

Sistema de medición de nivel de agua

Asignatura:

Diseño de sistemas electrónicos

Presentado a:

Jaime Guillermo Barrero Perez

Por:

Jhonathan Alexander Murcia Galan

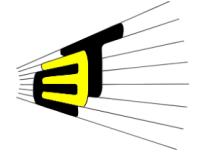
Código: 2191653

Angel Ricardo Castro Jaimes

Código: 2200501

Universidad Industrial de Santander

Bucaramanga



Resumen

Este informe presenta el diseño y desarrollo de un sistema de monitoreo de nivel para un tanque de agua subterránea utilizando el microcontrolador ESP32-C3. El objetivo de este proyecto es medir y transmitir de manera inalámbrica los niveles de agua en un tanque subterráneo de almacenamiento y enviarlo a un celular por medio de una conexión inalámbrica.

Introducción

El diseño de hardware es una de las áreas más robustas y trascendentales de la ingeniería electrónica, dada su creciente presencia en nuestra sociedad. Este informe se sumerge en el diseño y desarrollo de un sistema de monitoreo de nivel para un tanque de agua subterránea, utilizando el microcontrolador ESP32-C3. A medida que la ingeniería electrónica se vuelve cada vez más omnipresente, los ingenieros en formación enfrentan desafíos significativos al ingresar a este campo, ya que deben adquirir una amplia gama de habilidades para desarrollar productos competentes.

Estos desafíos incluyen la selección y distribución efectiva de componentes, la gestión eficiente de la potencia, el análisis y mitigación de elementos parásitos, y la optimización económica del producto final. La necesidad imperante de adquirir estas habilidades impulsa la solución al problema planteado en este informe. La propuesta se centra en el diseño de una PCB que incorpora un microcontrolador y varios periféricos, garantizando su correcto funcionamiento y permitiendo una posible expansión futura.

Objetivos

- El objetivo de este proyecto es proponer una solución energéticamente viable, expandible y económicamente sostenible.
- Adquirir habilidades en el desarrollo, planteamiento y diseño de sistemas electrónicos.

Marco Teórico

Sensores de Nivel de Agua:

Dispositivos diseñados para medir y monitorizar el nivel de agua en diversos entornos, proporcionando información crucial para aplicaciones como control de inundaciones, gestión de recursos hídricos y monitoreo ambiental.

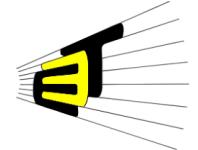
Microcontroladores y Sistemas Embebidos:

Microcontroladores son componentes electrónicos integrados en sistemas embebidos para realizar tareas específicas de control y procesamiento de datos. Estos sistemas autónomos encuentran aplicación en una variedad de dispositivos, desde electrodomésticos hasta sistemas industriales complejos.

Almacenamiento de Datos y Comunicación:

Conjunto de tecnologías y protocolos que facilitan la retención y transmisión de información. Incluye métodos de almacenamiento como memoria flash, así como tecnologías de comunicación como Wi-Fi, Bluetooth o Ethernet, fundamentales en sistemas embebidos para la interconexión y transferencia de datos.

Memoria SRAM y Flash:



Escuela de ingeniería Eléctrica, Electrónica y de Telecomunicaciones

Diseño de circuitos Electrónicos

Tipos de memoria utilizados en sistemas embebidos para almacenar datos temporalmente (SRAM) o de manera no volátil (Flash). La SRAM proporciona acceso rápido pero temporal, mientras que la Flash retiene la información incluso cuando se apaga la energía, siendo común en el almacenamiento de programas y datos.

Protecciones contra Descargas Electroestáticas (ESD):

Medidas implementadas para prevenir daños a componentes electrónicos debido a descargas electroestáticas. Estas protecciones incluyen dispositivos y técnicas diseñadas para disipar o redirigir la carga estática de manera segura, evitando posibles fallos en los sistemas.

Regulación de Tensión y Eficiencia Energética:

Proceso mediante el cual se mantiene constante la tensión de alimentación en un sistema electrónico, garantizando un suministro de energía estable y adecuado para el funcionamiento correcto de los componentes. La regulación de tensión es esencial para prevenir daños y asegurar la fiabilidad del sistema.

Osciladores de Cristal:

Componentes que generan señales de frecuencia estable basadas en la resonancia de un cristal. Son fundamentales en sistemas embebidos para sincronizar operaciones y proporcionar referencias temporales precisas.

Metodología

La metodología utilizada en este proyecto se centra en el diseño e implementación de un sistema de

monitoreo de nivel de agua subterránea que integra el microcontrolador ESP32-C3, el sensor de nivel de agua ultrasónico JSN-SR04T y otros componentes relacionados. El proceso se divide en las siguientes etapas:

● Diseño del Circuito:

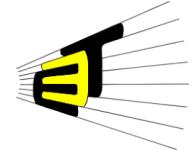
Se elaboró un circuito electrónico siguiendo las pautas de diseño proporcionadas por el fabricante del microcontrolador ([Ver Guía](#)). Este documento detalla los diversos componentes que deben ser seleccionados, junto con una descripción de consideraciones clave a tener en cuenta durante el proceso de selección. Además, se utilizó software de diseño asistido por computadora (CAD) en específico [easyEDA](#) y [Orcad](#) para diseñar el esquemático y el PCB de 4 capas, que incluye el microcontrolador, los componentes de memoria, el oscilador de cristal, las protecciones ESD, el conector USB-C, el regulador de tensión, botones boot/reset, manejo de señales en serie, adaptador o convertidor USB TO UART y la antena IFA.

Utilizando estas herramientas, respaldadas por una investigación sobre cada componente, se proponen los siguientes elementos para el diseño del circuito.

Nota: Cada elemento seleccionado tiene disponibilidad a la fecha en el stock de la tienda [LCSC ELECTRONICS](#), para que sea posible su implementación.

1. Microcontrolador.

Se propone un listado de opciones de microcontroladores en base a factores como: precio, consumo energético, tamaño, conectividad, documentación y si es un micro nuevo o lleva



mucho tiempo en el mercado. Con estos factores se proponen los siguientes microcontroladores: nRF52840, STM32L4+, AVR DA, PSoC 6, Seeeduino XIAO, ESP32-S2, ESP32-C3, RP2040; de los cuales se toma la decisión de trabajar con el ESP32-C3 ([Ver datasheet](#)) que cuenta con ligeras ventajas sobre los competidores mencionados:

Device	ESP32-C3	nRF52840	STM32L4	AVR DA	PSoC 6	Seeeduino XIAO
Precio	1.35	3.93	4.20	2,99	19,99	7,99
Consumo[mW]	100	1,8	0,090	0,0055	0,01	100
Velocidad	180 MHz	64 MHz	180 MHz	24 MHz	150 MHz	48 MHz
# Nucleos	1	1	1	1	1	1
Conectividad	Wi-Fi de 2.4 GHz y Bluetooth 5	Bluetooth 5.3 y protocols 2.4GHz	Bluetooth 5.0, Wi-Fi	NO	Bluetooth 5.0, Wi-Fi	No

Figura 1. Tabla Comparativa Microcontroladores

A partir de esto mediante su [guía de diseño](#) se procede a conectar el microcontrolador así:

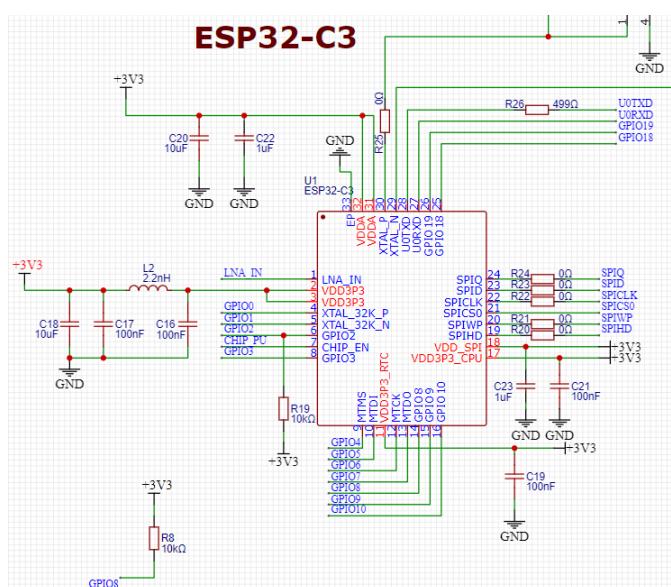


Figura 2. Esquemático ESP32-C3

1.1. Digital Power Supply

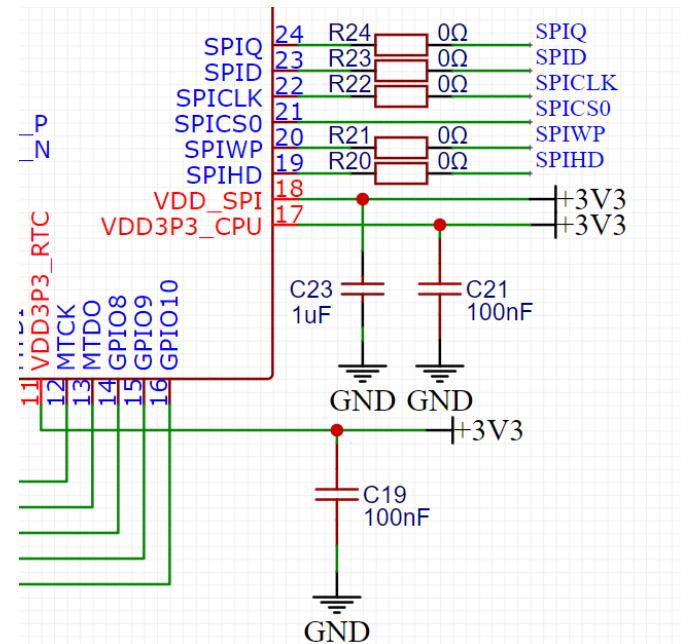
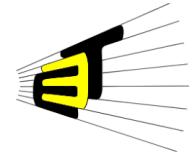


Figura 3. Conexiones de alimentación digital

Los pines 11 y 17 sirven como fuentes de alimentación para RTC IO y CPU IO, respectivamente. El fabricante recomienda adicionar un capacitor de 100nF cerca de estos dos pines.

Por otro lado, como se planea el uso de memorias externas (SRAM y FLASH) el fabricante recomienda un condensador de filtro de 1uF entre el pin 18 y tierra.



1.2. Analog Power Supply

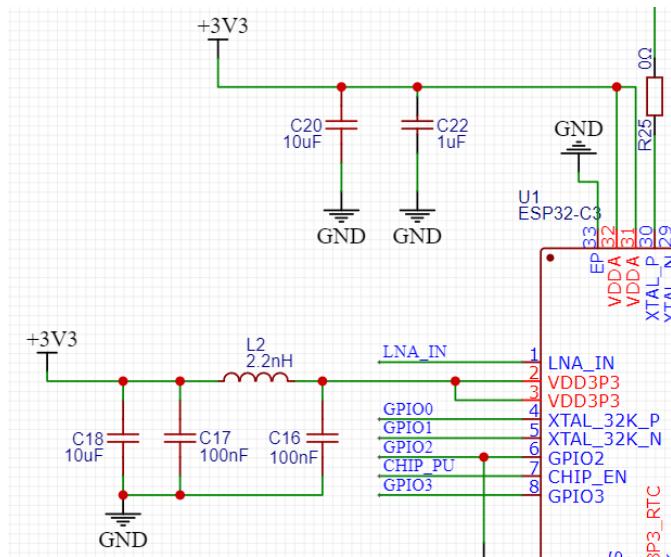


Figura 4. Esquemático ESP32-C3

El pin 2, el pin 3, el pin 31 y el pin 32 son los pines de la fuente de alimentación analógica y funcionan a 3,0 V ~ 3,6 V, estos son alimentados a través del regulador de voltaje de donde se obtiene una tensión de 3.3V. Por otra parte el fabricante pide para suprimir armónicos de alta frecuencia, un filtro LC cercano a los pines 2 y 3 teniendo en cuenta que se estima que la corriente que circula por la inductancia sea de alrededor de los 500 mA (factor que determina la bobina a usar). Además, se agrega en conjunto como se señala en la guía, dos capacitores de desacoplamiento de valores 1uF y 10uF cercanas a los pines 31 y 32.

1.3. Chip Enable

CHIP ENABLE

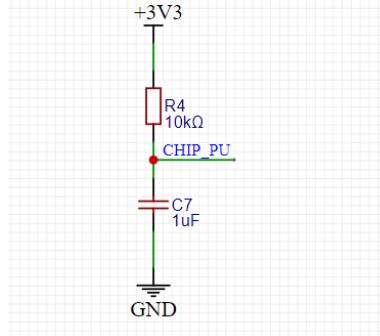


Figura 5. Esquemático Chip Enable

Una vez se cuente con una fuente de alimentación y se haya distribuido la energía, es esencial que, para la activación del microcontrolador, todos los rieles o pistas de alimentación alcancen una estabilización adecuada. Para lograr esto, se implementa un mecanismo mediante el pin 7, denominado CHIP_EN, que se encarga de activar el chip. Se introduce un circuito RC con valores típicos de resistencia ($R = 10k\Omega$) y capacitancia ($C = 1uF$) con el propósito de generar un delay calculado. Este delay asegura que el chip se encienda únicamente después de que los rieles de alimentación hayan alcanzado una estabilidad suficiente, garantizando un arranque adecuado del sistema.

1.4. Botones Reset y Boot

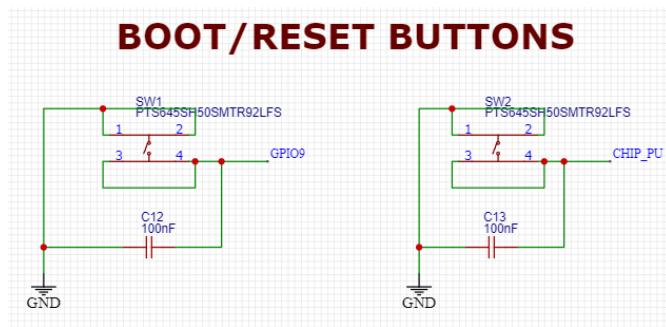


Figura 6. Esquemático Boot/Reset Buttons

Por otra parte el pin 7 CHIP_EN también puede ser utilizado como un reset, este componente proporciona la capacidad de reiniciar el microcontrolador, permitiendo la recuperación de errores y facilitando la depuración durante el desarrollo del software. En situaciones donde el sistema puede experimentar fallas o bloqueos, el botón de reset ofrece una solución rápida y efectiva para restablecer el microcontrolador a su estado inicial, contribuyendo así a la mejora de la integridad del sistema. Por esta razón, se decide implementar el botón de reset.

Además, el botón reset y el botón de boot, el cual es esencial para la versatilidad y adaptabilidad de un microcontrolador. Al iniciar el microcontrolador en un modo especial, este botón facilita la programación y la carga de firmware nuevo o actualizado. Su utilidad se extiende a la capacidad de poner el microcontrolador en un estado que permita la recuperación del sistema y la restauración de configuraciones predeterminadas. En situaciones donde se requiere la actualización o modificación del firmware, el botón de boot se convierte en una herramienta valiosa para la gestión eficiente y segura de estas operaciones

críticas; en conjunto estos botones ofrecen herramientas valiosas para el desarrollo, la depuración y el mantenimiento de sistemas basados en microcontroladores. Por esta razón, a través del pin 15 que funciona como GPIO9, ya que es el pin que recomienda usar el fabricante (Ver figura 7.), se decide hacer la implementación del botón de boot.

GPIO9 Input	Mode
Low/GND	ROM serial bootloader for esptool
High/VCC	Normal execution mode

Figura 7. Estados GPIO9

Ambos botones son conectados de la misma forma, siguiendo el diseño que ofrecen tarjetas de desarrollo como el [devkitc02-esp32-3c-v1.1](#).

El funcionamiento del microcontrolador se ve afectado dependiendo del estado los pines declarados como reset y boot (7 y 15, CHIP_EN y GPIO9 respectivamente), pero a su vez, este chip tiene una característica peculiar: el pin 14 (GPIO8) no puede presentar un estado LOW/GND a la vez que lo hace el GPIO9 ya que es una combinación inválida y desencadenará un comportamiento inesperado, en el modo de arranque normal (GPIO9 alto), se ignora GPIO8. Para dar solución a este problema, se conecta GPIO8 a alimentación la cual hace que el pin quede en estado HIGH.

Escuela de ingeniería Eléctrica, Electrónica y de Telecomunicaciones

Diseño de circuitos Electrónicos

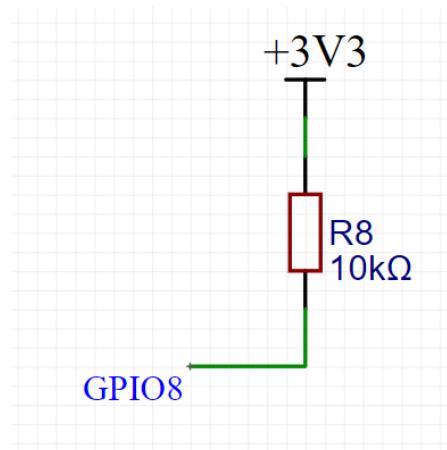


Figura 8. Esquema GPIO8

Para mirar el correcto funcionamiento de este proceso se implementa un LED controlado por señal [SK6812MINI](#) para que se apague cada vez que la señal GPIO8 este en LOW.

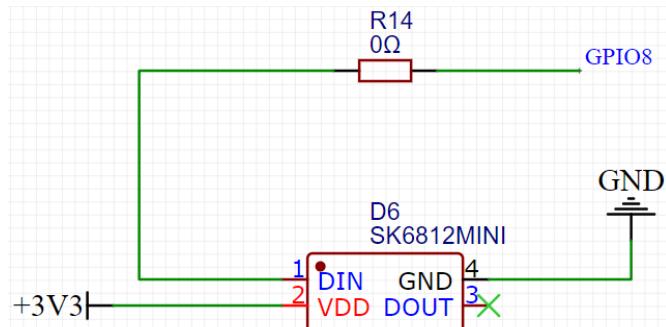


Figura 9. Esquema LED mini

Por otra parte, se busca que cuando se realiza la conexión del microcontrolador a través del puerto USB-C se pueda a realizar la programación del de este dispositivo, lo que se conoce como “[Automatic Bootloader](#)”, en consecuencia se desarrolló un conversor USB to UART o USB a SERIAL que permite esto. Posteriormente se detalla las conexiones del conversor, por ahora se tiene que este dispositivo cuenta con dos señales que harán

que el microcontrolador entre en “Automatic Bootloader”, estas señales que pueden estar en LOW o en HIGH son DTR (Data Terminal Ready) y RTS (Request To Send). Dependiendo de la combinación de los estados de estas señales el microcontrolador responde a la siguiente tabla de verdad:

Nota: Si se considera las señales EN, GPIO9 como booleanas, se debe tener en cuenta que para 0 = True y para 1 = False.

Estado	DTR	RTS	EN	GPIO9
Función Normal	1	1	1	1
Función Normal	0	0	1	1
Si está listo el firmware enable se activa y se reinicia el micro	1	0	0	1
Si RTS es HIGH osea si solicita permiso para enviar datos, el modo boot se activa	0	1	1	0

Figura 10. Tabla de funcionamiento del bootloader.

Para la implementación de la anterior tabla de verdad se desarrolla un circuito latch con [transistores bjt](#).

SERIAL SIGNALS HANDLING

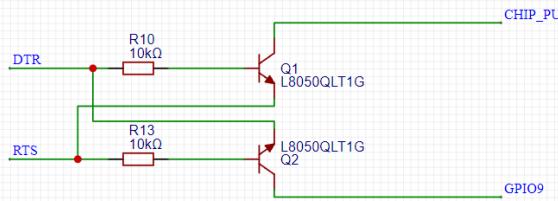


Figura 11. Esquema Latch BJT

2. Memoria FLASH

FLASH MEMORY

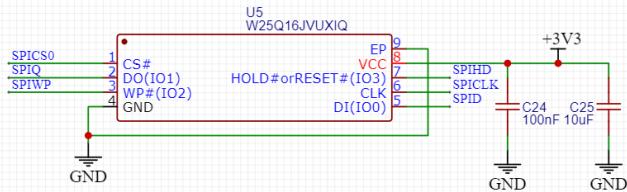


Figura 12. Esquema memoria Flash

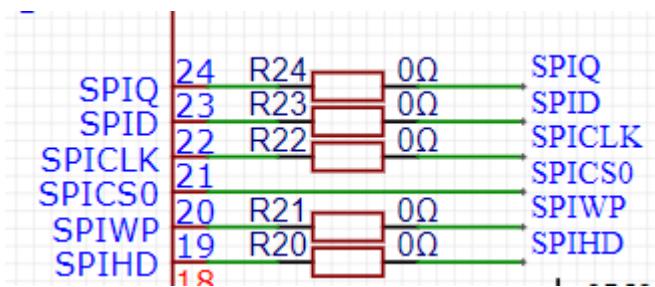


Figura 13. Conexiones Memoria Flash con ESP

La familia de microcontroladores ESP32 soportan hasta 16 MB de memoria flash externa, aunque las tarjetas de desarrollo implementadas por expressif tan solo cuentan con 4 MB; para este diseño se usó la memoria flash [W25Q128JVSIQ](#).

Esta memoria se conecta a través del protocolo de comunicación SPI a los pines 19, 20, 22, 23, 24 de la esp32-c3 por medio de una resistencia de 0 [Ω] para reducir la corriente de conducción, ajustar la sincronización, reducir la diafonía y la interferencia externa, conforme con lo indicado por el fabricante del microcontrolador. Por otro lado, se conectan dos capacitores de desacople para proporcionar una vía de baja impedancia para la corriente de alta frecuencia, eliminando ruido y manteniendo la estabilidad del voltaje cerca al pin 8 (VCC) donde se tiene la alimentación de la memoria.

A continuación se muestra una pequeña comparación con otras memorias, es de aclarar que todas funcionan con la misma tensión de alimentación:

Memoria	W25Q1 28JVSI M	W25Q128 JVSIQ	W25Q16 JVU XIQ
Capacidad [MB]	16	16	2
Consumo [uV]	1	1	1
Precio [US]	1.06	0.61	0.58

Figura 14. Tabla de comparativas de memorias Flash.

En la tabla se puede observar que la memoria seleccionada cuenta con una mayor capacidad de almacenamiento. Las tres opciones tienen un

consumo aproximadamente igual, y aunque el precio de la opción seleccionada no es el más bajo, es el único trade-off perceptible. No obstante, la diferencia de precio es mínima.

3. Memoria SRAM

SRAM MEMORY

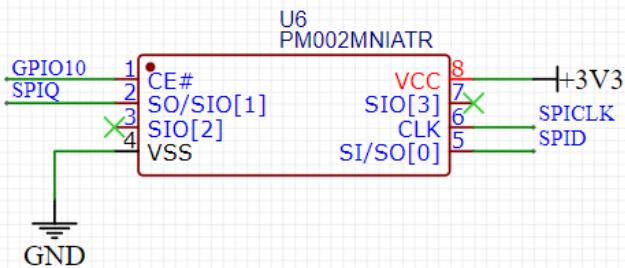


Figura 15. Esquema de memoria Sram

Aunque el microcontrolador cuenta con 400 kB de memoria SRAM interna, se opta por conectar una de estas memorias de manera externa mediante el protocolo SPI con el fin de proporcionar más espacio para datos y contribuir a mejorar el rendimiento del sistema al reducir la necesidad de acceder a la memoria principal interna del microcontrolador con frecuencia, lo que puede ralentizar la velocidad de ejecución.

Aunque módulos como [ESP32-C3-WROOM-02](#) y tarjetas de desarrollo como [esp32-c3-devkitc-02](#) no cuenten con memoria SRAM externa que permita realizar un punto de comparación, además, el fabricante Espressif no parece especificar explícitamente la referencia del modelo exacto o número de parte de la SRAM integrada dentro de sus módulos ESP32 en la hoja de datos pública o

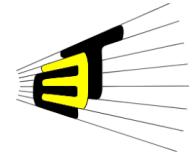
documentación accesible que se ha consultado. Sin embargo, basado en las características técnicas: Capacidad de 520KB, Velocidad de 80 MHz, Active Current: 80 mA, Standby Current: 10 mA, Sleep Current: 15-25 μ A, la memoria SRAM usada para este caso es [PM002MNIATR](#) ya que a pesar de trabajar a una frecuencia menor (50MHz) y contar con una capacidad de almacenamiento menor 250 KB se logra tener un consumo energético menor con las corrientes bajas que maneja, Active Current: 4.3 mA, Standby Current: 2 mA, Sleep Current: 2 μ A. Al no ser un programa que necesite realizar gran cantidad de cálculos al mismo tiempo, se opta por seleccionar una memoria SRAM externa que no consume mucha energía pero que esté ahí si el microcontrolador la necesita.

Al igual que la memoria Flash, la SRAM se conecta a los pines 19, 20, 22, 23, 24 de la esp32-c3 por medio de una resistencia de 0 $[\Omega]$, la única diferencia es el Chip Select de la memoria SRAM estará conectado el GPIO10 del microcontrolador.

4. Fuente de poder (Power Supply)

Según las especificaciones del fabricante del microcontrolador ([Ver Guía](#)), se requiere una alimentación de 3.3V. En este diseño, se obtendrá dicha alimentación a través de un conector USB-C que suministra un voltaje superior, el cual será regulado para obtener la tensión necesaria para alimentar el microcontrolador.

4.1. Conector USB-C



USB CONECTOR

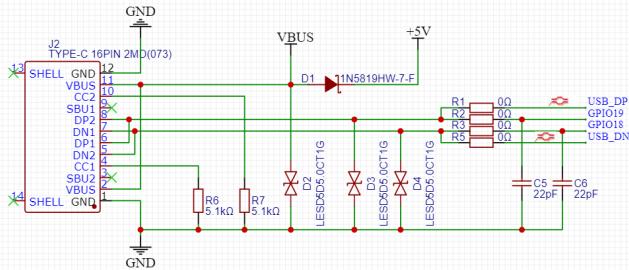


Figura 16. Esquema Conector USB

Además de ser una condición de diseño, se escoge un conector USB-C debido a que no solo cuenta con velocidades más rápidas que el tipo B, sino, permite cargar dispositivos a mayor potencia, hasta 100W, permitiendo carga rápida.

Para este diseño se opta por el conector TYPE-C de 16 pines que tiene un diseño de contactos dobles (2MD) que mejora la durabilidad y soporta más de 10,000 ciclos de inserción, cualidad importante para aplicaciones como la propuesta para el proyecto implementado.

Gracias a la guía de diseño de la tarjeta de desarrollo [ESP32-C3-DEVKITC-02](#) y al [datasheet](#) del conector se realizan las conexiones necesarias: DN y DP como par diferencial, los pines de canal de comunicación a tierra a través de un par de resistencias de $5.1\text{ k}\Omega$ ya que no nos interesa un control y comunicación continua entre el host y dispositivo a través del puerto. También se agrega sus respectivas protecciones ESD recomendadas por los fabricantes [LESD5D5.0CT1G](#).

Por último el conector USB-C no solo permitirá la alimentación, sino, también la programación del microcontrolador gracias al conversor USB TO UART.

4.2. Regulador de tension 5V a 3.3V

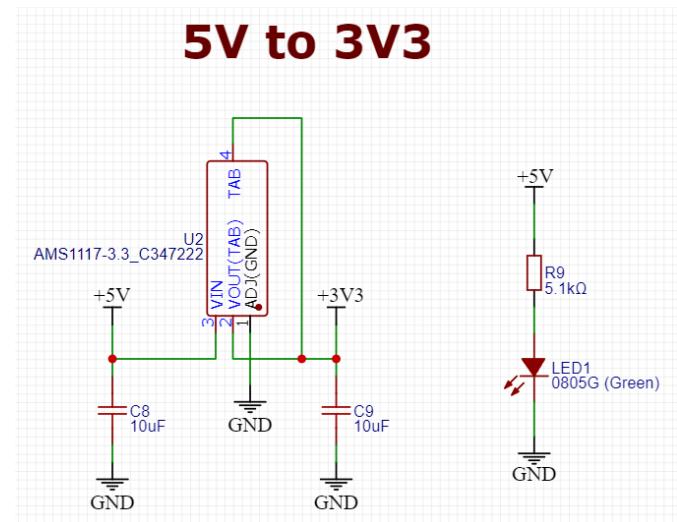


Figura 17. Esquema Regulador

Teniendo una entrada de +5 V mediante el regulador AMS1117-3.3 y mediante el modelo de conexión que recomienda el fabricante ([Ver datasheet](#)) se logra reducir a una tensión de 3.3V. Se añaden los capacitores de desacople de valor de 10 uF conforme indica el fabricante del regulador.

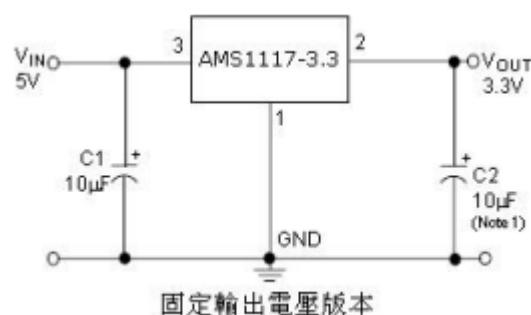
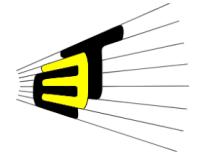


Figura 18. Guia para conexión Regulador

Este regulador maneja una tensión de entrada en el intervalo de 4.75V a 12V junto con una corriente de salida en el intervalo de 0 a 1A lo cual es ideal



debido a que, adicionalmente funciona en un rango de temperatura de -40 a 125 grados celsius.

Por otro lado, se agregó un diodo emisor de luz (LED) verde alimentado por los 5V que permite identificar que está en funcionamiento/encendida la PCB.

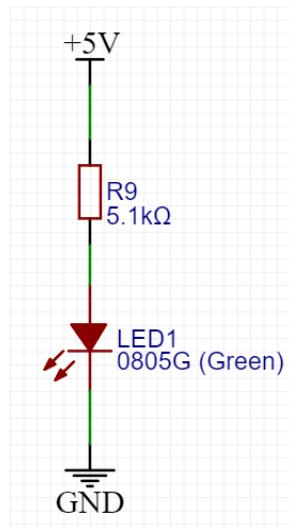


Figura 19. Esquema Diodo indicador.

5. Oscilador de cristal

En la actualidad, el firmware de la familia ESP32-C3 solo es compatible con cristales de 40 MHz. La capacitancia específica de se determinará mediante la hoja de datos del cristal seleccionado. El fabricante sugiere agregar un componente en serie en la traza de reloj XTAL_P para reducir la capacidad de conducción del cristal y minimizar el impacto de las armónicas del cristal en el rendimiento de RF. Es importante destacar que la

precisión del cristal seleccionado debe estar dentro de ± 10 ppm.

Se decide adquirir para el diseño el oscilador [R940000171](#) el cual cuenta con una frecuencia única de 40 MHz, por ello no necesita de una resistencia entre sus terminales para cuadrar el valor de frecuencia, y cumple con el resto de requerimientos planteados anteriormente, su fabricante pide que las capacitancias específicas sean de 10 pF.

CRYSTAL OSCILLATOR

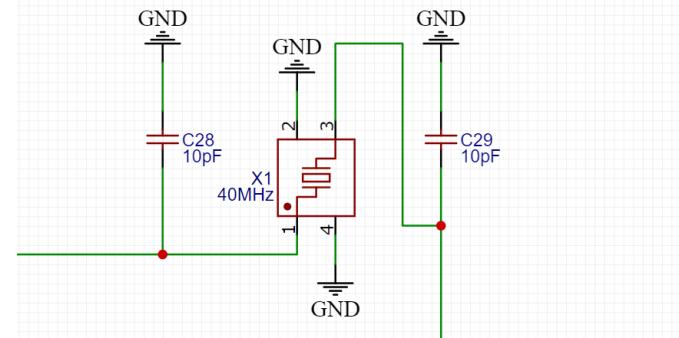
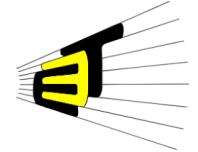


Figura 20. Oscilador de cristal



Figura 21. Resistencia Oscilador



6. Antena IFA

ANTENA 2.4GHz

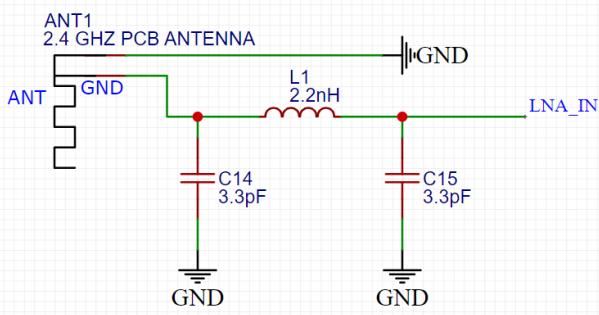


Figura 22. Esquema Antena IFA

Es la encargada de comunicar inalámbricamente el microcontrolador con el dispositivo que recibirá los datos. Su conexión empieza desde el Pin 1 de la ESP32-C3, pin capaz de transmitir y recibir señales; este se conecta a la antena diseñada para PCB, debido a que no se cuenta con CAD especializado para la simulación de diseños RF, se procede a utilizar un diseño para una antena de 2.4 GHz (frecuencia central para Wi-Fi y Bluetooth) realizado por la [Texas Instruments](#) en el cual se especifica la el tamaño, simulaciones de cómo se comporta la antena y el cómo se debe incluir en una PCB, por suerte este documento cuenta con un archivo DXF que permite importar el diseño a un CAD para su implementación.

Esta antena cuenta con una impedancia de entrada de 50Ω y el pin 1 del microcontrolador cuenta con una impedancia de $(35+j10)\Omega$, por esta diferencia de impedancias se requiere diseñar un circuito red de adaptación de impedancia el cual consta de un filtro LC en topología pi cuyos valores son

calculados gracias a la herramienta en linea [Pi-Matching Impedance Network Calculator](#)

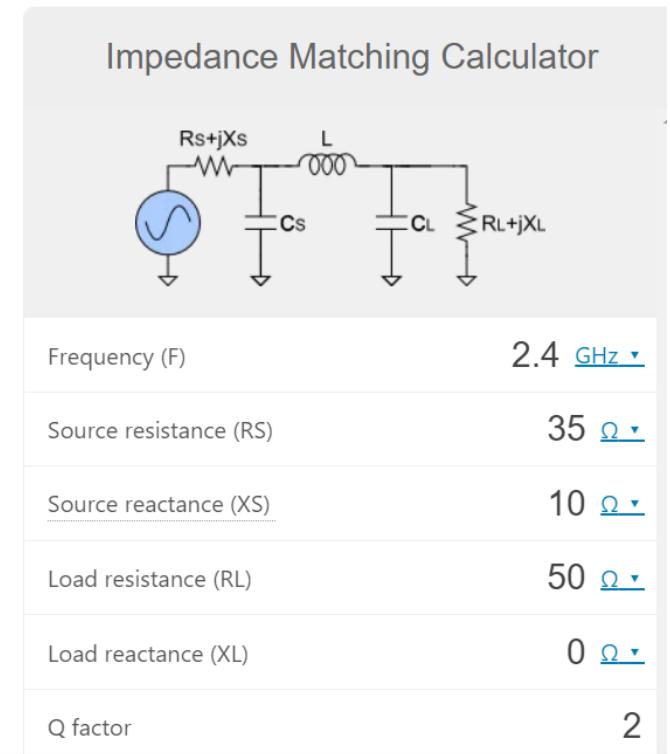
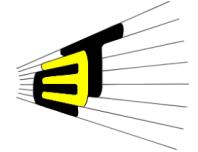


Figura 23. Calculadora Red de adaptación

Results	
Inductance (L)	2.433 nH
Source capacitance (CS)	3.424 pF
Load capacitance (CL)	2.6526 pF

Figura 24. Resultados de la red de adaptación.



Escuela de ingeniería Eléctrica, Electrónica y de Telecomunicaciones

Diseño de circuitos Electrónicos

7. USB to UART

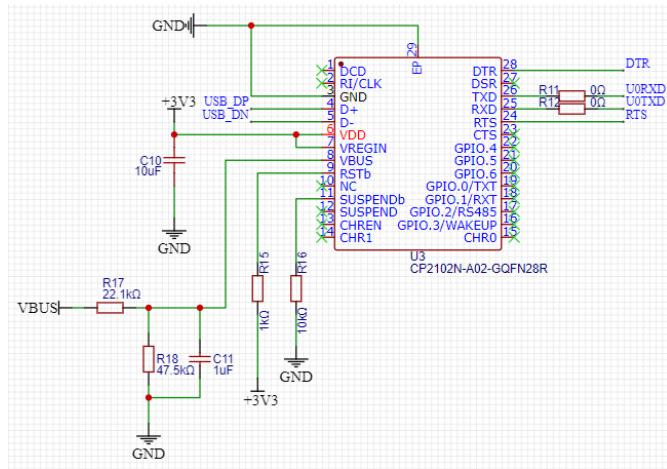


Figura 25. Esquemático USB TO UART

Como se había nombrado anteriormente se desea que no solo se alimenta energéticamente el microcontrolador a través del puerto USB-C, también se pretende realizar la programación del chip a través de este; para que esto sea posible es necesario la implementación de un chip USB to UART o USB a serial, la ventaja de este sistema es que permitirá la programación del microcontrolador por medio de un entorno conocido y amigable como lo es el entorno de desarrollo integrado (IDE) y el lenguaje de programación C/C++. Para el diseño se utilizó el USB TO UART [CP2102N-A02-GQFN28R](#), el cual cuenta con un reloj interno de 48 MHz (no necesita la implementación de uno externo).

Para su conexión se siguen las recomendaciones dadas por el fabricante en su hoja de datos, donde resaltan la importancia de la correcta conexión de la señal diferencial que llega desde el conector USB (D+, D-). Por otra parte, se sigue la indicación del uso de un divisor de tensión a partir de la

alimentación que suministra el mismo conector USB para la alimentación del chip USB to UART

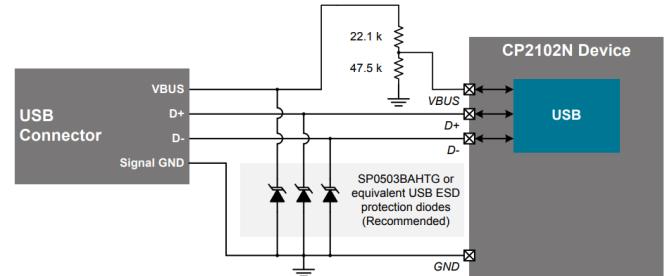


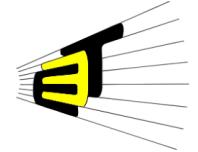
Figure 2.6. Self-Powered Connection Diagram for USB Pins

Figura 26. Esquema conexión dada por [fabricante](#)

Por otro lado la comunicación con la ESP32-C3 se divide en dos:

Control del modo Auto Bootloader
(Explicado en la sección del Chip Enable)

Transmisión y Recepción de datos: Para programar el firmware del microcontrolador, es esencial contar con las señales de transmisión y recepción de datos, las cuales se conectan a los pines 27 y 28 de la ESP32 (U0TXD y U0RXD). El fabricante recomienda la inclusión de una resistencia de $499\ \Omega$ con el propósito de mitigar los armónicos de 80 MHz.



Escuela de ingeniería Eléctrica, Electrónica y de Telecomunicaciones

Diseño de circuitos Electrónicos

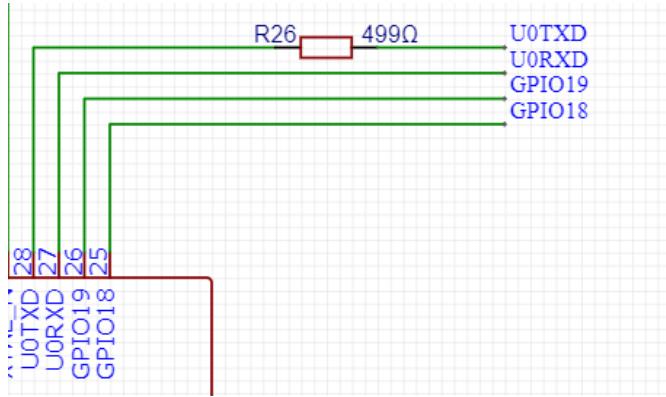


Figura 27. Resistencia Reducción Armonicos

● Diseño del PCB:

Para la realización del PCB a partir del esquemático realizado se probaron dos CAD distintos, Orcad y EasyEDA, por su parte Orcad permite de una manera mas facil la configuracion de los materiales, tamaños, grosor, tipo de material de las distintas capas, pero, la versatilidad, facilidad y comodidad que se siente con el software de easyEDA fueron parámetros determinantes para su elección.

Ya con software establecido, el fabricante del microcontrolador a través de su [guía de diseño](#), da claras indicaciones y recomendaciones sobre el posicionamiento de los componentes, el espesor y material de las capas, entre otros factores.

● **Generalidades:** Para que el diseño de la PCB pueda ser implementado, se necesita un fabricante confiable que imprima la tarjeta, para este caso se escoge a la empresa LCSC debido a que es el lugar donde se hizo la respectiva búsqueda de disponibilidad de componentes. En el diseño de PCB un factor muy importante

son las reglas, estas definen cosas como: ancho de pistas, diámetro de las vías, etc. Estas reglas se deben consultar con el fabricante para saber si el diseño que se está haciendo puede ser impreso por ellos. La empresa LCSC muestra las reglas que pueden imprimir en su [página web](#), con esta información se establece la siguiente regla para la PCB.

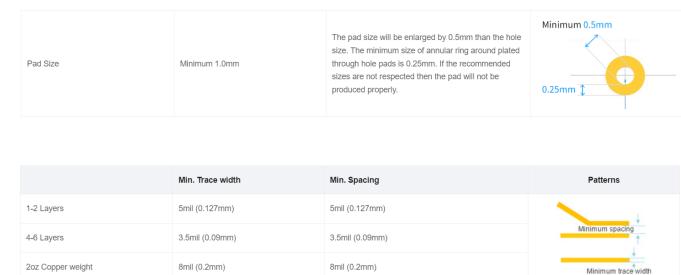


Figura 28. Reglas de diseño

Teniendo en cuenta estas condiciones se plantea la regla para el diseño:

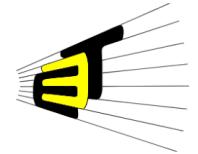
Rule	Track Width (min)	Clearance (min)	Via Diameter (min)	Via Drill Diameter (min)	Track Length (max)
Default	0.254	0.254	0.61	0.305	

Figura 29. Reglas de diseño aplicadas

● **Capas:** El fabricante del microcontrolador recomienda la implementación del PCB en 4 capas que se dividen de la siguiente manera:

La primera capa es la capa superior para componentes y rastros de señales.

La segunda capa es la capa GND sin enrutar rastros de señal para garantizar un plano GND completo.



Escuela de ingeniería Eléctrica, Electrónica y de Telecomunicaciones

Diseño de circuitos Electrónicos

La tercera capa es la capa de POWER donde se debe aplicar un plano GND para aislar mejor el módulo de RF y el cristal. Es aceptable enrutar trazas de señal en esta capa, siempre que haya un plano GND completo debajo del módulo de RF y el cristal.

La cuarta capa es la capa inferior, donde se enrutan las trazas de energía. No se recomienda colocar ningún componente sobre esta capa.

Por otra parte el fabricante da una indicación sobre el grosor de las capas, el material de estas y lo que debe ir entre ellas.

Thickness (mm)	Impedance (Ohm)	Gap (mil)	Width (mil)	Gap (mil)
-	50	12.2	12.6	12.2
Stack up	Material	Base copper (oz)	Thickness (mil)	DK
SM			0.4	4
L1 Top	Finshed Copper 1 oz	0.33	0.8	
PP	7628 TG150 RC50%		8	4.39
L2 Gnd		1	1.2	
Core	Core		Adjustable	4.43
L3 Power		1	1.2	
PP	7628 TG150 RC50%		8	4.39
L4 Bottom	Finshed Copper 1 oz	0.33	0.8	
SM			0.4	4

Figure 20: ESP32-C3 Family PCB Stack-up Design

Figura 30. Guia para diseño de capas

Configuración que se aplica a la PCB en el CAD usado.

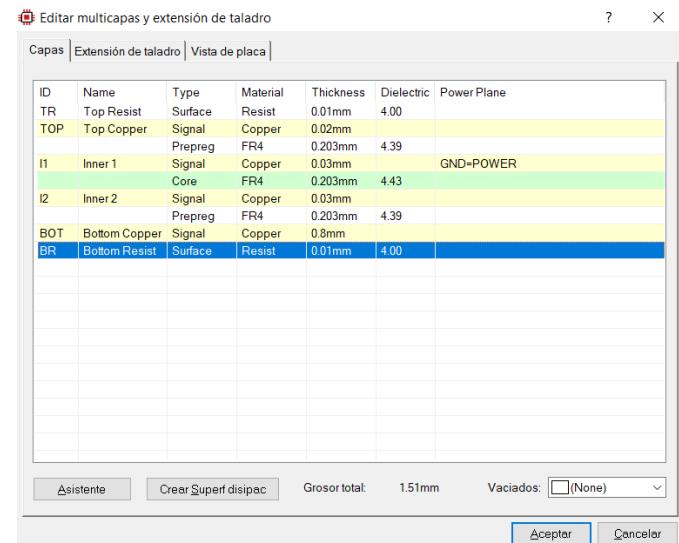
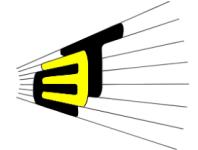


Figura 31. Capas de la PCB

Tras establecer la configuración de capas y las reglas de enruteamiento, la siguiente fase implica la disposición y conexión de los componentes correspondientes en la PCB. Se incorporan "headers" que posibilitan la expansión potencial del sistema, al dejar pines de entrada y salida del microcontrolador disponibles. Esto asegura un sistema flexible y preparado para la integración de múltiples sensores, además del sensor utilizado para medir el nivel del agua.



Escuela de ingeniería Eléctrica, Electrónica y de Telecomunicaciones

Diseño de circuitos Electrónicos

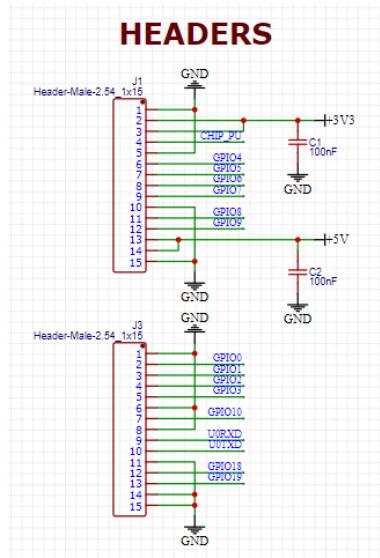


Figura 32. Esquemático Headers

El diseño final se presenta a continuación:

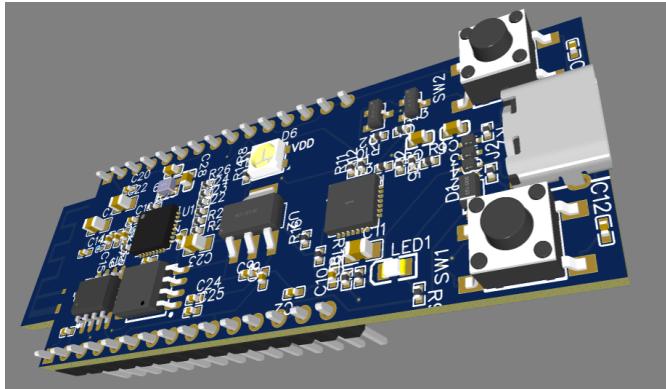


Figura 33. Modelo 3D I

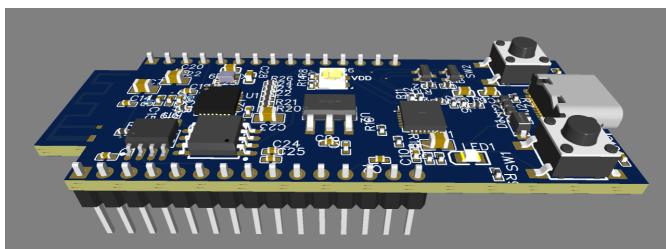


Figura 34. Modelo 3D II

SENSORES

La elección del sensor es probablemente uno de los procesos para los cuales se debe tener mayor rigurosidad. Es crucial tener en cuenta las diferentes opciones y los compromisos (trade-offs) que puedan surgir con cada sensor, dependiendo de la aplicación específica.

Ultrasónico	Contacto	Presión
Bajo costo	Costo medio	Alto costo
3 US	44 US	-

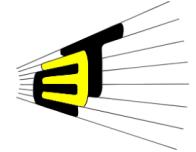
Figura 35. Tabla de sensores

Después de analizar las tres opciones consideramos que un sensor de presión sería la más costosa y en esta aplicación no es necesaria la precisión que puede proporcionar este, por lo que analizamos el caso de un sensor ultrasónico y debido a la aplicación encontramos que puede tener problemas al ser un tanque sellado. Por lo que la mejor opción fue un sensor de contacto, sin embargo si el tanque no estuviese sellado un sensor ultrasónico debería ser la mejor opción.

Made In China



Figura 36. Sensor de contacto RS485



Escuela de ingeniería Eléctrica, Electrónica y de Telecomunicaciones

Diseño de circuitos Electrónicos



Figura 37. Sensor ultrasónico JSN-SR04T

Programación del Microcontrolador:

Se programó el microcontrolador ESP 32-C3 utilizando un entorno de desarrollo integrado (IDE) y el lenguaje de programación C/C++. El firmware se diseñó para controlar la operación del sensor JSN-SR04T inicialmente, gestionar la comunicación Wi-Fi y almacenar los datos en la memoria flash.

Simulación de firmware:

Se llevaron a cabo pruebas para verificar el funcionamiento del firmware. Al no tener impresa la PCB se optó por usar una tarjeta de desarrollo con microcontrolador ESP32 el cual a través del IDE arduino se programó para recibir una señal analógica que simulaba el nivel del agua a través de la variación de un potenciómetro. ([Código usado](#))

Valor analógico:	Voltaje (V):	Grados:
1068	0.86	73.03
1060	0.85	72.48
1063	0.86	73.88
1075	0.87	73.50
1058	0.85	72.34
1067	0.86	72.96
1053	0.85	73.11
1069	0.86	73.09
1072	0.86	73.30
1058	0.85	72.34

Figura 38. Ventana de datos

Para mostrar los resultados en un móvil a través de una conexión Bluetooth se desarrolló un aplicativo móvil a través de la aplicación MIT App Inventor. ([descargar APK](#)).

Nota: El paradigma implementado para el código permite una posible expansión de recepción de más sensores que puedan ser implementados.

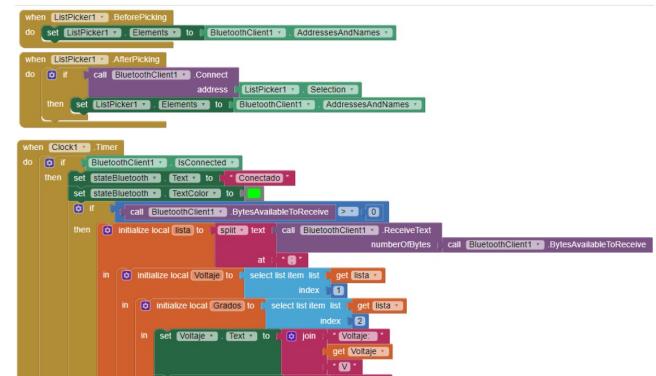
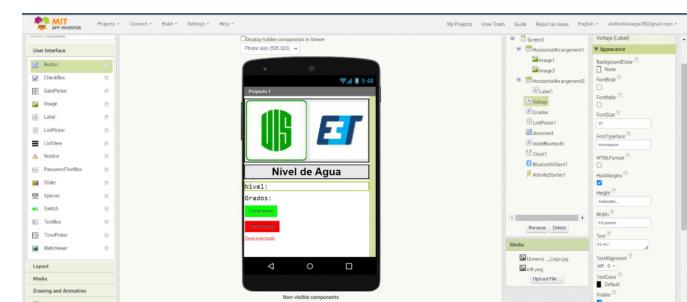
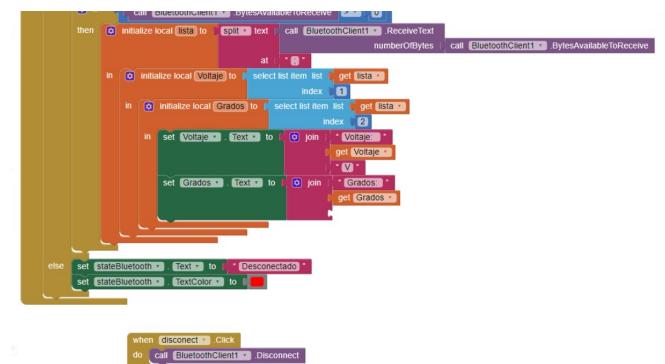


Figura 39. Programa de APP INVENTOR



Escuela de ingeniería Eléctrica, Electrónica y de Telecomunicaciones

Diseño de circuitos Electrónicos

Figura 40. Vista gráfica de la APP

El envío de datos se completó satisfactoriamente.

Presupuesto

Después de realizar el diseño de la pcb es necesario calcular el costo total de todos los elementos que llevaría el sistema:

ELEMENTO	PRECIO US
Elementos PCB	7.09
PCB UNITARIA	4.55
SENSOR	44-3
Total	55.5-15

Figura 41.Tabla de presupuesto

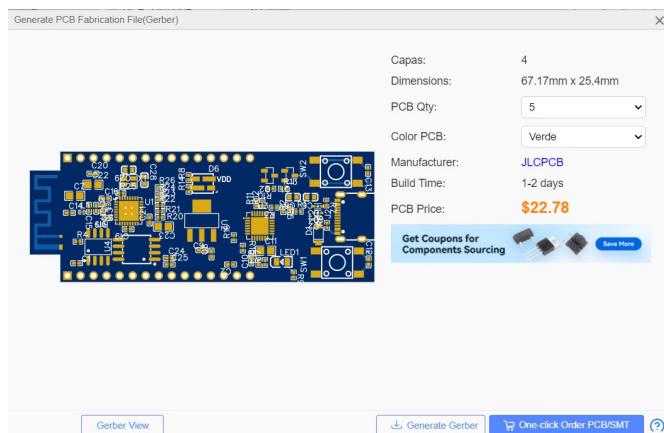


Figura 42.Costo Impresión PCB

Conclusiones

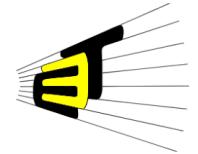
- El diseño de sistemas electrónicos es una de las ramas más complejas de la electrónica

debido a que este necesita de la correcta aplicación de diferentes conceptos para desarrollar correctamente un producto que pueda cumplir con las expectativas del mercado .

- Dependiendo de la aplicación, la elección del sensor puede volverse más compleja debido a las diferentes condiciones del problema.
- Los microcontroladores modernos proporcionan una gran cantidad de posibilidades al momento de diseñar sistemas electrónicos debido a su alta eficiencia y bajo costo
- Desarrollar un dispositivo que compita con las soluciones ya propuestas por diferentes empresas en el mercado requiere de un planteamiento bien estructurado

Referencias

- [1] ESP32C3 Family Hardware Design Guidelines. Available at: https://www.espressif.com/sites/default/files/documentation/esp32-c3_hardware_design_guidelines_en.pdf
- [2] Diseño de placas de circuito impreso utilizando Proteus Available at: <https://www.hubor-proteus.com/proteus-pcb/ares.html>
- [3] Texas Instruments Application note AN043 Available at: <https://www.ti.com/lit/an/swra117d/swra117d.pdf>
- [4] ESP32-C3 Series Datasheet. Available at: https://www.espressif.com/sites/default/files/documentation/esp32c3_datasheet_en.pdf



Escuela de ingeniería Eléctrica, Electrónica y de Telecomunicaciones

Diseño de circuitos Electrónicos

[entation/esp32-c3_datasheet_en.pdf#page=19&zoom=100,76,354](#)

[5] ESP32 + PCB Antenna Hardware Design
Tutorial - Phil's Lab #90:
https://www.youtube.com/watch?v=yxU_Kw2de08

[6] ESP32-C3 Specifications:
<https://c3.espressif.com/>