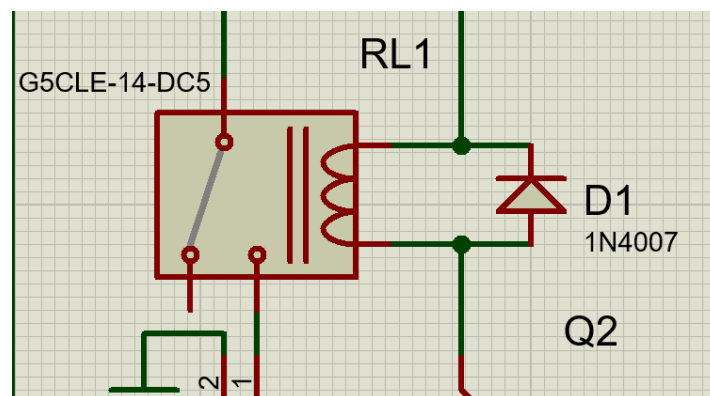


fig. 2.7 Diodo apagachispa conectado al relé

Es importante que de reemplazar el relé por uno de mayor potencia o en caso de no conseguir el mismo el reemplazo debe cumplir con las medidas (2.1) de lo contrario el montaje en el pbc no será posible ya que los pines de conexión no permiten ajustarse como son los de una resistencia o un capacitor

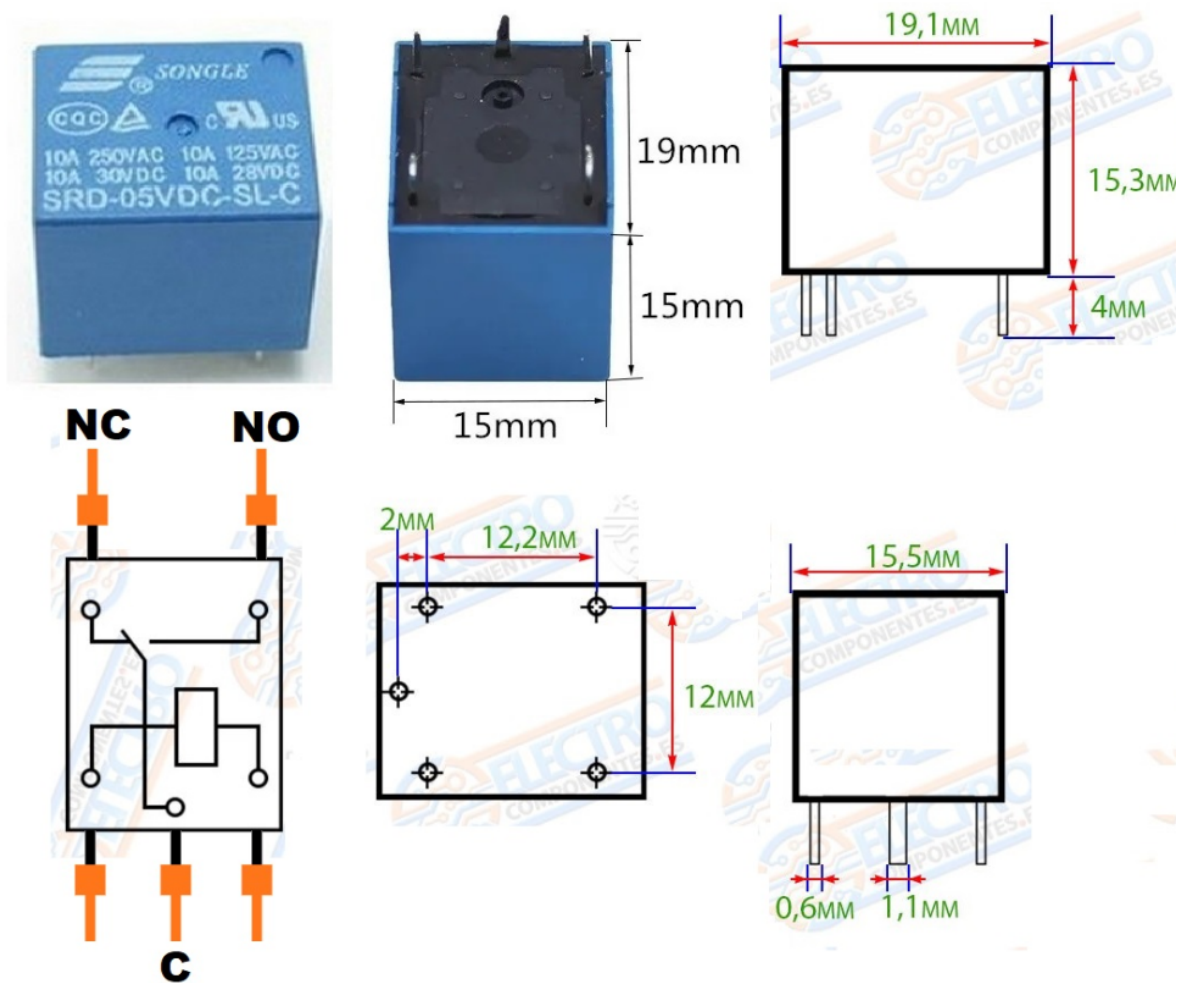


fig. 2.8 Relé SRD-05VDC-SL-C medidas y cortes

SRD Series SUBMINATURE HIGH POWER RELAY 7A/10A/15A

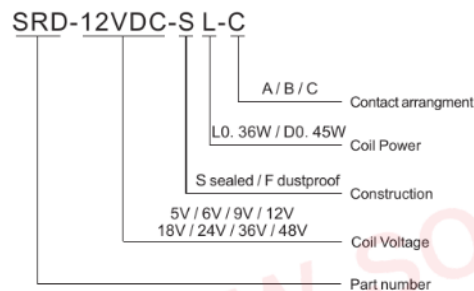




Features

- Miniature volume, PCB usage
- Contact rating: 7A/10A/15A
- Temperature range: +85°C/+105°C

Ordering Information



Contact Rating

Contact arrangement	1A(spstno)/1B(Spstnc)/1C(Spdt)
Contact resistance	100mΩ (1A 6VDC)
Contact material	Silver alloy: AgCdO, AgSnO ₂ , AgNi
Contact rating	7A/250VAC 10A/250VAC 15A/250VAC
Max switching voltage	250VAC
Max switching current	15A
Max switching power	3750VA
Electrical endurance (frequency: 1800 ops/h)	1x10 ⁵ ops / 1x10 ⁴ ops
Mechanical endurance (frequency: 18000 ops/h)	1x10 ⁷ ops

Characteristics

Insulation system		Class B / Class F
Insulation resistance		100MΩ (500VDC)
Dielectric Strength	Between contacts and coil	1500VAC 1 minute
Leakage current 1mA	Between open contacts	1000VAC 1 minute
Operate time(Under nominal voltage)		≤10ms
Release time(Under nominal voltage)		≤10ms
Humidity		85% RH (20℃)
Ambient Temperature		-40℃~+85℃/-40℃~+105℃
Shock resistance	Functional	98m/s ²
	Destructive	980m/s ²
Vibration resistance (Double amplitude)		10Hz~55Hz 1.5mm
Weight		Approx 9g
Construction		Sealed

Coil Data

(at 20°C)

Power dissipation W	Voltage VDC	Current mA	Resistance Ω ± 10%	Pick-up voltage	Drop-out voltage	Max allowable
0.36W (L)	05	71.4	70	75%Max	10%Min	130%
	06	60	100			
	09	40	225			
	12	30	400			
	18	20	900			
	24	15	1600			
	36	10	3600			
	48	7.5	6400			
0.45W (D)	05	89.3	55	75%Max	10%Min	130%
	06	75	80			
	09	50	180			
	12	37.5	320			
	18	25	720			
	24	18.7	1280			
	36	12.5	2880			
	48	10	4500			

fig. 2.9 Datasheet rele SRD-05VDC-SL-C

Diodo 1N4007

El 1N4007 es un diodo rectificador de silicio de uso general, ampliamente utilizado en electrónica para convertir corriente alterna (AC) en corriente continua (DC). Pertenece a la familia de diodos 1N400x, siendo el modelo con mayor capacidad de tensión inversa. Se optó por este diodo porque es el que más se usa para proteger transistores o relés de picos de voltaje inducidos por cargas inductivas, es económico y se consigue fácil si necesita reemplazarlo.

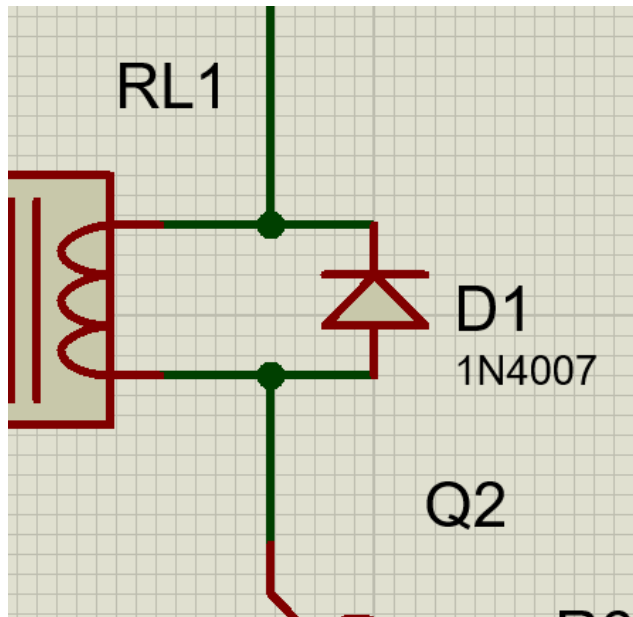


fig. 3.0 Diodo en paralelo a la bobina para protección

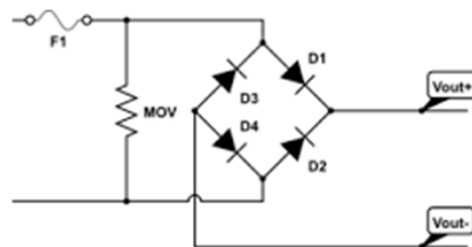
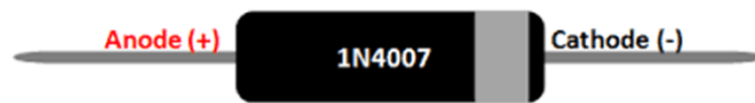


fig. 3.1 Diodo 1n4007, simbología y configuración en puente

Fuente de lógica

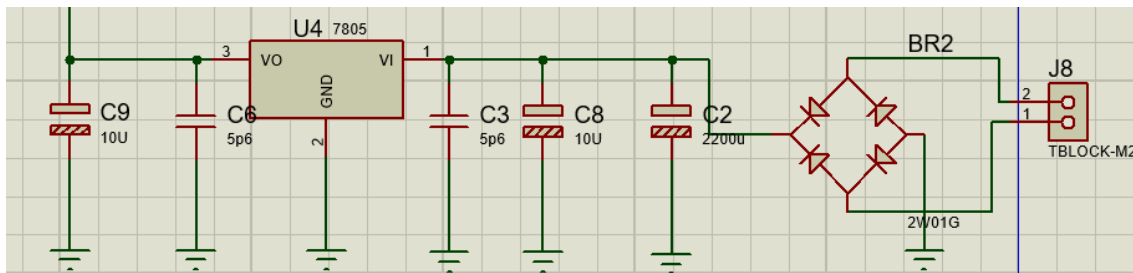


fig. 3.2 Fuente lógica

Como se mencionó anteriormente la fuente de etapa lógica tiene las mismas características de la etapa de potencia e inclusive comparten el mismo transformador. La diferencia es que este diseño solo se ocupa de alimentar la etapa lógica.

NodeMCU ESP32S

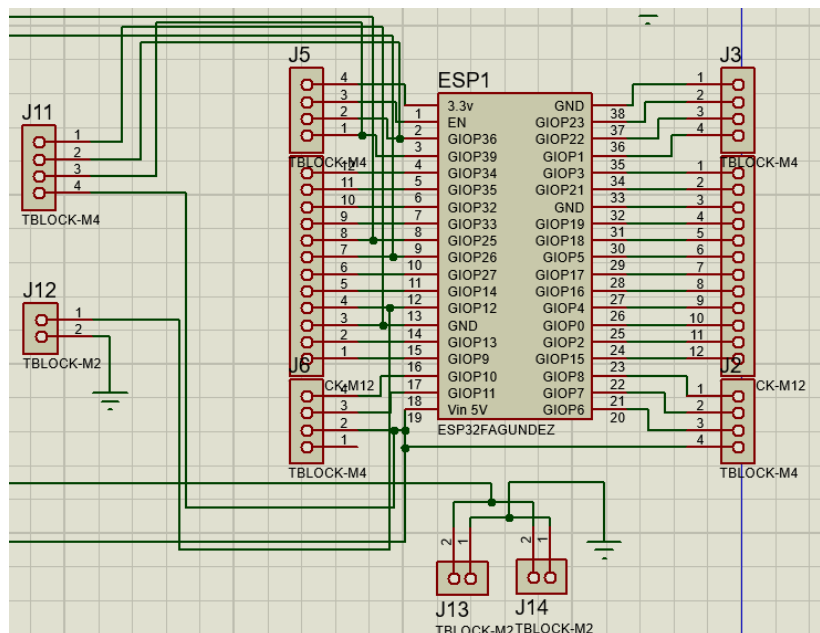


fig. 3.3 Módulo esp 32 esquemático

Para el control de la parte lógica se utilizó NodeMCU ESP32S que es una placa de desarrollo basada en el microcontrolador ESP 32 de bajo consumo, con abundantes pines programables, fabricado por Expressif Systems. Se trata de un sistema en chip (SoC) altamente integrado que combina conectividad WiFi, Bluetooth y una arquitectura de doble núcleo, lo que lo convierte en una plataforma ideal para aplicaciones de Internet de las Cosas (IoT), automatización y sistemas embebidos. Se desarrolló el código en lenguaje C++ (Arduino IDE).

Características técnicas destacadas

- Procesador: Xtensa LX6 dual-core a 240 MHz
- Memoria RAM: 520 KB
- Memoria Flash: 4 MB
- Voltaje de operación: 3.3V
- Conectividad: WiFi 802.11 b/g/n y Bluetooth v4.2
- Entradas/Salidas: Hasta 34 pines GPIO, ADC, DAC, PWM, SPI, I2C, UART

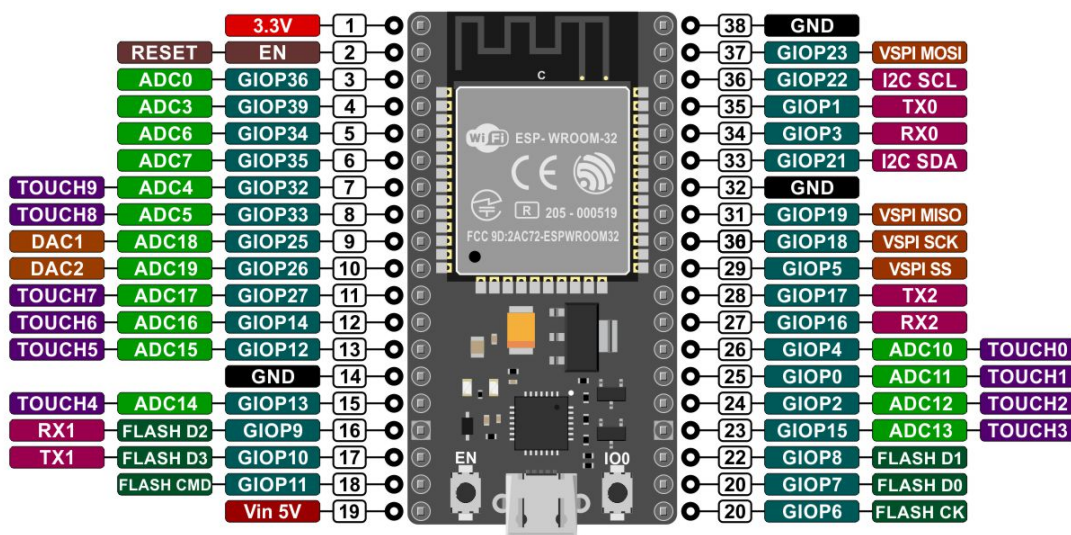


fig. 3.4 NodeMCU32S

Sensor ultrasónico hc-sr04

Para la medición de distancias se escogió un sensor de ultrasonido. Este dispositivo utiliza ondas en altas frecuencias para que el oído humano no sea capaz de percibirlo y así no moleste o interfiera con el día a día. Esto también permite que no exista contacto directo con el líquido.

- Transmisor: Dentro del emisor encontramos un cristal piezoeléctrico que cuando se le introduce una señal eléctrica este empieza a vibrar y por consecuencia de la vibración se crea una onda de sonido que viaja hacia la dirección en que esté colocado el sensor.
- Receptor: El receptor básicamente hace el proceso inverso. Un cristal piezoeléctrico recibe el sonido que rebotó en algún objeto o superficie y lo convierte en una señal eléctrica.
- Alimentación: Sirve para hacer funcionar el sensor.

- Circuito eléctrico: Este circuito se encarga de procesar la señal y traducir el tiempo a distancia.

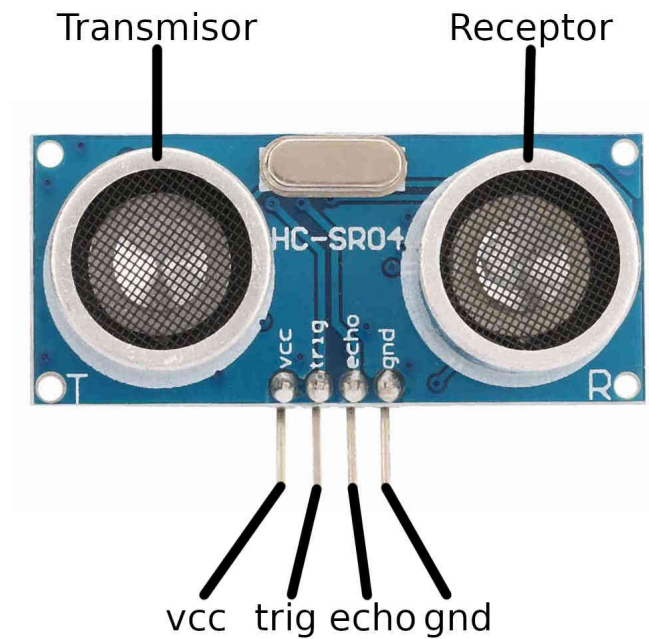


fig. 3.5 Terminales de un sensor de ultrasonido hc-sr04

Funcionamiento

El funcionamiento de los ultrasónicos se basa en mandar una señal de sonido hacia algún lugar en concreto y esperar a que esta señal rebote en algún objeto y regrese al mismo sensor, es decir que estos dispositivos tienen un emisor y un receptor que trabajan en conjunto. Para saber a qué distancia se encuentra un objeto él sensor cuenta el tiempo desde que se mandó la señal hasta lo que tarda en regresar y a través de cálculos matemáticos traduce este tiempo en distancia.



pulso de 10us

us/58 es igual a distancia (cm)

fig. 3.6 Funcionamiento sensor de ultrasonido hc-sr04

Sensor de flujo YF-S401

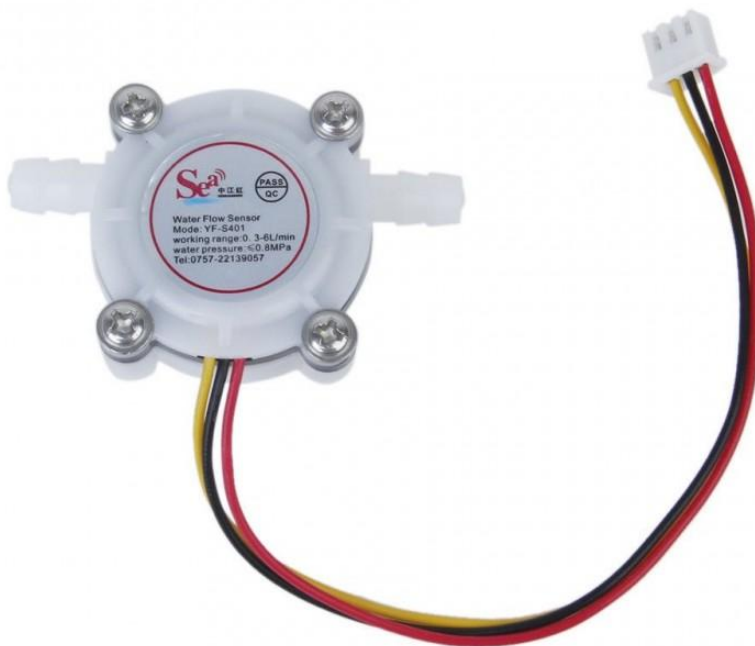


fig. 3.7 Sensor de flujo yf-s401

Se eligieron estos sensores debido a su tamaño bajo consumo y bajo costo es un sensor de flujo de construcción sólido el cual está constituido por un cuerpo de plástico, un rotor de agua y un sensor de efecto Hall. El diseño y el funcionamiento de este tipo de sensor es simple. Utiliza un sensor con aspas o álabes para medir la cantidad de líquido que se ha movido a través de él. Las aspas tienen un pequeño imán y un sensor magnético de efecto Hall que registra cada vuelta de las aspas, esto genera impulsos de salida a una velocidad proporcional a la velocidad de flujo. La flecha indica la dirección del flujo. Al contar los pulsos de la salida del sensor, puede seguir fácilmente el movimiento del fluido: cada pulso es de aproximadamente 2,25 mililitros

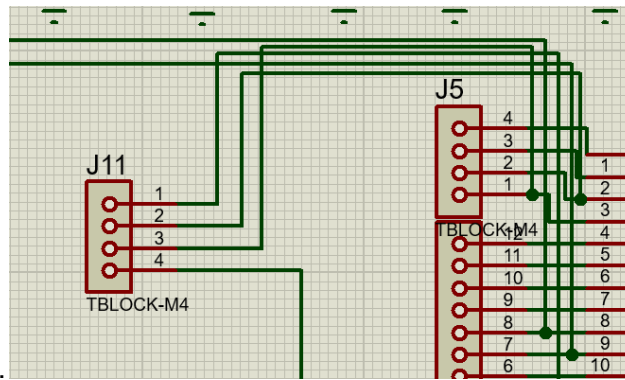


fig. 3.8 Tblock donde conectará el sensor de flujo

El sensor viene con tres cables: rojo (energía 5-18VDC), negro (tierra) y amarillo (salida de pulsos de efecto Hall). Al contar los pulsos de la salida del sensor, se puede calcular fácilmente el flujo de agua. Cada pulso es de aproximadamente 2.25 mililitros. La señal de pulso es una simple onda cuadrada por lo cual es bastante fácil de registrar y convertir en litros por minuto utilizando la siguiente fórmula:

Frecuencia de pulsos (Hz) / 7 = caudal en L / min.

Este no es un sensor de precisión, y la frecuencia del pulso varía un poco dependiendo de la velocidad de flujo, la presión del fluido y la orientación del sensor. Se necesitará una cuidadosa calibración si se requiere más que un 10% de precisión.

Display tft 3.5

Para mostrar los valores y armar un menú de configuración táctiles se escogió una pantalla tft de 3.5 pulgadas con controlador gráfico ili 9488 y un controlador touch resistivo XPT2046. Por tener un tamaño grande resulta espacioso para el armado de los botones del menú y a pesar de su bajo costo posee características interesantes.

Controlador gráfico ILI 9488

- SPI de 3/4 cables, configurable por software
- 480x320 píxeles
- 3 modos compatibles: 16 bits (65.000 colores) / 18 bits (262.000 colores) / 24 bits (16,7 millones de colores)

Controlador touch XPT2046

- Panel táctil resistivo de 12 bits y 125 kHz
- sensible a la presión
- sensor de temperatura
- SPI de 4 cables
- Admite interrupción táctil

- Regulador de 5v a 3,3v, se debe cortocircuitar J1 para usar 3,3v

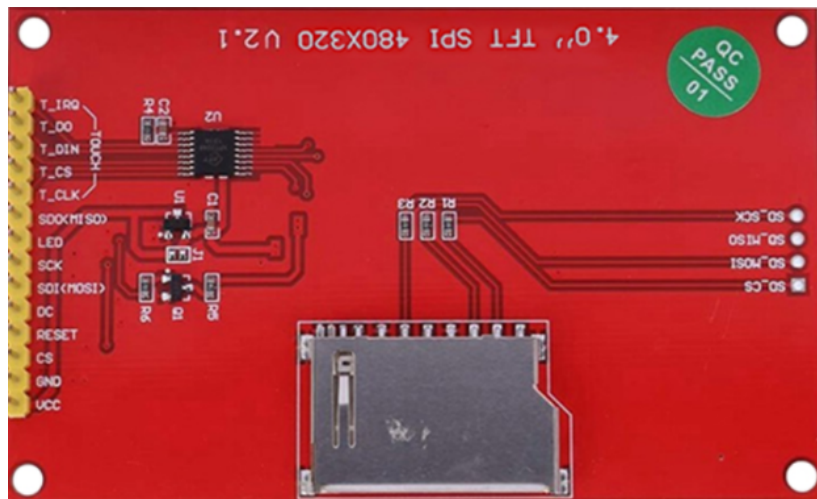


fig. 3.9 Pines de un módulo tft ili 9488

Diagrama de conexión con esp32



fig. 4.0 Pines de conexión ili 9488

Para el conexionado de la pantalla se requieren 9 pines gpio, además de la alimentación 3v-gnd.

TFT LCD Touchscreen	ESP32
T_IRQ	
T_DO	GPIO 19
T_DIN	GPIO 23
T_CS	GPIO 13
T_CLK	GPIO 18
SDO(MISO)	GPIO 19
LED	5v (or 3.3V)*
SCK	GPIO 18
SDI(MOSI)	GPIO 23
D/C	GPIO 2
RESET	GPIO 4
CS	GPIO 15
GND	GND
VCC	5V (or 3.3V)*

fig. 4.1 Tabla de conexionado pines

**La alimentación puede 3.3v o 5v en este caso se tomó la primera que es la salida que nos ofrece el esp32*

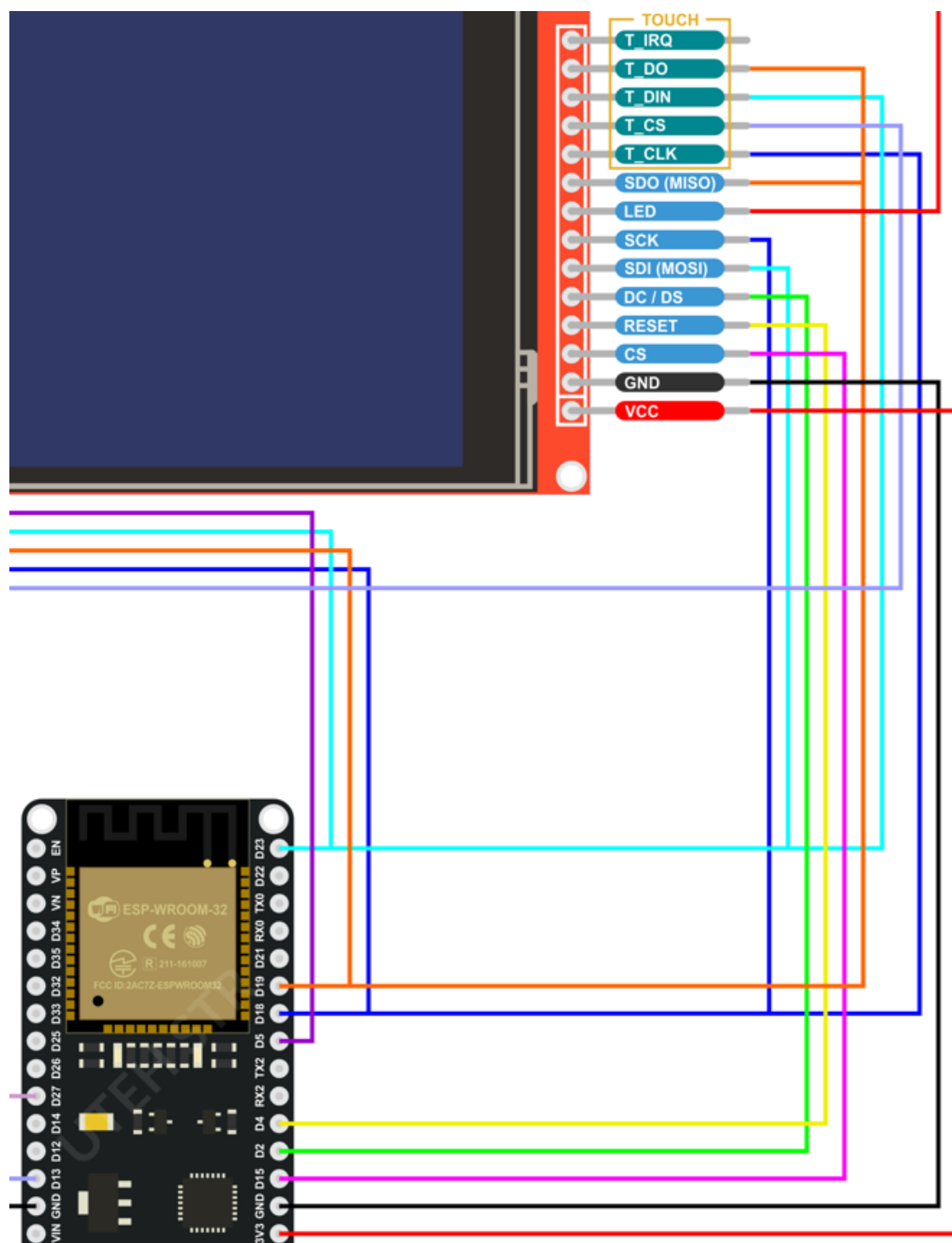


fig.4.2 Diagrama de conexión de la pantalla tft

Conclusión

Este proyecto no solo constituye una integración eficaz entre hardware y software, sino también la concreción de una idea concebida para resolver una problemática real. A lo largo de su desarrollo, fue necesario replantear y rediseñar diversas etapas, producto de los desafíos técnicos y conceptuales que surgieron en el camino. Desde la formulación inicial, que no cumplía con los objetivos esperados, hasta alcanzar un funcionamiento óptimo, el proyecto evolucionó de manera dinámica. Aun así, permanece abierto a futuras mejoras, lo cual resalta el valor del aprendizaje continuo y el poder transformador de la innovación en la ciencia electrónica.

Consideraciones Finales

A pesar de haberse completado el desarrollo del proyecto, se considera que aún no se encuentra en condiciones óptimas para su comercialización. Se identificó la necesidad de realizar ensayos adicionales enfocados en la calibración y precisión de los sensores, especialmente en lo referido a la medición de caudales. La mejora de estos aspectos técnicos permitirá, en el futuro, posicionar la solución como una alternativa válida en el mercado.

Durante el proceso de fabricación, se observó un cambio significativo en la percepción de los aspectos vinculados a la programación. Inicialmente percibida como una disciplina compleja y poco accesible, con una fuerte dependencia de la memorización de estructuras, la práctica continua permitió transformar dicha visión. Con el transcurso del proyecto, la programación se convirtió en una actividad estimulante, desafiante y gratificante. Si bien no se proyecta un ejercicio profesional exclusivo en esta área, se reconoce su potencial como actividad recreativa de alto valor personal. Este cambio de perspectiva constituye uno de los logros más relevantes del proceso formativo.

Por otro lado, se destaca el papel fundamental de las herramientas de inteligencia artificial como soporte en la redacción de textos técnicos y la generación de recursos visuales. Su intervención fue clave para mejorar la sintaxis, optimizar la presentación de la información y agilizar tareas de documentación técnica, aportando un valor añadido significativo al desarrollo del proyecto.

Bibliografía

- agua.org.mx “El acceso al agua potable”

<https://agua.org.mx/mundo-el-acceso-al-agua-potable-en-el-mundo-statista/>

- Clarín “Día mundial del agua”

[Día Mundial del Agua: cuánto consume una canilla que gotea, una ducha por minuto y cada descarga de inodoro](#)

- BBC “acusar a petrolera de contaminar el suministro de agua potable”

[China: acusan a petrolera de contaminar el suministro de agua potable - BBC News Mundo](#)

- La portada de Canadá “indignante brote de E.coli”

[Indignante brote de E. coli en Canadá manda a 264 niños al hospital, algunos están muy graves | La Portada Canadá](#)

- CiberCuba “Aguas de La Habana saca una enorme claria de una alcantarilla”

[Aguas de La Habana saca una enorme claria de una alcantarilla](#)

- Cubita New “Extrae enorme claria durante limpieza por el Meteoro 2025”

[Aguas de La Habana extrae enorme claria durante limpieza por el Meteoro 2025](#)

- CiberCuba “Alarma entre cubanos por estado crítico de la potabilizadora”

[Alarma entre cubanos por estado crítico de la potabilizadora de La Habana](#)

- Tribuna cu “Afectaciones en el servicio de agua potable”

[Tribuna de La Habana: Afectaciones en el servicio de agua potable](#)

- Acades cl “Minería y desalación”

[Minería y desalación: ¿El nuevo dilema hídrico de Chile?](#)

- Estudiando “La Crisis del Agua en Chile”

[La Crisis del Agua en Chile: Causas, Impactos y Soluciones](#)

- Reporte Minero “impacto de la minería en la seguridad hídrica”

[El positivo impacto de la minería en la seguridad hídrica del país](#)

- opsur.org.ar “La lucha por el agua”

[La lucha por el agua: la minería la malgasta y los pueblos la pierden – Observatorio Petrolero Sur](#)

- miningpress.com “Cómo enfrentan las empresas argentinas la crisis hídrica”

Cómo enfrentan las empresas argentinas la crisis hídrica - Mining Press

- udvabony.com “Transformador”

<https://udvabony.com/product/transformer-12v-3a/>

- electronicabyp.com ”caudalimetro”

<https://www.electronicabyp.com.ar/producto/caudalimetro-6-ltr-min-yf-s401/>

- forum.fritzing.org “ Rele”

<https://forum.fritzing.org/uploads/default/original/2X/d/d52c7b2debb250b356c5467353c24f6443a05b05.jpeg>

- rlocman.ru “datasheet rele”

<https://www.rlocman.ru/datasheet/img.php?di=135545>

- i.ibb.co “ili9488”

<https://i.ibb.co/j3hLg8J/TFT-4-T-SPI-ILI9488-003.jpg>

- mecha-tronx.com “capacitor”

<https://mecha-tronx.com/wp-content/uploads/2022/08/2200uF-25V-Electrolytic-Capacitor-750x750-2.png>

Anexo I: Software

Para el diseño del pcb se utilizó el software Proteus 8 professional v8.10 sp3. Proteus es un software de simulación y diseño electrónico desarrollado por Labcenter Electronics. Se optó por este porque permite crear, simular y probar circuitos electrónicos antes de implementarlos físicamente.

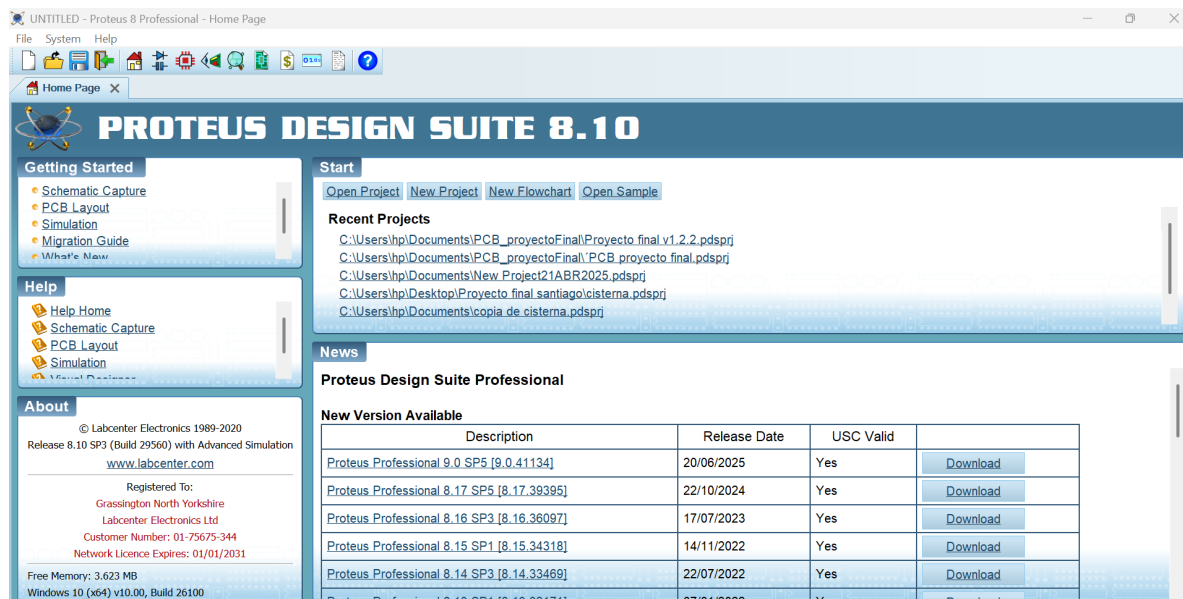


fig. 4.3 Panel de inicio Proteus

Anexo II: esquemático, modelo 3D, gerber y PCB

A continuación se adjuntan imágenes sobre el esquemático, el modelo 3D , gerbers y el circuito impreso del dispositivo.

I. Esquemáticos

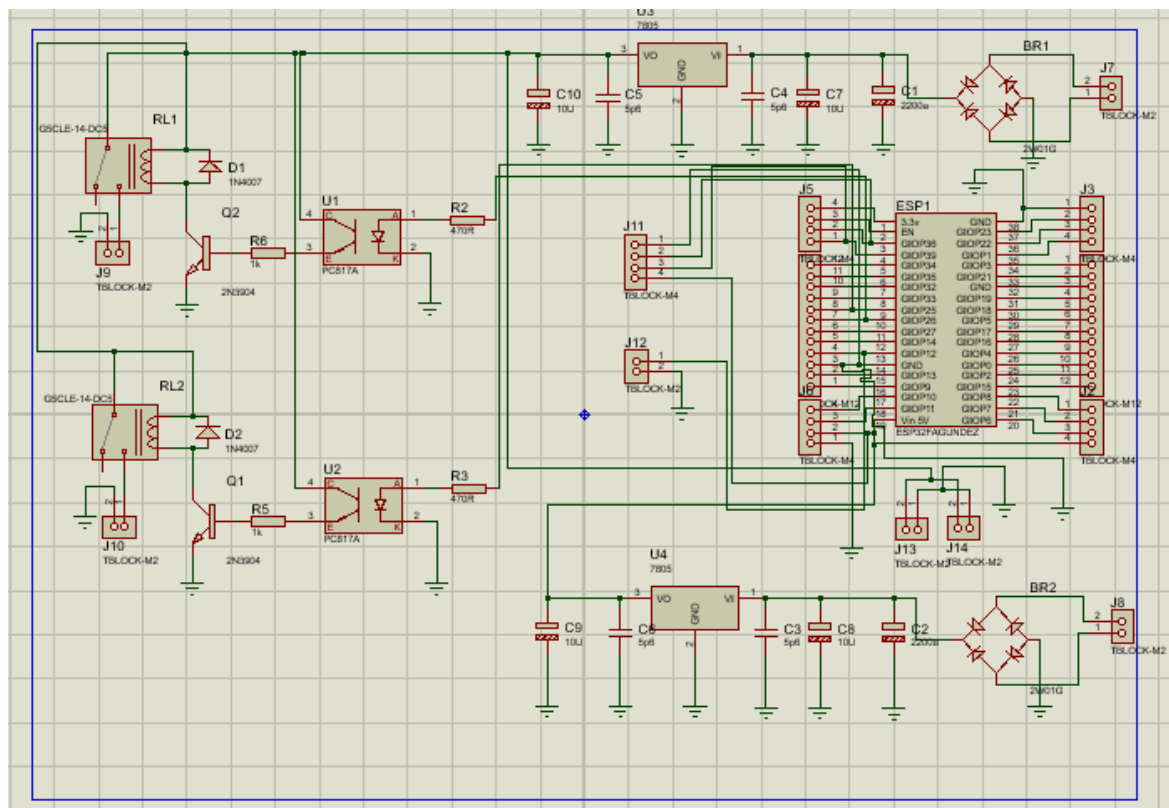


fig. 4.4 Esquemático

II. Modelo 3D

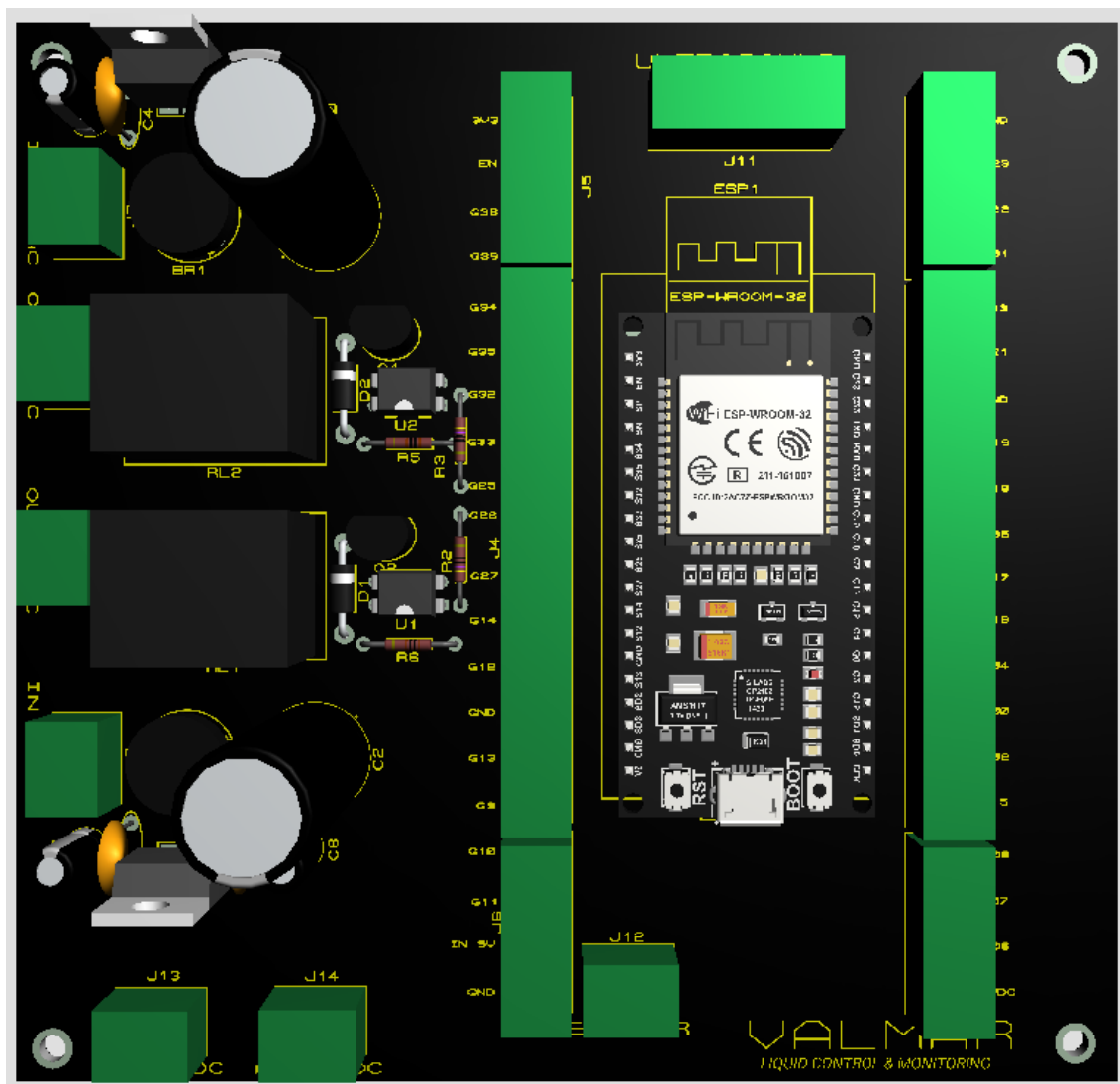


fig. 4.5 Modelo 3D frente

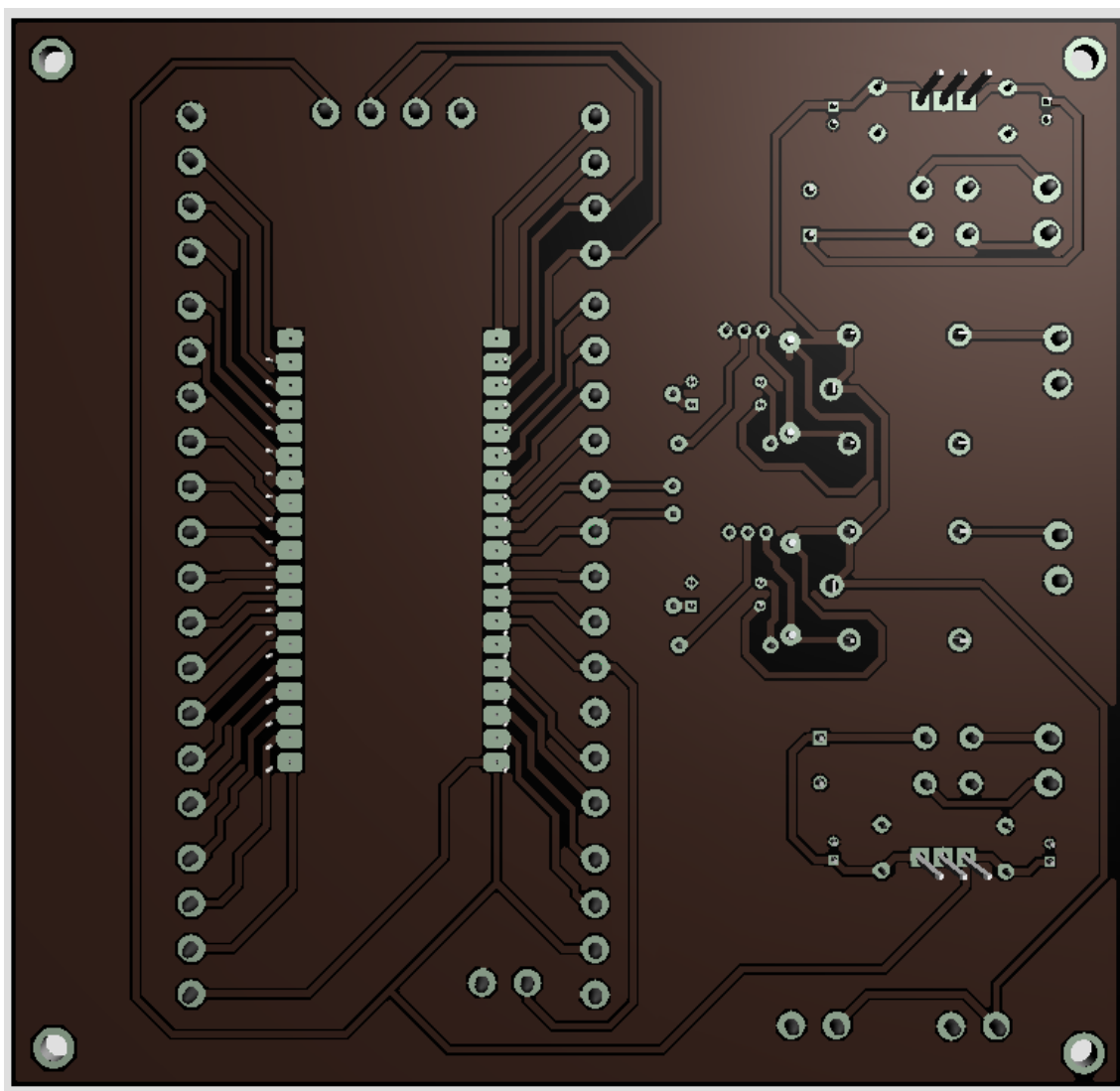


fig. 4.6 Modelo 3D dorso

III. Gerber

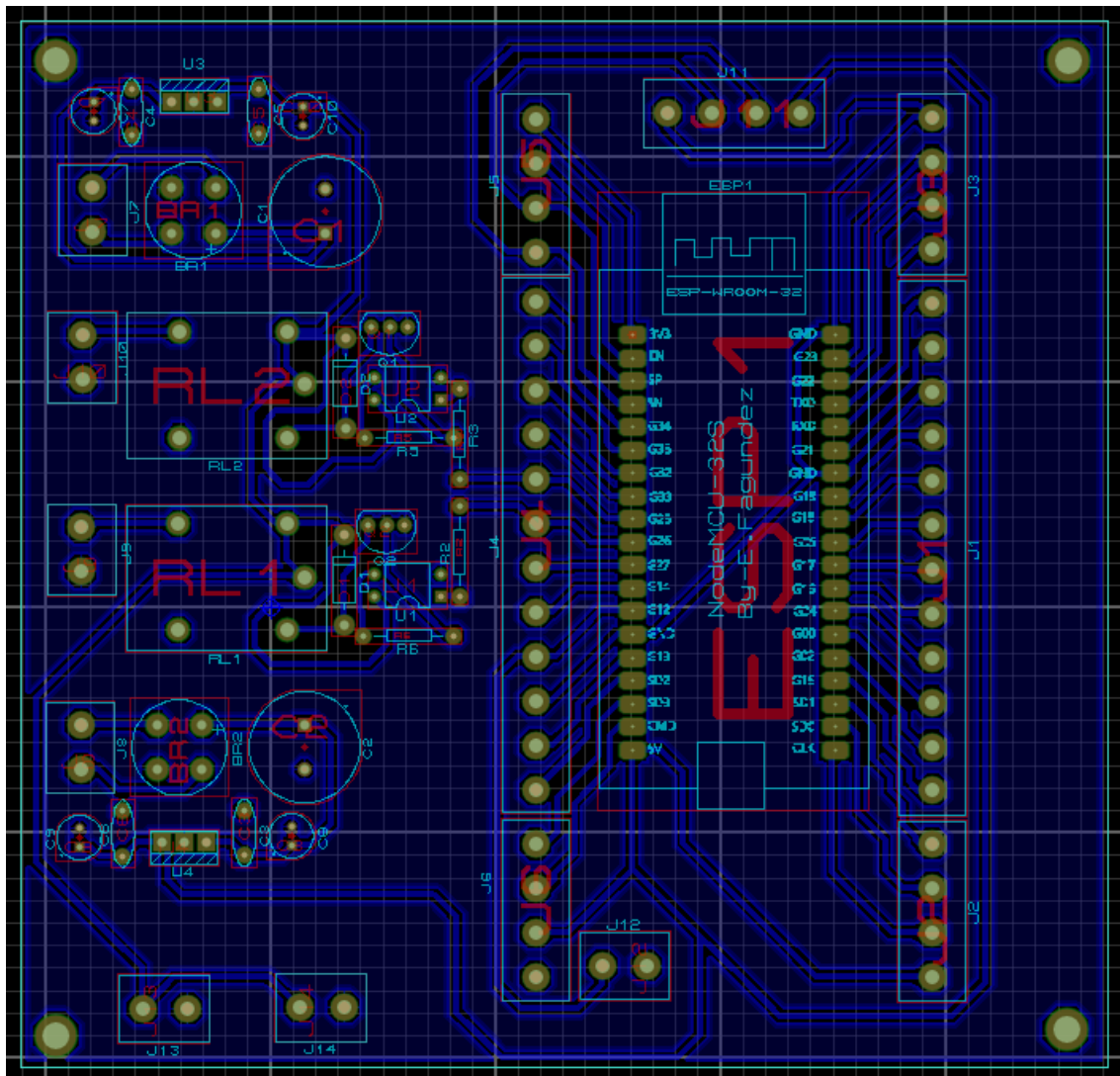


fig. 4.7 Gerber

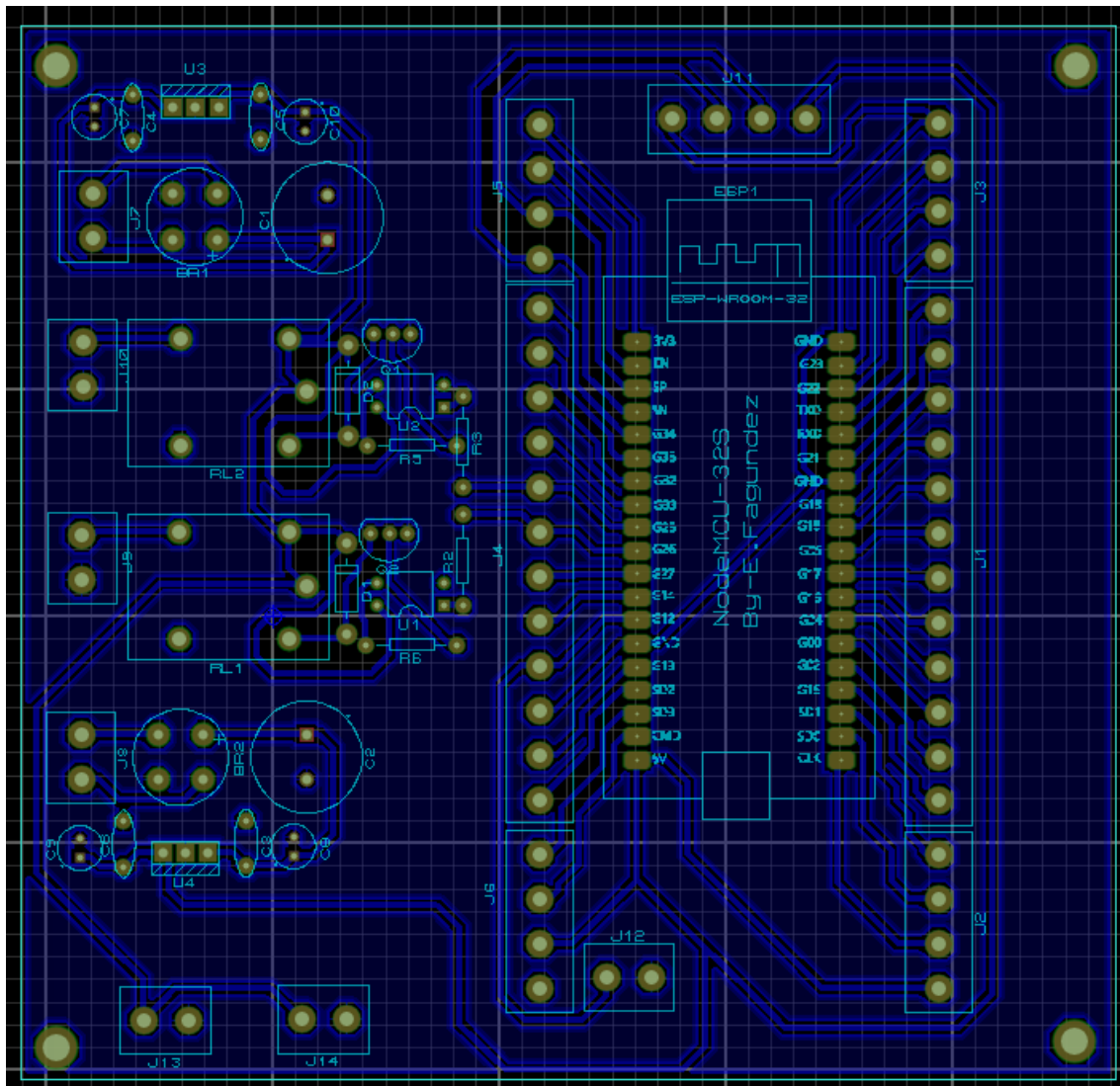


fig. 4.8 Gerber

IV. PCB

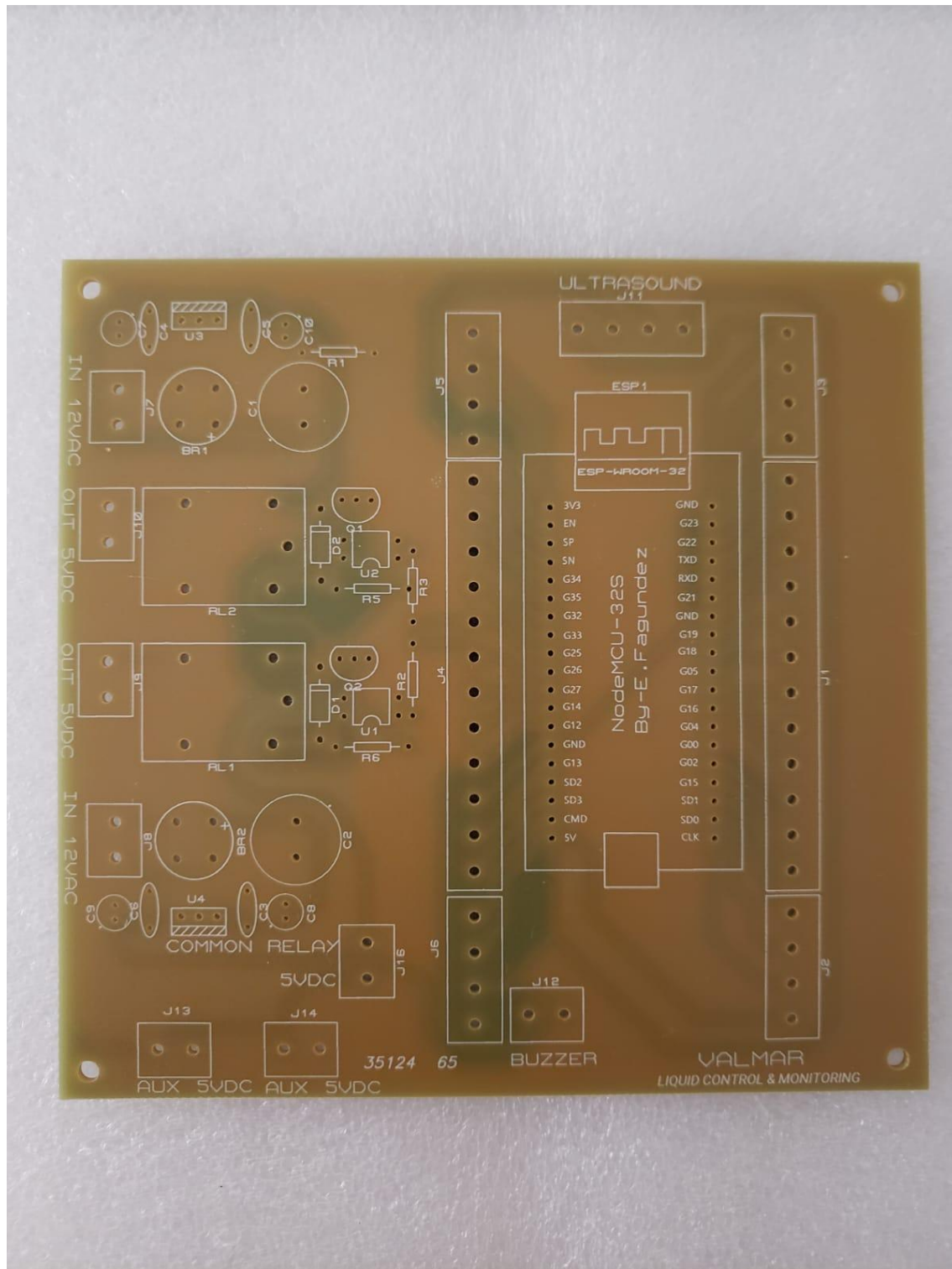


fig. 5.0 Frente pcb

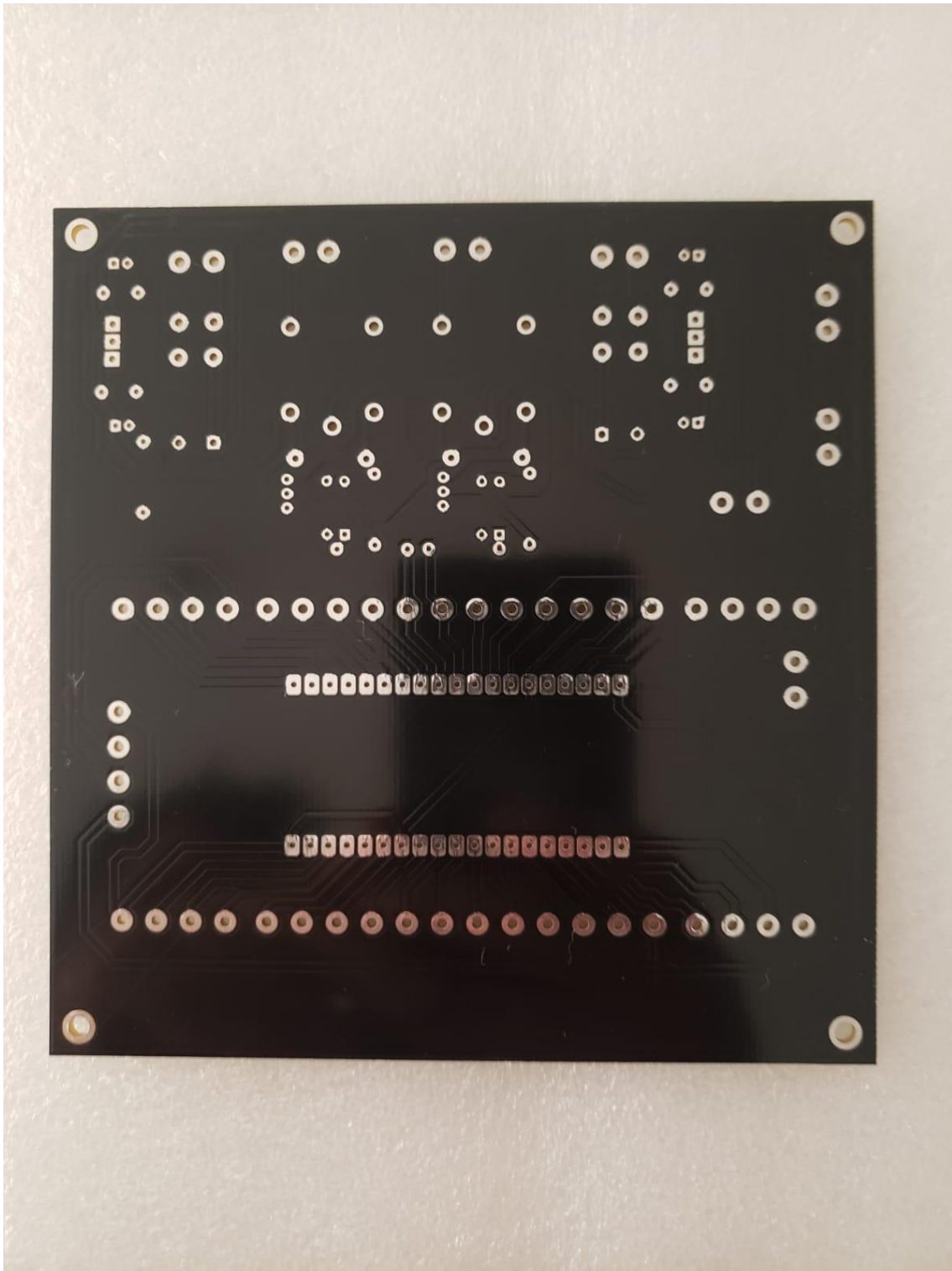


fig. 5.1 Dorso pcb

Anexo III: Código para control y monitoreo, envío a la nube y uso de periféricos

```
#include "Wire.h"
#include "ThingSpeak.h"
#include "WiFi.h"
#include "FS.h"
#include <SPI.h>
#include <TFT_eSPI.h>
#define POT_PIN 14 // Define el pin analógico donde está conectado el
potenciómetro
#define BLACK_SPOT
#define SCREEN_WIDTH 240
#define SCREEN_HEIGHT 320
TFT_eSPI tft = TFT_eSPI();

const char *ssid = "Personal Samsung SM-G96001757";
const char *password = "tepresto";

unsigned long channelID = 2925857;
const char *WriteAPIkey = "BEF89K8SALOII15A";

WiFiClient cliente;

// Variables para los sensores A, B y C
volatile unsigned long pulseCountA = 0;
volatile unsigned long pulseCountB = 0;
volatile unsigned long pulseCountC = 0;

float consumptionLitersA = 0.0;
float previousconsumptionLitersA = 0.0;
float consumptionLitersB = 0.0;
float previousconsumptionLitersB = 0.0;
float consumptionLitersC = 0.0;
float previousconsumptionLitersC = 0.0;
float totalLiters = 0.0;

// Pines para los sensores
const int pinFlowA = 14;
const int pinFlowB = 33;//26
const int pinFlowC = 34;//27
```

```
unsigned long previousMillis = 0;
```

```

// Funciones de interrupción
void IRAM_ATTR countPulseA() {
    pulseCountA++;
}

void IRAM_ATTR countPulseB() {
    pulseCountB++;
}

void IRAM_ATTR countPulseC() {
    pulseCountC++;
}

int rawValue = analogRead(POT_PIN);
float measureTank = map(rawValue, 0, 1023, 0, 400); // Convertir a cm
int CurrentWaterLevel;
int WaterLevelStatus;

const int EMPTY_LEVEL_LIMIT = 80;
const int HALF_LEVEL_LIMIT = 50;

int valueMenu = 0;
float statusTank;
int showflowSwitchA = consumptionLitersA;
int previousshowflowSwitchA = 0;
bool showflowSwitchB;
bool showflowSwitchC;
bool statusPumpOne;
bool statusPumpTwo;
bool statusPump;
int Echo = 36;
int Trigger = 39;
int Buzzer = 12;

int Pump = 25;
String pumpInScreen;
String Full;
String Medium;
String Low;
int LedRedPin = 35;
int LedYellowPin = 32;
int LedGreenPin = 33;
int retryConn = 0; // reintentos de conexion
int retryConnMax = 5; // maxima cantidad de reintentos

```

```

/*
 *
=====
=====
 *
 *
=====
=====
 */
void CalculateWaterTankLevel();
void UltraSound();
void IndicatorLevelled();
void inicioTft();
void pantallaInicio();
void Pins();
void Screen();
void instruction();
void tankVolume();
void Screen();

/*
=====
=====
 *
 *
=====
=====
 */

void setup() {
    Serial.begin(115200);

    conection();
    inicioTft();
    pantallaInicio();
    instruction();
    tankVolume();
    Screen();
    showValue();
    Pins();

    pinMode(pinFlowA, INPUT_PULLUP);
    pinMode(pinFlowB, INPUT_PULLUP);
    pinMode(pinFlowC, INPUT_PULLUP);

    // Asociar interrupciones

```

```

attachInterrupt(digitalPinToInterrupt(pinFlowA), countPulseA, RISING);
attachInterrupt(digitalPinToInterrupt(pinFlowB), countPulseB, RISING);
attachInterrupt(digitalPinToInterrupt(pinFlowC), countPulseC, RISING);
}
/*
=====
=====
*
*
*
=====
=====
*/
void loop() {

    // delay(4000);
    screenData();
    CurrentWaterLevel = measureWater();
    CalculateWaterTankLevel();
    IndicatorLevelLed();
    IndicatorBuzzer();
    drawWifiAntenna();
    pumpOnOff();
    // showPumpSatateOne();
    // showPumpSatateTwo();
    showFlowSwitchStateA();
    showFlowSwitchStateB();
    showFlowSwitchStateC();
    // dibujarIconoWiFi(440, 98, TFT_DARKCYAN);
    serialMonitor();

    uint16_t x, y; // DEFINE ENTERO SIN SIGNO,BUSCA SU DIRECCION EN EJES
X;Y
    if (tft.getTouch(&x, &y))
    {
        int buttonX = (SCREEN_WIDTH - 0, 260);
        int buttonY = (SCREEN_HEIGHT - 0, 210);
        // Detecta si el botón "MENU" ha sido tocado
        if ((x > buttonX) && (x < buttonX + 260) && (y > buttonY) && (y
< buttonY + 210))
        {

            Serial.println("exit");
        }
    }
}

```

```

    uint16_t x1, y1; // DEFINE ENTERO SIN SIGNO,BUSCA SU DIRECCION EN
EJES X;Y
    if (tft.getTouch(&x, &y))
    {
        int buttonX1 = (SCREEN_WIDTH - 0, 260);
        int buttonY1 = (SCREEN_HEIGHT - 0, 10);
        // Detecta si el botón "ATRAS" ha sido tocado
        if ((x > buttonX1) && (x < buttonX1 + 260) && (y > buttonY1) &&
(y < buttonY1 + 10))
        {
            Serial.println("atras");
        }
    }

    uint16_t x4, y4; // DEFINE ENTERO SIN SIGNO,BUSCA SU DIRECCION EN
EJES X;Y
    if (tft.getTouch(&x, &y))
    {
        int buttonX4 = (SCREEN_WIDTH - 0, 260);
        int buttonY4 = (SCREEN_HEIGHT - 0, 100);
        // Detecta si el botón "reset" ha sido tocado
        if ((x > buttonX4) && (x < buttonX4 + 260) && (y > buttonY4) &&
(y < buttonY4 + 100))
        {
            valueMenu = 0;
            Serial.println("RESET");
            //delay(500);
        }
    }

    uint16_t x2, y2; // DEFINE ENTERO SIN SIGNO,BUSCA SU DIRECCION EN
EJES X;Y
    if (tft.getTouch(&x, &y))
    {
        int buttonX2 = (SCREEN_WIDTH - 0, 220);
        int buttonY2 = (SCREEN_HEIGHT - 0, 30);
        // Detecta si el botón "decena" ha sido tocado
        if ((x > buttonX2) && (x < buttonX2 + 220) && (y > buttonY2) &&
(y < buttonY2 + 20))
        {
            valueMenu = valueMenu + 10;
            Serial.println(valueMenu);
            //delay(500);
        }
    }

```

```

    uint16_t x3, y3; // DEFINE ENTERO SIN SIGNO, BUSCA SU DIRECCION EN
    EJES X;Y
    if (tft.getTouch(&x, &y))
    {
        int buttonX3 = (SCREEN_WIDTH - 0, 100);
        int buttonY3 = (SCREEN_HEIGHT - 0, 20);
        // Detecta si el botón "centena" ha sido tocado
        if ((x > buttonX3) && (x < buttonX3 + 100) && (y > buttonY3) &&
        (y < buttonY3 + 20))
        {
            valueMenu = valueMenu + 100;
            Serial.println(valueMenu);
            // delay(500);
        }
    }

    unsigned long currentMillis = millis();

    if (currentMillis - previousMillis >= 1000) { // Cada 1 segundo
        // Detenemos interrupciones para evitar conflictos mientras
        calculamos
        detachInterrupt(digitalPinToInterrupt(pinFlowA));
        detachInterrupt(digitalPinToInterrupt(pinFlowB));
        detachInterrupt(digitalPinToInterrupt(pinFlowC));

        // Cálculo de flujo (pulsos → L/min → L/s)
        float litersA = (pulseCountA / 7.5) / 60.0;
        float litersB = (pulseCountB / 7.5) / 60.0;
        float litersC = (pulseCountC / 7.5) / 60.0;

        // Acumulamos
        consumptionLitersA += litersA;
        consumptionLitersB += litersB;
        consumptionLitersC += litersC;

        // Total general
        totalLiters = consumptionLitersA + consumptionLitersB +
        consumptionLitersC;

        // Mostrar en el monitor serie
        Serial.println("===== LITROS POR SENSOR =====");
        Serial.print("Sensor A: "); Serial.println(consumptionLitersA, 3);
        Serial.print("Sensor B: "); Serial.println(consumptionLitersB, 3);
        Serial.print("Sensor C: "); Serial.println(consumptionLitersC, 3);
        Serial.println("-----");
        Serial.print("TOTAL acumulado: "); Serial.println(totalLiters, 3);
    }

```

```

Serial.println("=====");
Serial.println();

// Reset de pulsos
pulseCountA = 0;
pulseCountB = 0;
pulseCountC = 0;
previousMillis = currentMillis;

// Restauramos interrupciones
attachInterrupt(digitalPinToInterrupt(pinFlowA), countPulseA,
RISING);
attachInterrupt(digitalPinToInterrupt(pinFlowB), countPulseB,
RISING);
attachInterrupt(digitalPinToInterrupt(pinFlowC), countPulseC,
RISING);
tft.setTextColor(TFT_RED); // color de texto en
blanco////////////////////
////////////////////MUESTRA VALUE MENU RED
tft.setTextSize(4); // escala de texto en 2
tft.setCursor(300, 165); // ubica cursor
(200,222)
tft.print(totalLiters); // imprime texto

// tft.setTextColor(TFT_WHITE); // color de texto en
blanco////////////////////
////////////////////MUESTRA VALUE MENU RED
// tft.setTextSize(4); // escala de texto en
2
// tft.setCursor(300, 165); // ubica cursor
(200,222)
//
//tft.fillRect(260, 120, 195, 90, TFT_WHITE);
tft.fillRect(260, 165, 195, 40, TFT_WHITE);

tft.setTextColor(TFT_RED); // color de texto en
blanco////////////////////
////////////////////MUESTRA VALUE MENU RED
tft.setTextSize(4); // escala de texto en 2
tft.setCursor(300, 165); // ubica cursor
(200,222)
tft.print(totalLiters); // imprime texto
}
}

/*

```

```

*
=====
=====
*
*
=====
=====
*/

void conection(){
    Serial.begin(115200);
    WiFi.begin(ssid, password);
    // ahora nos fijamos que no este conectado y que no haya alcanzado
    el maximo de reintentos
    while (WiFi.status() != WL_CONNECTED && retryConn <= retryConnMax)
    {
        retryConn++; // sumo 1 al contador de intentos
        delay(1000); // aca eran 500
        Serial.print(".");
        Serial.println("CONNECTING");
    }

    if (WL_CONNECTED)
    {
        Serial.println("SUCCESSFUL CONNECTION");
        ThingSpeak.begin(cliente);
    }
}

void inicioTft()
{
    Serial.begin(115200);
    tft.init();
    tft.setRotation(1);
}

void pantallaInicio()
{
    tft.fillScreen(TFT_CYAN);
    tft.fillRect(0, 0, 320, 150, TFT_WHITE);
    tft.setTextColor(TFT_BLUE);
    tft.setTextSize(5);
    tft.drawCentreString("VALMAR", 160, 20, 2); //( "VALMAR", 160, 20,
2);
    tft.setTextColor(TFT_RED);
    tft.setTextSize(2);

```

```

    tft.drawCentreString("monitoring and measuring", 160, 100, 2);
    tft.setTextSize(1);
    tft.setTextColor(TFT_BLACK);
    tft.drawString("FAGUNDEZ ELECTRONICS", 10, 220);
    delay(8000);
}

void instruction()
{
    tft.fillScreen(TFT_WHITE);          ////////////////
    PANTALLA NEGRA
    tft.fillRect(0, 0, 480, 320, TFT_WHITE); // Total de la pantalla
    blanca////////////////////////FONDO BLANCO DE TODA LA PANTALLA
    tft.fillRect(10, 10, 460, 300, TFT_DARKCYAN); // rectangulo cyan
    oscuro a modo
    tft.fillRect(15, 15, 450, 290, TFT_WHITE);
    // tft.fillScreen(TFT_BLACK);
    tft.setTextColor(TFT_BLACK);
    tft.setTextSize(2);
    tft.drawCentreString("Felicitaciones, usted adquirio un medidor ",
160, 20, 2);
    tft.drawCentreString("controlador de liquidos ", 100, 40, 2);
    tft.drawCentreString("Con este sistema podra ver el estado de ",
160, 60, 2);
    tft.drawCentreString("su tanque y registrar el consumo en lts ",
150, 80, 2);
    tft.drawCentreString("recuerde https://thingspeak.mathworks.com/ ",
150, 100, 2);
    tft.drawCentreString("insertando sus credenciales puede ver", 150,
120, 2);
    tft.drawCentreString("el funcionamiento en vivo del sistema.", 150,
140, 2);
    tft.drawCentreString("Por consultas tecnicas contactenos.", 150,
160, 2);
    tft.setTextColor(TFT_RED);
    tft.setTextSize(1);
    tft.drawCentreString("valmar.questions@gmail.com", 150, 180, 2);
    tft.setTextColor(TFT_SKYBLUE);
    tft.setTextSize(1);
    tft.drawCentreString(" VALMAR ", 200, 40, 2);
    delay(8000);
}

void tankVolume()
{
    tft.fillScreen(TFT_BLACK);

```

```

tft.setTextColors(TFT_SKYBLUE);
tft.setTextSize(2);
tft.drawCentreString("Verifique la altura en ", 180, 0, 2);
tft.drawCentreString("rojo sea la correcta y ", 180, 20, 2);
tft.drawCentreString("aguarde,sino seleccionela ", 180, 40, 2);
tft.drawCentreString("desde set y pulse reset. ", 180, 60, 2);
tft.setTextColors(TFT_RED);
tft.setTextSize(3);
tft.setCursor(100, 110);
tft.print(measureTank); // Aquí se reemplaza "NNN" con el valor del
potenciómetro
tft.print(" cm");
tft.setTextColors(TFT_GREEN);
tft.setTextSize(2);
tft.drawCentreString(" Altura en centimetros ", 180, 140, 2);
delay(8000);
}

void Screen()
{
    tft.fillScreen(TFT_BLACK); // PANTALLA NEGRA
    tft.fillRect(0, 0, 480, 320, TFT_WHITE); // Total de la pantalla
    blanca////////FONDO BLANCO DE TODA LA PANTALLA
    tft.fillRect(0, 0, 480, 50, TFT_DARKCYAN); // rectangulo cyan oscuro
    a modo de encabezado////////ENCABEZADO CYAN OSCURO
    tft.loadFont("Roboto18"); // VALMAR COLOR
    BLANCO
    tft.setTextColors(TFT_WHITE); // color de texto en
    blanco
    tft.setTextSize(3); // escala de texto en 2
    tft.setCursor(8, 15); // ubica cursor
    tft.print("VALMAR"); // imprime texto VALMAR
    // tft.unloadFont();
    tft.setTextSize(2); // escala de
    texto en 2////////LIQUID CONTROL & MONITORING COLOR BLANCO
    tft.setCursor(110, 20); // ubica cursor
    tft.print(" liquid control & monitoring"); // imprime
    texto liquid control & monitoring
    tft.fillRect(10, 60, 460, 205, TFT_DARKCYAN); // rectangulo
    azul naval a modo de fondo de titulo////////RECTANGULO CYAN
    OSCURO DEBAJO DEL ENCABEZADO
    tft.setTextColors(TFT_WHITE); // color de
    texto en blanco////////MONITORING COLOR BLANCO
    tft.setTextSize(2); // escala de
    texto en 2

```

```

        tft.setCursor(180, 63); // ubica cursor
        tft.print("MONITORING"); // imprime
texto
    tft.fillRect(25, 80, 430, 30, TFT_WHITE);
    ///////////RECTANGULO BLANCO FINO
    tft.setTextColor(TFT_BLACK); // color de
texto en blanco ///////////SYSTEM STATUS:NNNNN
    tft.setTextSize(2); // escala de
texto en 2
    tft.setCursor(120, 88); // ubica cursor
    tft.print("SYSTEM STATUS:"); // imprime
texto
    tft.fillRect(25, 120, 200, 90, TFT_WHITE);
    ///////////RECTANGULO BLANCO TANK STATUS
    tft.setTextColor(TFT_BLACK); // color de
texto en blanco ///////////ESCRIBE TANK STATUS
    tft.setTextSize(2); // escala de
texto en 2
    tft.setCursor(60, 123); // ubica cursor
    tft.print("TANK STATUS"); // imprime
texto VALMAR
    tft.fillRect(260, 120, 195, 90, TFT_WHITE);
    ///////////RECTANGULO BLANCO CONSUMPTION IN LITERS
    tft.setTextColor(TFT_BLACK); // color de
texto en blanco//////////ESCRIBE CONSUMPTION
    tft.setTextSize(2); // escala de
texto en 2
    tft.setCursor(290, 123); // ubica cursor
    tft.print("CONSUMPTION "); // imprime
texto VALMAR
    tft.setTextColor(TFT_BLACK); // color de
texto en blanco//////////ESCRIBE IN LITERS
    tft.setTextSize(2); // escala de
texto en 2
    tft.setCursor(300, 140); // ubica cursor
    tft.print("IN LITERS"); // imprime
texto VALMAR
    tft.fillRect(25, 220, 430, 30, TFT_WHITE);
    ///////////RECTANGULO BLANCO TOTAL LITERS & RESET
    tft.fillRect(10, 275, 460, 40, TFT_DARKCYAN);
    ///////////RECTANGULO CYAN OSCURO
    tft.fillRect(20, 280, 350, 30, TFT_WHITE);
    ///////////RECTANGULO BLANCO FINO
    int buttonX4 = (SCREEN_WIDTH - 0, 386); // Posición X
DEL BOTON // no tocar
    int buttonY4 = (SCREEN_HEIGHT - 0, 282); // Posición Y

```

```

DEL BOTON //210
    tft.fillRect(buttonX4, buttonY4, 70, 26, TFT_WHITE); // TAMAÑO DEL
BOTON
    tft.setTextColor(TFT_BLACK);
    tft.setTextSize(2);
    tft.setTextDatum(MC_DATUM);
    tft.drawString("RESET", buttonX4 + 35, buttonY4 + 13); // CENTRADO
DEL NOMBRE DEL BOTON
    // tft.setTextColor(TFT_WHITE);
    // tft.setTextSize(3);
    // tft.setCursor(150,30);
    // tft.print("AJUSTE ALTURA");
    tft.setTextColor(TFT_SKYBLUE);
    // tft.drawLine(0, 55, 480, 55, TFT_RED); // linea horizontal de
color rojo 170
    tft.setTextColor(TFT_WHITE);
    int buttonX = (SCREEN_WIDTH - 0, 260); // Posición X DEL
BOTON////////////////////////////////////
////////////////////////////////////EXIT BUTTON
    int buttonY = (SCREEN_HEIGHT - 0, 310); // Posición Y DEL
BOTON
    tft.fillRect(buttonX, buttonY, 60, 30, TFT_NAVY); // TAMAÑO DEL
BOTON
    tft.setTextColor(TFT_WHITE);
    tft.setTextSize(2);
    tft.setTextDatum(MC_DATUM);
    tft.drawString("EXIT", buttonX + 30, buttonY + 15); // CENTRADO DEL
NOMBRE DEL BOTON
    // int buttonX4 = (SCREEN_WIDTH -0,120 ) ; // Posición X DEL BOTON
//
140////////////////////////////////////
////////////////////////////////////RESET BUTTON
    // int buttonY4 = (SCREEN_HEIGHT -0, 220) ; // Posición Y DEL BOTON
//210
    // tft.fillRect(buttonX4, buttonY4, 60, 30, TFT_NAVY); //TAMAÑO DEL
BOTON
    // tft.setTextColor(TFT_WHITE);
    // tft.setTextSize(2);
    // tft.setTextDatum(MC_DATUM);
    // tft.drawString("RESET", buttonX4 + 30, buttonY4 + 15);//CENTRADO
DEL NOMBRE DEL BOTON
    int buttonX1 = (SCREEN_WIDTH - 0, 0); // Posición X
DEL
BOTON////////////////////////////////////
////////////////////////////////////ATRAS BUTTON
    int buttonY1 = (SCREEN_HEIGHT - 0, 310); // Posición Y

```

```

DEL BOTON
    tft.fillRect(buttonX1, buttonY1, 60, 30, TFT_NAVY); // TAMAÑO DEL
BOTON
    tft.setTextColor(TFT_WHITE);
    tft.setTextSize(2);
    tft.setTextDatum(MC_DATUM);
    tft.drawString("ATRAS", buttonX1 + 30, buttonY1 + 15); // CENTRADO
DEL NOMBRE DEL BOTON
    // int buttonX2 = (SCREEN_WIDTH -0,0 ) ;    // Posición X DEL BOTON//
BOTONES DECENA CENTENA
    // int buttonY2 = (SCREEN_HEIGHT -0,60) ; // Posición Y DEL BOTON
    // tft.fillRect(buttonX2, buttonY2, 120, 60, TFT_WHITE); //TAMAÑO
DEL BOTON
    // tft.setTextColor(TFT_BLACK);
    // tft.setTextSize(3);
    // tft.setTextDatum(MC_DATUM);
    // tft.drawString("SUMA 100", buttonX2 + 60, buttonY2 +
30);//CENTRADO DEL NOMBRE DEL BOTON
    // int buttonX3 = (SCREEN_WIDTH -0,0 ) ;    // Posición X DEL BOTON
    // int buttonY3 = (SCREEN_HEIGHT -0,140) ; // Posición Y DEL BOTON
    // tft.fillRect(buttonX3, buttonY3, 120, 60, TFT_WHITE); //TAMAÑO
DEL BOTON
    // tft.setTextColor(TFT_BLACK);
    // tft.setTextSize(3);
    // tft.setTextDatum(MC_DATUM);
    // tft.drawString("SUMA 10", buttonX3 + 60, buttonY3 +
30);//CENTRADO DEL NOMBRE DEL BOTON
    tft.setTextColor(TFT_BLACK); // color de texto en blanco
    tft.setTextSize(2);          // escala de texto en 2
    tft.setCursor(26, 287);      // ubica cursor
    tft.print("PUMP STATE");     // imprime texto
    tft.setTextColor(TFT_BLACK); // color de texto en blanco
    tft.setTextSize(2);          // escala de texto en 2
    tft.setCursor(176, 287);     // ubica cursor
    tft.print("ONE");            // imprime texto
    tft.setTextColor(TFT_BLACK); // color de texto en blanco
    tft.setTextSize(2);          // escala de texto en 2
    tft.setCursor(276, 287);     // ubica cursor
    tft.print("TWO");            // imprime texto
}

void showValue()
{
    tft.setTextColor(TFT_BLUE);          // color de texto en
blanco
    tft.setTextSize(3);                  // escala de texto en 2

```

```

    tft.fillRect(270, 165, 180, 40, TFT_WHITE); // rectangulo azul naval
a modo de fondo de titulo////////////////////////////////////
    tft.setCursor(350, 180);                      // ubica cursor
    tft.setTextColor(TFT_BLACK);                  // color de texto en
blanco////////////////////////////////////
////////////////////////////////////MUESTRA FLOW SWITCH STATE
    tft.setTextSize(2);                          // escala de texto en 2
    tft.setCursor(28, 228);                      // ubica cursor
    tft.print("FLOW SWITCH STATE");              // imprime texto
    tft.setTextColor(TFT_BLACK);                // color de texto en
blanco////////////////////////////////////
////////////////////////////////////MUESTRA FLOW A
    tft.setTextSize(2);                          // escala de texto en 2
    tft.setCursor(250, 228);                    // ubica cursor
    tft.print("A");                             // imprime texto
    tft.setTextColor(TFT_BLACK);                // color de texto en
blanco////////////////////////////////////
////////////////////////////////////MUESTRA FLOW B
    tft.setTextSize(2);                          // escala de texto en 2
    tft.setCursor(310, 228);                    // ubica cursor
    tft.print("B");                             // imprime texto
    tft.setTextColor(TFT_BLACK);                // color de texto en
blanco////////////////////////////////////
////////////////////////////////////MUESTRA FLOW C
    tft.setTextSize(2);                          // escala de texto en 2
    tft.setCursor(370, 228);                    // ubica cursor
    tft.print("C");                             // imprime texto

////////////////////////////////////
////////////////////////////////////RESET BUTTON
    // tft.setTextColor(TFT_BLACK); // color de texto en blanco
    // tft.setTextSize(3);          // escala de texto en 2
    // tft.setCursor(330,220);      // ubica cursor
    // tft.print("RESET");          // imprime texto

    // tft.setTextColor(TFT_SKYBLUE); // color de texto en blanco
    // tft.setTextSize(2);          // escala de texto en 2
    // tft.setCursor(125,60);       // ubica cursor
    // tft.print("Ejemplo 220cm "); // imprime texto
    //
    // tft.setTextColor(TFT_GOLD);
    //
    // tft.setCursor(125,80);       // ubica cursor
    // tft.print("centena centena"); // imprime texto

```

```

    //
    //
    //    tft.setCursor(125,100);        // ubica cursor
    //    tft.print("decena decena");    // imprime texto
}

void Pins()
{
    pinMode(POT_PIN, INPUT);
    pinMode(Pump, OUTPUT);
    digitalWrite(Pump,LOW);
    pinMode(Buzzer, OUTPUT);
    pinMode(Triquer, OUTPUT);
    pinMode(Echo, INPUT);
    pinMode(LedRedPin, OUTPUT);
    pinMode(LedYellowPin, OUTPUT);
    pinMode(LedGreenPin, OUTPUT);
    digitalWrite(Triquer, LOW); // Inicializamos el pin con 0
    digitalWrite(LedRedPin, LOW);
    digitalWrite(LedYellowPin, LOW);
    digitalWrite(LedGreenPin, LOW);
    noTone(Buzzer);
}

/*
*
=====
=====
*                               FUNCIONES DEL LOOP
*
=====
=====
*/

void screenData()
{
    if (WaterLevelStatus == 2)
    {
        tft.setTextColor(TFT_WHITE, TFT_WHITE); //PUSE EMPTY NEGRO
        PARA SACAR EL FANTASMA QUE ME QUEDABA CUANDO CAMBIABA LA IMAGEN
        tft.setTextSize(5);
        tft.setCursor(50, 150);
        tft.print("EMPTY");
        tft.setTextColor(TFT_GREEN, TFT_WHITE); // texto en color
        amarillo
    }
}

```

```

        tft.setTextSize(5);                // escala de texto en 6
        tft.setCursor(50, 150);           // ubica cursor
        tft.print("FULL");                // valor que representa
la temperatura del sensor en zona 1
    }
    else if (WaterLevelStatus == 1)
    {
        tft.setTextColor(TFT_WHITE, TFT_WHITE);
        tft.setTextSize(5);
        tft.setCursor(50, 150);
        tft.print("FULL");
        tft.setTextColor(TFT_WHITE, TFT_WHITE);
        tft.setTextSize(5);
        tft.setCursor(50, 150);
        tft.print("EMPTY");
        tft.setTextColor(TFT_YELLOW, TFT_WHITE); // texto en color
amarillo
        tft.setTextSize(5);                // escala de texto en 6
        tft.setCursor(50, 150);           // ubica cursor
        tft.print("MID");                  // valor que representa
la temperatura del sensor en zona 1
        tft.setTextColor(TFT_WHITE, TFT_WHITE);
    }
    else if (WaterLevelStatus == 0)
    {
        tft.setTextColor(TFT_RED, TFT_WHITE); // texto en color amarillo
        tft.setTextSize(5);                // escala de texto en 6
        tft.setCursor(50, 150);           // ubica cursor
        tft.print("EMPTY");                // valor que representa la
temperatura del sensor en zona 1
        tft.setTextColor(TFT_WHITE, TFT_WHITE);
    }
}

void CalculateWaterTankLevel()
{
    if (CurrentWaterLevel < 30)
    {
        WaterLevelStatus = 2;
    }
    else if (CurrentWaterLevel > 30 && CurrentWaterLevel < 80)
    {
        WaterLevelStatus = 1;
    }
    else if (CurrentWaterLevel > 80)

```

```

    {
        WaterLevelStatus = 0;
    }
    statusTank = WaterLevelStatus;
    if (WL_CONNECTED)
    {
        ThingSpeak.setField(1, statusTank);
    }
}

float measureWater()
{
    float d;
    float t;
    digitalWrite(Triple, HIGH);
    delayMicroseconds(10);
    digitalWrite(Triple, LOW);
    t = pulseIn(Echo, HIGH);
    d = t / 59;
    Serial.print("Distancia: ");
    Serial.print(d);
    Serial.println("");
    return d;
}

void IndicatorLevelled()
{
    if (WaterLevelStatus == 2)
    {
        digitalWrite(LedRedPin, LOW);
        digitalWrite(LedYellowPin, LOW);
        digitalWrite(LedGreenPin, HIGH);
    }
    else if (WaterLevelStatus == 1)
    {
        digitalWrite(LedRedPin, LOW);
        digitalWrite(LedYellowPin, HIGH);
        digitalWrite(LedGreenPin, LOW);
    }
    else if (WaterLevelStatus == 0)
    {
        digitalWrite(LedRedPin, HIGH);
        digitalWrite(LedYellowPin, LOW);
        digitalWrite(LedGreenPin, LOW);
    }
}

```

```

    }
}

void IndicatorBuzzer()
{
    if (CurrentWaterLevel >= HALF_LEVEL_LIMIT && CurrentWaterLevel <
EMPTY_LEVEL_LIMIT)
    {
        tone(Buzzer, 1000, 10);
    }
    else if (CurrentWaterLevel >= EMPTY_LEVEL_LIMIT)
    {
        tone(Buzzer, 1000);
    }
    else
    {
        noTone(Buzzer);
    }
    delay(100);
}

void drawWifiAntenna()
{
    if (WiFi.status() == WL_CONNECTED)
    {
        tft.setTextColor(TFT_WHITE);
        tft.setTextSize(2);
        tft.setCursor(120, 88);
        tft.print("SYSTEM STATUS:OFF LINE");
        tft.setTextColor(TFT_BLACK);
        tft.setTextSize(2);
        tft.setCursor(120, 88);
        tft.print("SYSTEM STATUS:ON LINE");
        dibujarWiFiSinConexion(440, 98, TFT_WHITE, TFT_WHITE);
        dibujarIconoWiFi(440, 98, TFT_DARKCYAN);
    }
    else
    {
        tft.setTextColor(TFT_WHITE);
        tft.setTextSize(2);
        tft.setCursor(120, 88);
        tft.print("SYSTEM STATUS:ON LINE");
        tft.setTextColor(TFT_BLACK);
        tft.setTextSize(2);
        tft.setCursor(120, 88);
    }
}

```

```

        tft.print("SYSTEM STATUS:OFF LINE");
        dibujarWiFiSinConexion(440, 98, TFT_CYAN, TFT_RED);
    }
}

void dibujarIconoWiFi(int x, int y, uint16_t color)
{
    tft.fillCircle(x, y, 2, color);
    for (int r = 6; r <= 15; r += 3)
    {
        for (int i = 220; i <= 320; i += 6)
        {
            float angle = i * 3.1416 / 180;
            int px = x + cos(angle) * r;
            int py = y + sin(angle) * r;
            tft.drawPixel(px, py, color);
        }
    }
}

void dibujarWiFiSinConexion(int x, int y, uint16_t colorBase, uint16_t
colorLinea)
{
    // Antenita base
    tft.fillCircle(x, y, 2, colorBase); // Punto central

    // Arcos de señal
    for (int r = 5; r <= 9; r += 2)
    {
        for (int i = 220; i <= 320; i += 6)
        {
            float angle = i * 3.1416 / 180;
            int px = x + cos(angle) * r;
            int py = y + sin(angle) * r;
            tft.drawPixel(px, py, colorBase);
        }
    }

    // Línea cruzada al revés y más arriba
    int offsetY = -3; // sube la línea un poco
    tft.drawLine(x + 8, y - 10 + offsetY, x - 8, y + 10 + offsetY,
colorLinea);
    tft.drawLine(x + 8, y - 11 + offsetY, x - 8, y + 9 + offsetY,
colorLinea);
}

```

```

void serialMonitor(){
    delay(100);
    Serial.print("WaterLevelStatus: ");
    Serial.print(WaterLevelStatus);
    Serial.println("");
    delay(100);
    ThingSpeak.writeFields(channelID, WriteAPIkey);
    Serial.print("Values sent");
    delay(875);    // lectura real cada 14segundos

}

void pumpOnOff()
{
    if (WaterLevelStatus == 2)
    {
        digitalWrite(25, LOW);
        Serial.print("valuePump");
        Serial.println();
        Serial.println(statusPump);
        ///////////////////////////////////
        tft.setCursor(176, 287);
        //    tft.print("ONE");
        tft.fillRect(220, 286, 15, 15, TFT_WHITE);
        tft.drawRect(220, 286, 15, 15, TFT_BLACK); // Borde por si está
blanco
    }
    else if (WaterLevelStatus == 1)
    {
        digitalWrite(25, HIGH);
        Serial.print("valuePump");
        Serial.println();
        Serial.println(statusPump);
        ///////////////////////////////////
        tft.setCursor(176, 287);
        //    tft.print("ONE");
        tft.fillRect(220, 286, 15, 15, TFT_RED);
        tft.drawRect(220, 286, 15, 15, TFT_BLACK); // Borde por si está
blanco
    }
    else if (WaterLevelStatus == 0)
    {
        digitalWrite(25, HIGH);
        Serial.print("valuePump");
        Serial.println();
        Serial.println(statusPump);
    }
}

```

```

////////////////////////////////////
tft.setCursor(176, 287);
//    tft.print("ONE");
tft.fillRect(220, 286, 15, 15, TFT_RED);
tft.drawRect(220, 286, 15, 15, TFT_BLACK); // Borde por si está
blanco
    }

    statusPump = digitalRead(25);
    if (WL_CONNECTED)
    {
        ThingSpeak.setField(2, statusPump);
    }
}

void showFlowSwitchStateA()
{
    if (consumptionLitersA > previousconsumptionLitersA){

        tft.setCursor(250, 228);
        //    tft.print("ONE");
        tft.fillRect(274, 227, 15, 15, TFT_RED);
        tft.drawRect(274, 227, 15, 15, TFT_BLACK); // Borde por si está
blanco
        Serial.println("¡Se incrementó FlowSwitch A!");
        previousconsumptionLitersA = consumptionLitersA ;

    }
    else
    {
        tft.setCursor(250, 228);
        //    tft.print("ONE");
        tft.fillRect(274, 227, 15, 15, TFT_WHITE);
        tft.drawRect(274, 227, 15, 15, TFT_BLACK); // Borde por si está
blanco
    }
}
////////////////////////////////////
///ESCRIBE "B"
void showFlowSwitchStateB()
{
    if (consumptionLitersB > previousconsumptionLitersB)
    {
        tft.setCursor(250, 228);
        //    tft.print("ONE");

```

```

        tft.fillRect(334, 227, 15, 15, TFT_RED);
        tft.drawRect(334, 227, 15, 15, TFT_BLACK); // Borde por si está
blanco
    }
    else
    {
        tft.setCursor(250, 228);
        // tft.print("ONE");
        tft.fillRect(334, 227, 15, 15, TFT_WHITE);
        tft.drawRect(334, 227, 15, 15, TFT_BLACK); // Borde por si está
blanco
    }
}
////////////////////////////////////
///ESCRIBE "C"
void showFlowSwitchStateC()
{
    if (consumptionLitersC > previousconsumptionLitersC)
    {
        tft.setCursor(250, 228);
        // tft.print("ONE");
        tft.fillRect(394, 227, 15, 15, TFT_RED);
        tft.drawRect(394, 227, 15, 15, TFT_BLACK); // Borde por si está
blanco
    }
    else
    {
        tft.setCursor(250, 228);
        // tft.print("ONE");
        tft.fillRect(394, 227, 15, 15, TFT_WHITE);
        tft.drawRect(394, 227, 15, 15, TFT_BLACK); // Borde por si está
blanco
    }
}

////////////////////////////////////MUESTRA CUADRADO DE COLOR DEPENDIENDO EL
ESTADO DE LA BOMBA ONE////////////////////////////////////
void showPumpSatateOne()
{
    if (statusPumpOne == 1)
    {
        tft.setCursor(176, 287);
        // tft.print("ONE");
        tft.fillRect(220, 286, 15, 15, TFT_WHITE);
        tft.drawRect(220, 286, 15, 15, TFT_BLACK); // Borde por si está
blanco
    }
}

```

```

    }
    else
    {
        tft.setCursor(176, 287);
        // tft.print("ONE");
        tft.fillRect(220, 286, 15, 15, TFT_RED);
        tft.drawRect(220, 286, 15, 15, TFT_BLACK); // Borde por si está
blanco
    }
}

////////////////////////////////////MUESTRA CUADRADO DE COLOR DEPENDIENDO EL
ESTADO DE LA BOMBA TWO
void showPumpSatateTwo()
{
    if (statusPumpTwo == 1)
    {
        tft.setCursor(276, 287);
        // tft.print("ONE");
        tft.fillRect(220, 286, 15, 15, TFT_WHITE);
        tft.drawRect(220, 286, 15, 15, TFT_BLACK); // Borde por si está
blanco
    }
    else
    {
        tft.setCursor(276, 287);
        // tft.print("ONE");
        tft.fillRect(320, 286, 15, 15, TFT_RED);
        tft.drawRect(320, 286, 15, 15, TFT_BLACK); // Borde por si está
blanco
    }
}

```

Anexo IV: Alta en plataforma IoT

Internet de las cosas (IoT)

La Internet de las cosas (IoT) describe la red de objetos físicos ("cosas") que llevan incorporados sensores, software y otras tecnologías con el fin de conectarse e intercambiar datos con otros dispositivos y sistemas a través de Internet. Estos dispositivos van desde objetos domésticos comunes hasta herramientas industriales sofisticadas. La "cosa" (Thing), en el Internet de las cosas puede ser una persona con un implante de monitoreo en el corazón, un animal de granja con un biochip inyectable de identificación, un sensor integrado en un auto para sensor la presión del conductor o cualquier otro tipo de sensor que se quiera recolectar información.

¿Por qué es tan importante el Internet de las Cosas?

En los últimos años, IoT se ha convertido en una de las tecnologías más importantes del siglo XXI. Ahora que podemos conectar objetos cotidianos, electrodomésticos, coches, termostatos, monitores de bebés, a Internet a través de dispositivos integrados, es posible una comunicación fluida entre personas, procesos y cosas. Mediante la informática de bajo costo, la nube, big data, analítica y tecnologías móviles, las cosas físicas pueden compartir y recopilar datos con una mínima intervención humana. En este mundo hiperconectado, los sistemas digitales pueden grabar, supervisar y ajustar cada interacción entre las cosas conectadas. El mundo físico y el digital van de la mano y cooperan entre sí.

Pasos en Thingspeak

ThingSpeak es una plataforma en la nube desarrollada por MathWorks, orientada al almacenamiento, visualización y análisis de datos en tiempo real para aplicaciones del Internet de las Cosas (IoT). Permite a los dispositivos conectados, como microcontroladores, sensores o sistemas embebidos, enviar datos a través de Internet hacia canales de usuario donde quedan registrados de forma estructurada.

La plataforma ofrece una interfaz gráfica accesible mediante navegador, en la que es posible observar datos en forma de gráficos personalizables y realizar análisis complejos gracias a su integración nativa con MATLAB. Además, proporciona herramientas para generar alertas automáticas, ejecutar funciones programadas y responder a eventos en función de condiciones definidas por el usuario.

ThingSpeak resulta especialmente útil en entornos académicos, prototipos de investigación y sistemas de monitoreo remoto, gracias a su compatibilidad con placas como Arduino, ESP32 o Raspberry Pi. Al facilitar la interacción entre hardware y la nube sin necesidad de servidores propios, se convierte en una solución versátil y de rápida implementación para proyectos basados en sensores conectados.

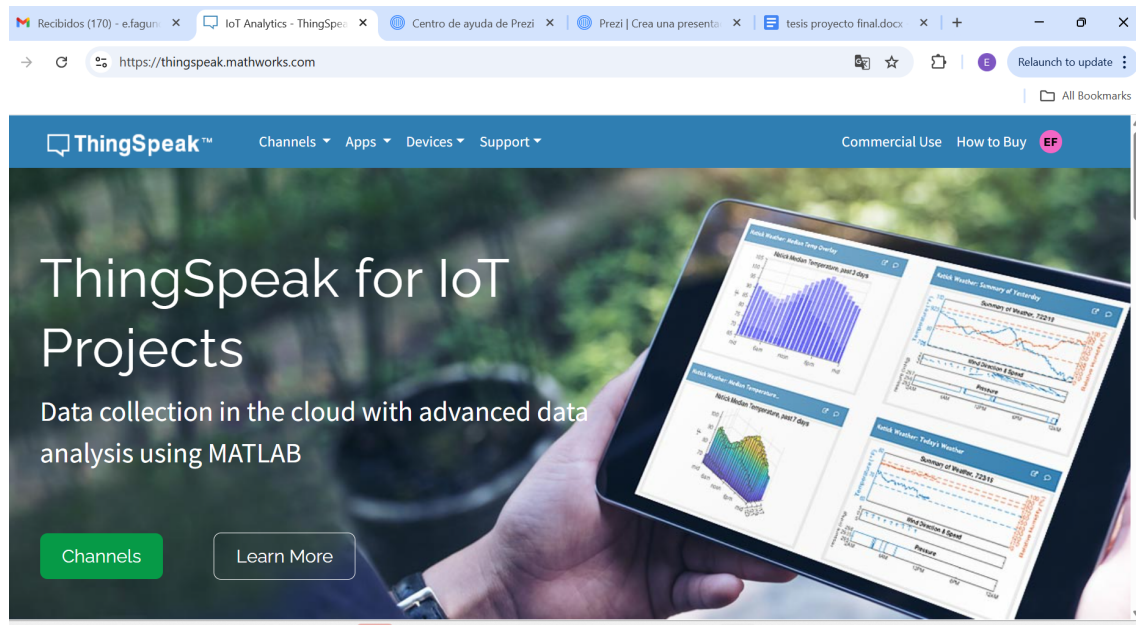


fig. 4.9 Pantalla presentación de la plataforma

Para comenzar se registró y generó un usuario en Thingspeak.

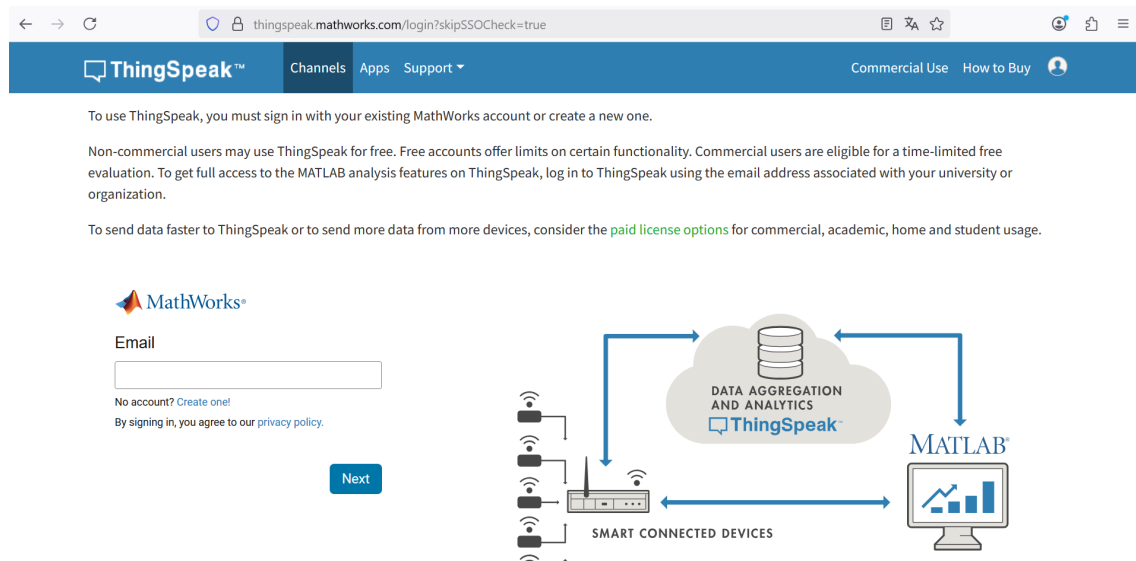


fig. 5.0 Pantalla registro de la plataforma

Una vez validado este usuario se ingresa al sistema en la pantalla se muestra la pantalla con el título “My channel”.

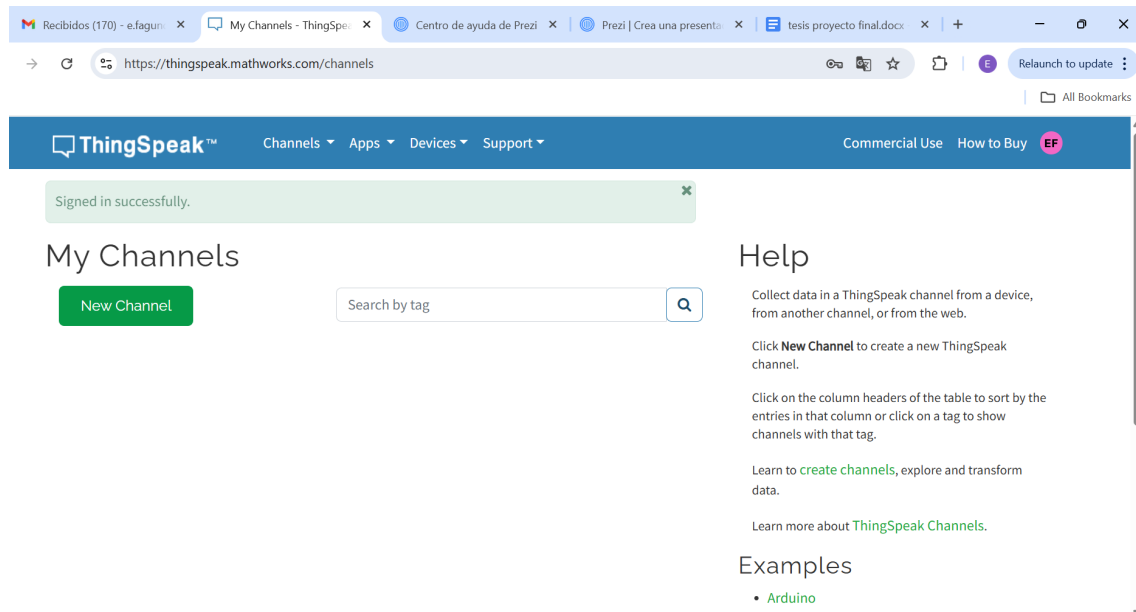


fig. 5.1 Pantalla channels de la plataforma

Seleccionamos new channel y dentro tipeamos el nombre, una breve descripción y las etiquetas en los puntos donde vamos a tomar mediciones o registros de nuestros sensores , se pueden colocar hasta 8 pagando, gratis permite 6 que es suficiente.

Una vez se cargó todo click en guardar.



Canales
Aplicaciones
Dispositivos
Apoyo

Usar comercial
Cómo comprar

Nuevo canal

Nombre

Nombre del canal

Descripción

descripcion de que muestra mediante el enlace el canal de que se trata el proyecto

Campo 1

punto de medicion 1

Campo 2

otro punto

Campo 3

un tercer punto

Campo 4

y así

Campo 5

sucesivamente

Campo 6

Field Label 6

Campo 7

Campo 8

Metadatos

Etiquetas

(Las etiquetas están separadas por comas)

Enlace a sitio externo

http://

Enlace a GitHub

https://github.com/

Elevación

Mostrar ubicación del canal

Latitud

0.0

Longitud

0.0

Mostrar vídeo

YouTube
Vimeo

URL del vídeo

http://

Mostrar estado

Guardar canal

Ayuda

Los canales almacenan todos los datos que recopila una aplicación ThingSpeak. Cada canal incluye ocho campos que pueden contener cualquier tipo de datos, además de tres campos para datos de ubicación y uno para datos de estado. Una vez recopilados los datos en un canal, se pueden usar las aplicaciones ThingSpeak para analizarlos y visualizarlos.

Configuración del canal

- Porcentaje completado:** Se calcula a partir de los datos ingresados en los distintos campos de un canal. Introduce el nombre, la descripción, la ubicación, la URL, el video y las etiquetas para completar tu canal.
- Nombre del canal:** ingrese un nombre único para el canal ThingSpeak.
- Descripción:** Ingrese una descripción del canal ThingSpeak.
- Campo n.º:** Marque la casilla para habilitar el campo e introduzca un nombre. Cada canal de ThingSpeak puede tener hasta 8 campos.
- Metadatos:** ingrese información sobre los datos del canal, incluidos datos JSON, XML o CSV.
- Etiquetas:** Introduce palabras clave que identifiquen el canal. Separa las etiquetas por comas.
- Enlace a sitio externo:** si tiene un sitio web que contiene información sobre su canal ThingSpeak, especifique la URL.
- Mostrar ubicación del canal:**
 - Latitud:** Especifique la latitud en grados decimales. Por ejemplo, la latitud de Londres es 51.5072.
 - Longitud:** Especifique la longitud en grados decimales. Por ejemplo, la longitud de Londres es -0,1275.
 - Elevación:** Especifique la posición de elevación en metros. Por ejemplo, la elevación de la ciudad de Londres es 35.052.
- URL del video:** si tiene un video de YouTube™ o Vimeo® que muestra la información de su canal, especifique la ruta completa de la URL del video.
- Enlace a GitHub:** si almacena su código ThingSpeak en GitHub®, especifique la URL del repositorio de GitHub.

Usando el canal

Puedes obtener datos en un canal desde un dispositivo, sitio web u otro canal de ThingsSpeak. Luego, puedes visualizarlos y transformarlos con [las aplicaciones de ThingSpeak](#).

Consulte ["Comience a usar ThingSpeak"](#) para ver un ejemplo de medición del punto de rocío desde una estación meteorológica que adquiere datos de un dispositivo Arduino®.

Más información

fig. 5.2 Pantalla configuración de la plataforma

Aquí ya se puede ver creado el canal que llamamos water control & monitoring, se seleccionó público, en esta misma ingresar en la pestaña API Keys.

70

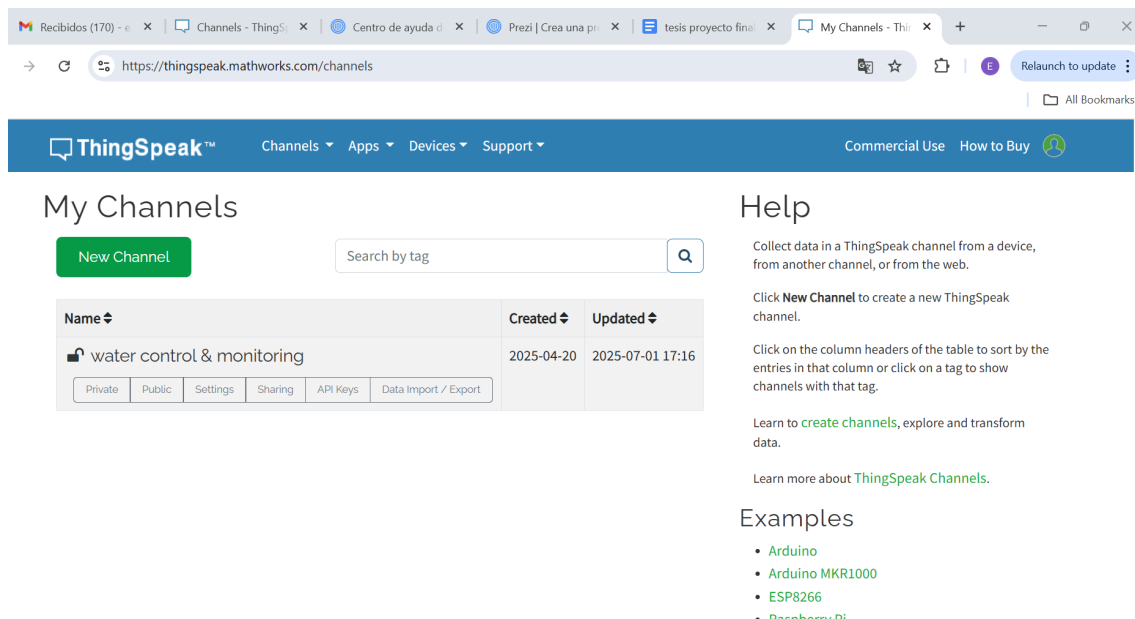


fig. 5.3 Pantalla channels con canal creado de la plataforma

Dentro de API key guardar el número de chanel que nos brinda el programa y el número Write API key que son necesarios para la conexión entre sensores y servidor.

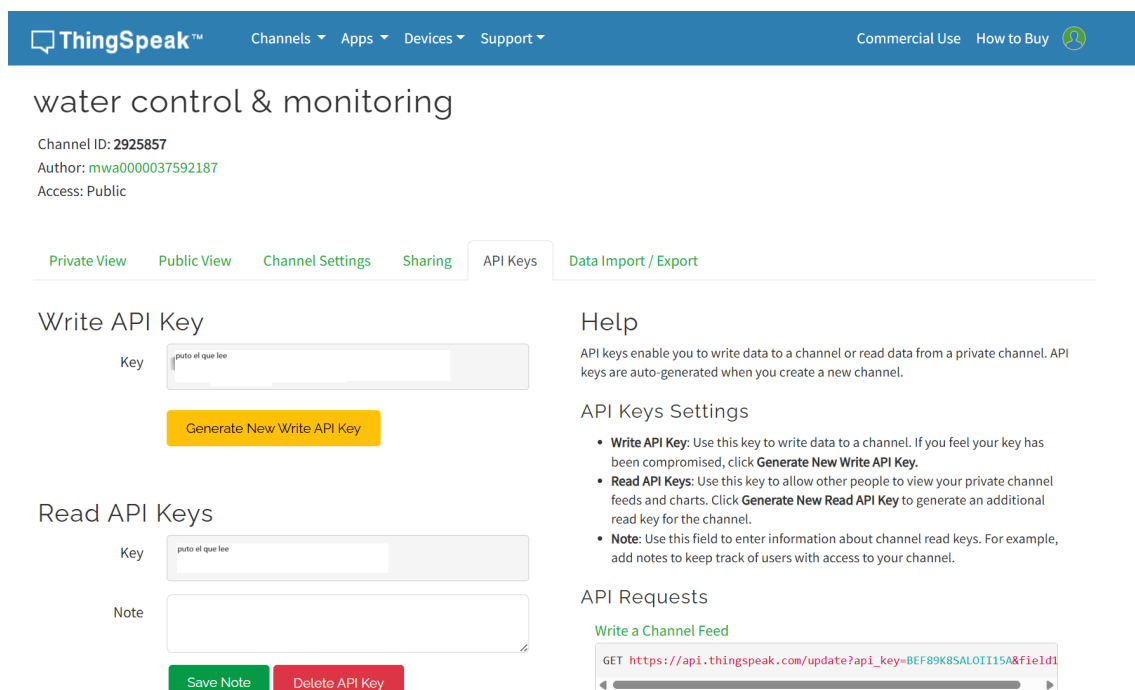


fig. 5.4 Pantalla con datos "write API Key" de la plataforma

Incluir en el código las librerías ThingSpeak y WiFi.h

```
#include "ThingSpeak.h"
#include "WiFi.h"
```

y para conectar a nuestra red anotamos nuestro ssid y password cada una en una variables del tipo const char que tenemos que declarar.

```
const char *ssid = "Aca va el nombre de red a la que quieres conectarte";
const char *password = "Aca va tu contraseña";
```

Declaramos dos variables más un del tipo unsigned Long para cargar el número del channelID y otra const char para cargar el write API key que guardamos anteriormente.

```
unsigned long channelID = "Aca va el numero del channelID";
const char *Write API key = "Número que guardamos de Write API key";
```

Para la librería WiFi.h creamos un cliente. WiFiClient permite comunicarse con otros dispositivos o servicios a través de Internet o una red local, como por ejemplo realizar peticiones HTTP, enviar datos a una API o conectarse a un servidor MQTT.

```
WiFiClient cliente;
```

Para conectarnos al wifi dentro del setup declaramos la función WiFi.begin en los argumentos cargar nombre de la red y contraseña (ssid,password).

```
void Setup ();
WiFi.begin(ssid,password);
```

Con un bucle while para con las variables retry Conn y retro Con Max se fijan 5 intentos de conexión y se muestra en el monitor serie o sigue ejecutando el programa.

```
Serial.begin(115200);
WiFi.begin(ssid, password); // ahora nos fijamos que no este conectado y
que no haya alcanzado el máximo de reintentos
while (WiFi.status() != WL_CONNECTED && retryConn <= retryConnMax)
{
    retry Con n++; // sumo 1 al contador de intentos
    delay(1000);
}
```

```

        Serial.print(".");
        Serial.println("CONNECTING");
    }

    if (WL_CONNECTED)
    {
        Serial.println("SUCCESSFUL CONNECTION");
        ThingSpeak.begin(cliente);
    }
}

```

En cada una de las funciones del void loop elegimos las variables que queremos mostrar y como argumento colocamos la posición del widget al que queremos visualizar y el nombre de la variable que vamos a mostrar.

```

ThingSpeak.setField(aca va la ubicación del cuadro, aca va el nombre de
la variable);

```

```

void CalculateWaterTankLevel()
{

    if (CurrentWaterLevel < 30)
    {
        WaterLevelStatus = 2;
    }
    else if (CurrentWaterLevel > 30 && CurrentWaterLevel < 80)
    {
        WaterLevelStatus = 1;
    }
    else if (CurrentWaterLevel > 80)
    {
        WaterLevelStatus = 0;
    }
    statusTank = WaterLevelStatus;
    if (WL_CONNECTED)
    {
        ThingSpeak.setField(1, statusTank);
    }
}

```

Anexo V: Presupuesto

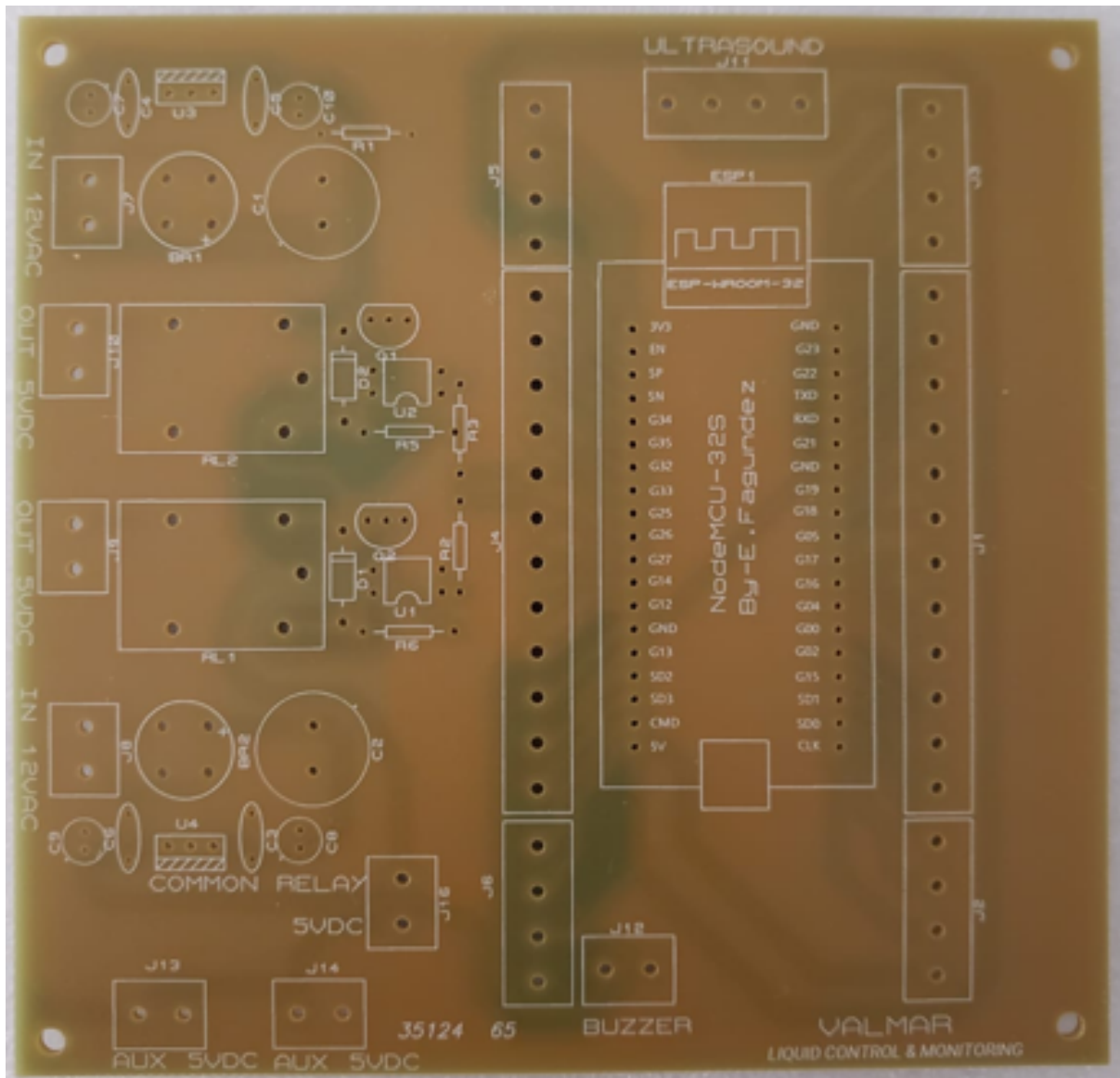
El coste material total de este monitor de líquidos fue de USD 183,6.

Materiales	# Utilizado	Catidad	Precio
NodeMCU esp-32s	1	1	USD 9,36
Sensor de flujo YF-S401	1	1	USD 5,89
Sensor de ultra sonido hc_sr04	1	1	USD 2,14
Display tft 3.5	1	1	USD 24,67
Transformador 220v 12v 3A	1	1	USD 24,68
Puente de diodos	2	2	USD 2,17
Capacitor 2200uF 50v	2	5	USD 3,55
Capacitor 1000uF 50v	2	10	USD 3,56
Capacitor poliester 0.47uF 650v	2	10	USD 5,25
Regulador LM7805	2	10	USD 4,60
Optoacoplador PC817	2	10	USD 4,68
Transistor 2N3904	2	5	USD 2,82
Relé SRD-05VDC-SL-C	2	5	USD 3,11
Diodo 1N4007	2	10	USD 2,58
Placa pertinax 15 x 15	1	1	USD 4,62
Resistor 100 ohm	6	30	USD 1,06
Resistor 470 ohm	1	30	USD 1,06
Resistor 1k	1	30	USD 1,06
Borneras	58	60	USD 10,60
Cables dupont macho hembra 40 x 30cm	1	1	USD 2,86
Contactador 16A	1	1	USD 11,30
Discipador	2	2	USD 2,13
Pasta termica	1	1	USD 2,12
Bomb 5V	2	2	USD 7
Bomba 220v	1	1	USD 12,62
Cooler 5v	1	1	USD 3,63
Diseño del pcb	1	1	USD 24,21

Anexo VI: Armado



fig. 5.5 Montaje del display



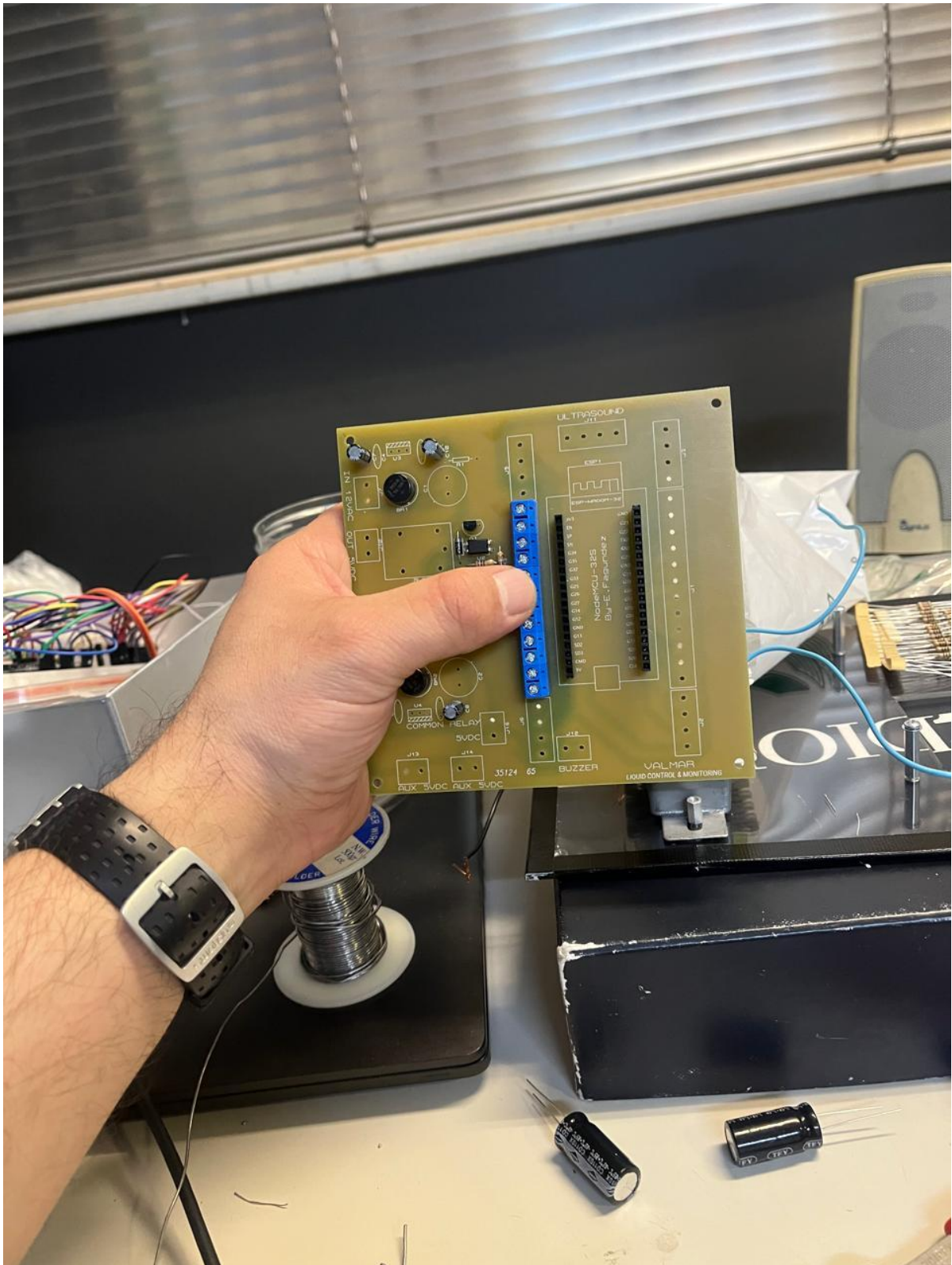


fig. 5.7 Colocación de conectores

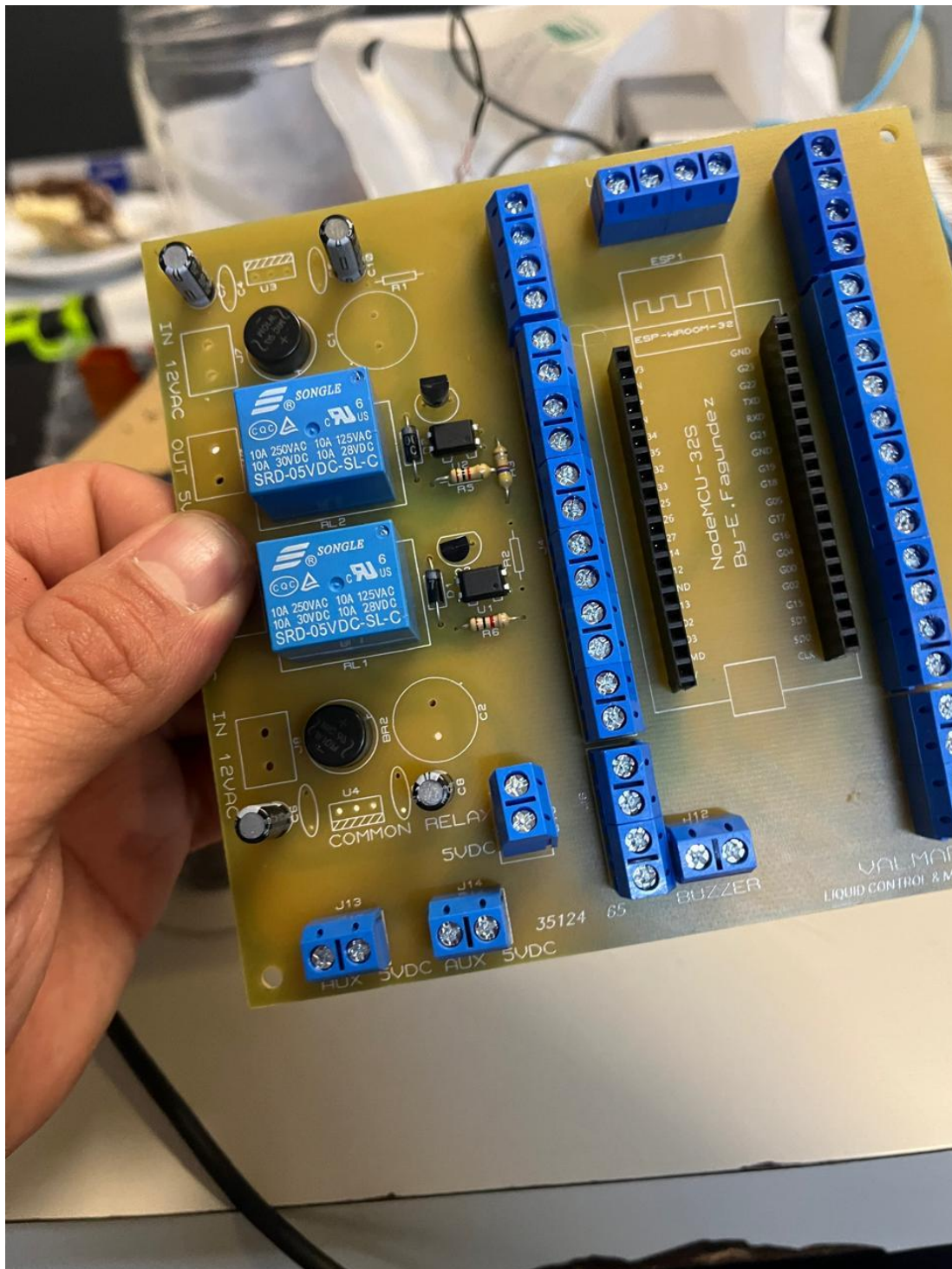


fig. 5.8 Colocación de relays



fig. 5.9 Colocación de tapa con display

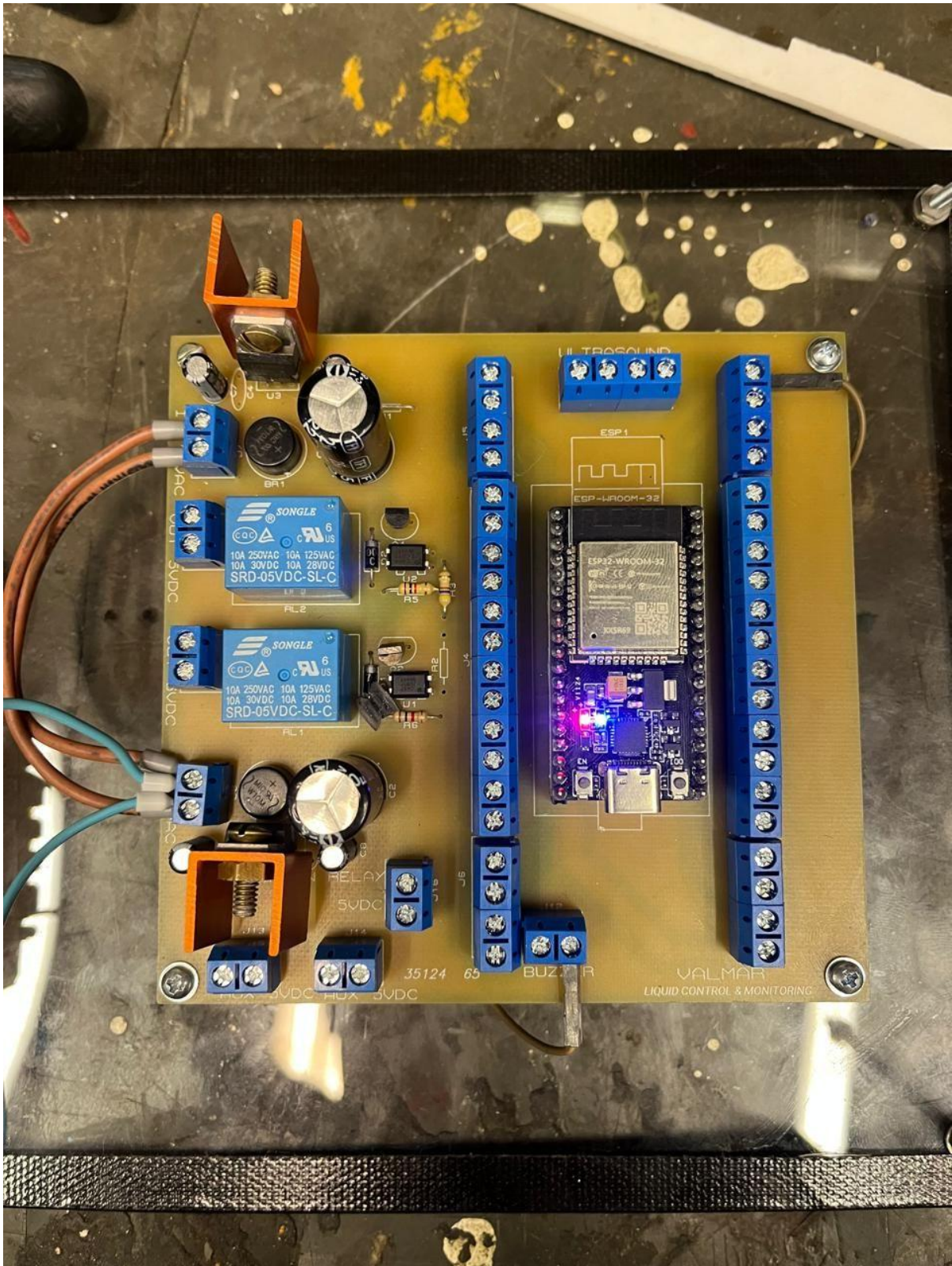


fig. 6.0 Visita superior con paneles montados

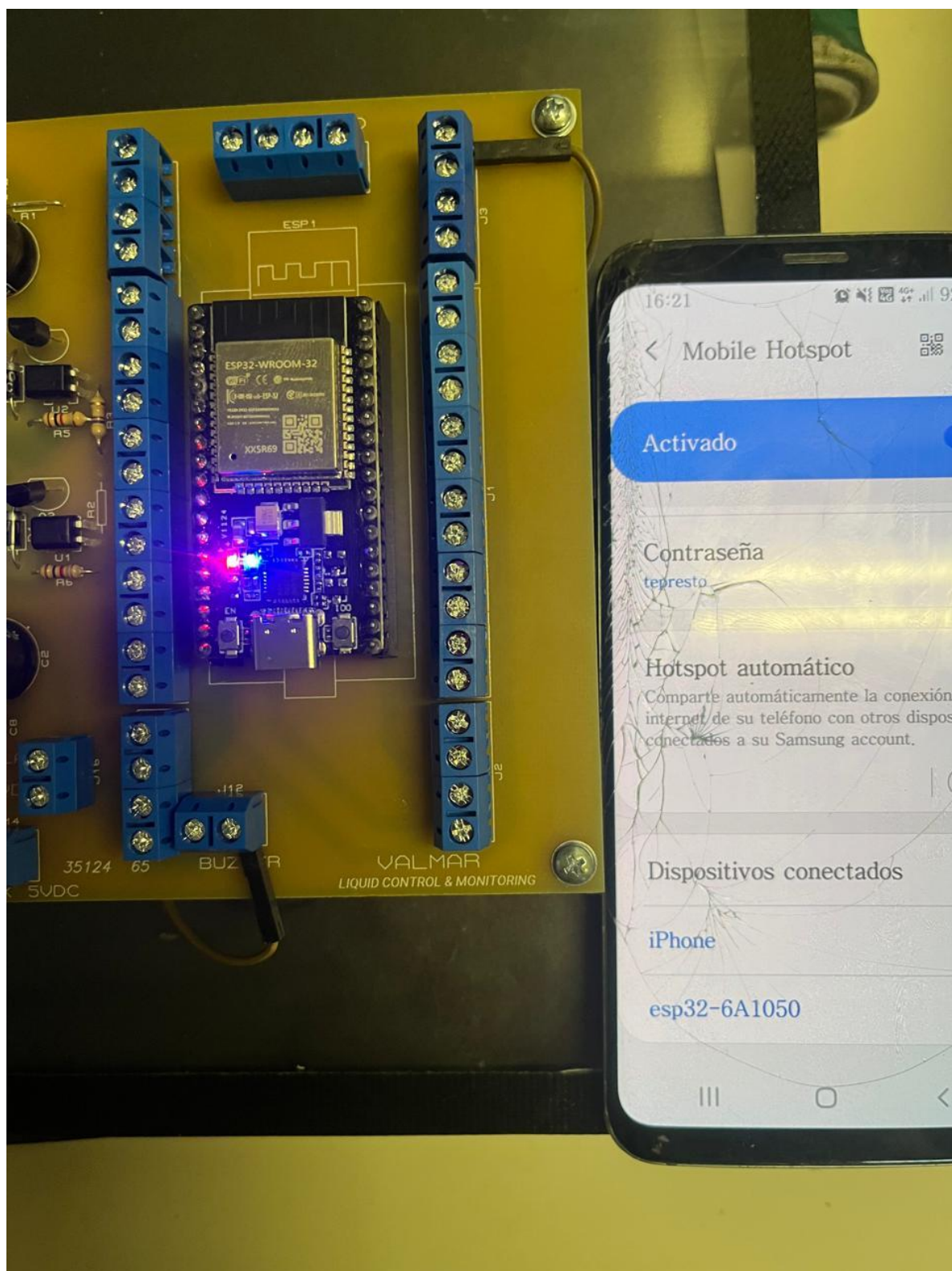


fig. 6.1 Conexión en la red wifi

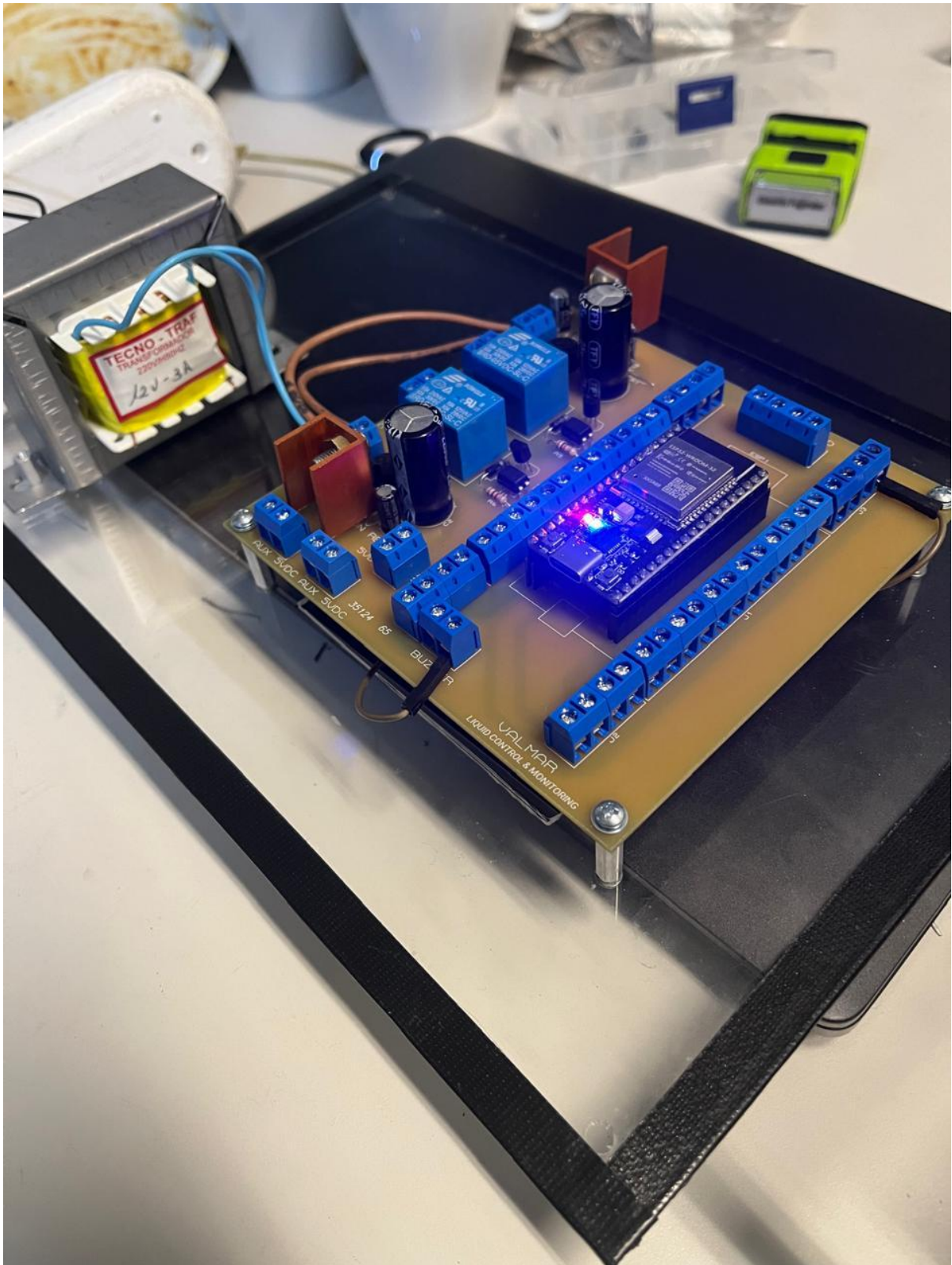


fig. 6.2 sistema operando sin sensores

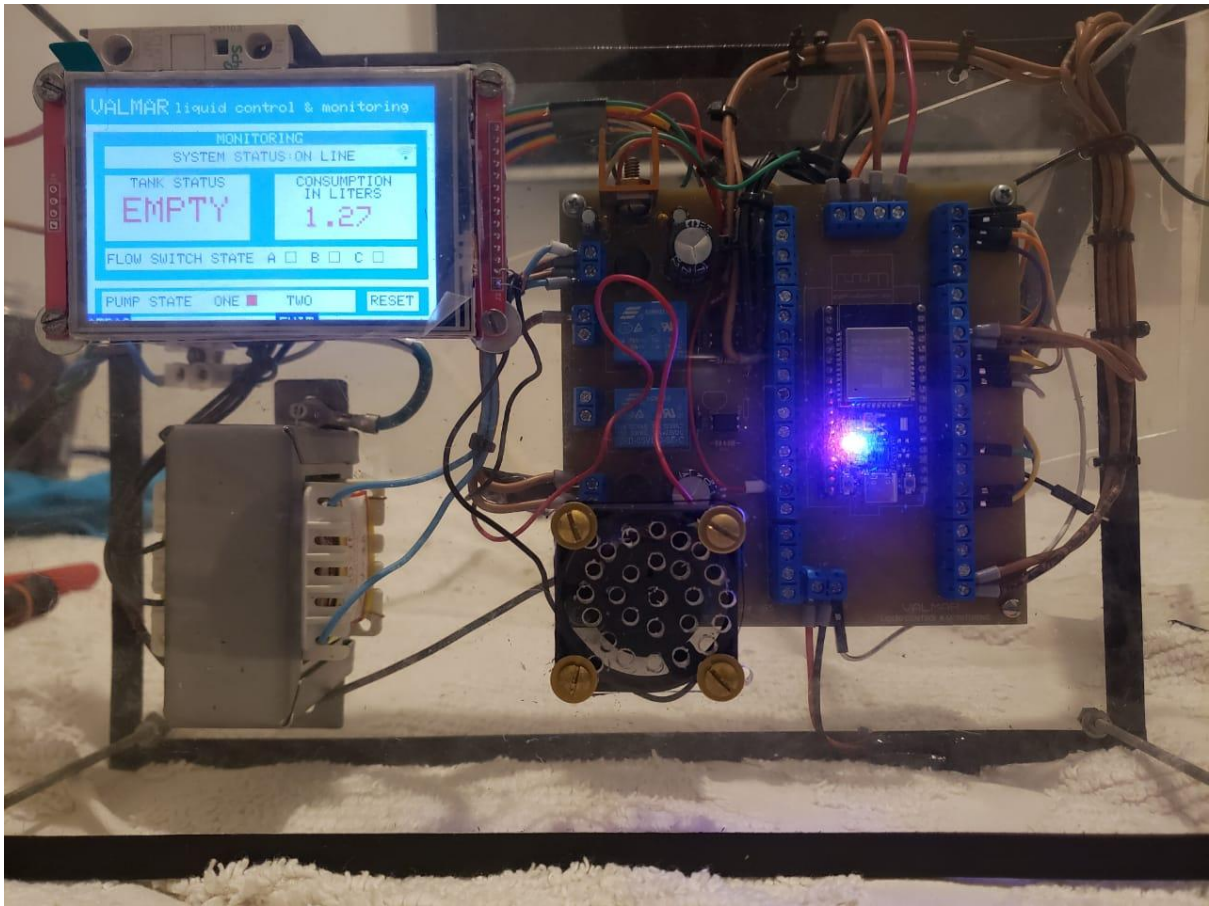


fig. 6.3 prototipo armado final con sensores y display