

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

**Desarrollo de tarjetas de comunicación para conexión a buses de
campo seriales**

**Diseño e implementación de un prototipo de tarjeta de
comunicación a una red modbus rtu para un sensor analógico de
0-10v**

**TRABAJO DE INTEGRACIÓN CURRICULAR PRESENTADO COMO
REQUISITO PARA LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN
ELECTRÓNICA Y AUTOMATIZACIÓN**

RENATO ANDRÉS VERA VILLAMIL

renato.vera@epn.edu.ec

DIRECTORA: Ph.D, SILVANA DEL PILAR GAMBOA BENITEZ

silvana.gamboa@epn.edu.ec

DMQ, agosto 2024

CERTIFICACIONES

Yo, Renato Andrés Vera Villamil declaro que el trabajo de integración curricular aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

Renato Andrés Vera Villamil

Certifico que el presente trabajo de integración curricular fue desarrollado por Renato Andrés Vera Villamil, bajo mi supervisión.

Ph.D, Silvana Del Pilar Gamboa Benítez
DIRECTORA

DECLARACIÓN DE AUTORÍA

A través de la presente declaración, afirmamos que el trabajo de integración curricular aquí descrito, así como el (los) producto(s) resultante(s) del mismo, son públicos y estarán a disposición de la comunidad a través del repositorio institucional de la Escuela Politécnica Nacional; sin embargo, la titularidad de los derechos patrimoniales nos corresponde a los autores que hemos contribuido en el desarrollo del presente trabajo; observando para el efecto las disposiciones establecidas por el órgano competente en propiedad intelectual, la normativa interna y demás normas.

Renato Andrés Vera Villamil

Ph.D, Silvana Del Pilar Gamboa Benítez

DEDICATORIA

El presente trabajo lo dedico a mis padres, los pilares fundamentales en mi vida, quienes día a día me enseñaron el valor de la perseverancia, la integridad y el buen accionar formándome en valores para ser siempre una buena persona antes que nada. Se lo dedico a mi hermano, quien me ha tomado como ejemplo en algunos momentos de su vida, siendo precisamente esa responsabilidad uno de los motores que me impulsó durante mi vida universitaria. A mi hermana, por todas las tardes y noches que me acompañó mientras armaba las prácticas de laboratorio, y que a pesar de su desconocimiento, siempre mostró un gran interés en mis actividades. Por último, se lo dedico a mi amada compañera, Samantha Felicita, quien me apoyó incondicionalmente durante todo este proceso, siendo luz guía en momentos de estancamiento mental, adquiriendo destrezas y conocimientos para aportar ideas en este trabajo, y aun con sus exigentes actividades, siempre supo darse tiempo para motivarme y darme su apoyo y cariño incondicional.

Este trabajo les pertenece de forma inalienable, ya que no sólo es el reflejo de mis esfuerzos y dedicación, sino también un tributo a sus contribuciones invaluable en mi vida, sin las cuales yo considero, hubiera sido muy difícil alcanzar este logro.

AGRADECIMIENTO

Con mucho cariño agradezco:

En primer lugar, a Dios porque solo él, en su infinita sabiduría, conoce los planes que me tiene preparado para haberme permitido llegar a este punto de mi formación profesional y personal y porque escuchó las peticiones, en momentos difíciles, y agradecimientos en momentos de triunfo de mi mamá.

A mi familia, mis padres José y Mariuxi quienes, a su manera, me brindaron todo su apoyo durante toda esta etapa, a mi papá particularmente por todas las madrugadas en las que se levantaba para salirme a dejar precautelando mi seguridad y por su interés y asombro conforme iba creciendo académicamente. A mi madre, que es mi principal motor de superación y ejemplo de perseverancia e integridad, por todas las madrugadas en las que me acompañó y renunciando a su sueño y descanso jamás permitió que saliera con el estómago vacío. A mis hermanos, por su paciencia y porque siempre estuvieron prestos a escucharme y porque fueron ustedes quienes me enseñaron el sentido de responsabilidad.

A la Lic. Samantha Felicita quien, como se dice coloquialmente, estuvo al pie del cañón viviendo conmigo las últimas etapas de este momento y supo darme su apoyo incondicional, impulsándome con sus consejos e ideas, ideas que se encuentran inmersas en este trabajo, y sobre todo por seguir a mi lado cuando la situación no parecía favorable.

A mis amigos más cercanos David Rodríguez, Alejandro Barros, Carlos Flores, Paúl Villagrán, Kevin Santillán y la Ing. Leydy Molina, con quienes compartí vivencias únicas y por quienes reafirmé que uno puede llegar más lejos cuando tiene la compañía correcta.

A mis compañeros Johao S., Marco S., Daniela N., Gabriela C., y Carla Y., por su complicidad en cada instancia académica y no académica de la vida universitaria, y a la Srta. Alisson Calero, quien, en su momento, fue un pilar de apoyo significativo en una situación adversa.

A mi directora la PhD. Silvana Gamboa, por su tiempo y por compartir sus conocimientos para hacer posible este documento, y de forma especial al Ing. Mateo Vásquez quien más que un excelente docente fue un gran amigo, por su tutela y guía en asuntos académicos, profesionales y personales.

A la empresa Solintec, en particular al Ing. Robert Guanoliquin y a su esposa Dora María quienes me dieron mi primera aproximación al mundo laboral, a todo el personal en general por su interés sobre el avance de este proyecto, y a Christian Moreno y Alexander Tierra por la ayuda que me supieron brindar, misma que fue crucial para llevar a término este trabajo.

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO.....	V
RESUMEN.....	VII
ABSTRACT.....	VIII
1 INTRODUCCIÓN	1
1.1 Objetivo general	2
1.2 Objetivos específicos	2
1.3 Alcance	2
A. Fase Teórica	2
B. Fase de diseño	3
C. Fase de implementación.....	3
D. Fase de pruebas/análisis de resultados	4
1.4 Marco teórico	4
1.4.1. Protocolo de Comunicación.....	4
1.4.2. Estándar de Comunicación	5
1.4.3. Protocolo Modbus	5
1.4.4. Tipos de Protocolo Modbus.....	6
1.4.5. Modbus RTU	7
1.4.6. Interfaz RS-485	8
1.4.7. Trama Modbus	9
1.4.8. Funciones Modbus.....	14
1.4.9. Modos de Transmisión	15
2 METODOLOGÍA	17
2.1 Requerimientos del prototipo	17
2.2 Selección de equipos y herramientas	18
Regulador lineal de voltaje	18
Convertidores Buck	19
ATMega 328P	19
Placa de desarrollo ESP32	20
Módulo Max-485.....	21

SN75176	21
LCD Oled 128*64	22
LCD 16*2	22
2.3 Diseño del prototipo.....	23
2.3.1. Etapa de Alimentación.....	23
2.3.1.1. Selección del regulador de voltaje:	24
2.3.1.2. Diseño de la etapa de alimentación	24
2.3.2. Etapa de adquisición.....	29
2.3.3. Etapa de comunicación.....	30
2.3.4. Etapa de visualización.....	31
2.3.5. Etapa de configuración	32
2.3.6. Etapa de control	33
2.4 Implementación del prototipo.....	41
3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES.....	44
3.1 Resultados	44
3.1.1. Presentación del Prototipo	44
3.1.2. Prueba de envío de mensaje	45
3.1.3. Prueba de comunicación con un ordenador	46
3.1.4. Prueba de comunicación con PLC uno a uno	48
3.1.5. Prueba de comunicación en una red Modbus	52
3.1.6. Análisis económico	57
3.2 Conclusiones.....	59
3.3 Recomendaciones.....	61
4 REFERENCIAS BIBLIOGRÁFICAS	62
5 ANEXOS.....	64
5.1 ANEXO I: Esquema del Prototipo	68
5.2 ANEXO II: Identificación de las etapas en la placa	66
5.3 ANEXO III: Vistas del prototipo	67
5.4 ANEXO IV: Manual de configuración software Modbus Poll.....	68
5.5 ANEXO V: Códigos de errores en el bloque MSG	76
5.6 ANEXO VIII: Manual de Usuario	79

RESUMEN

El presente trabajo de integración curricular se centra en el diseño y desarrollo de un módulo de adquisición y comunicación basada en el protocolo de bus de campo serial Modbus RTU. Modbus RTU es un protocolo de comunicación muy utilizado en sistemas industriales para la interconexión de dispositivos electrónicos dedicados al control y la adquisición de datos. El objetivo principal es crear un módulo de adquisición de una señal analógica que, a su vez, permita una comunicación eficiente y fiable entre sí y un controlador lógico programable.

El desarrollo del módulo incluyó varias etapas clave: análisis del protocolo Modbus RTU, diseño del hardware, programación del firmware y pruebas de integración y validación. El diseño del hardware abarcó aspectos como la selección de microcontroladores adecuados, circuitos de interfaz y comunicación RS-485, y la disposición de componentes para minimizar interferencias y maximizar la integridad de la señal. En cuanto al firmware, se implementaron funciones para la codificación y decodificación de tramas Modbus, manejo de errores, y gestión de la comunicación en tiempo real.

Las pruebas de integración y validación se llevaron a cabo en el laboratorio de redes industriales de la Escuela Politécnica Nacional, utilizando equipos de simulación y dispositivos reales. Los resultados demostraron que el módulo desarrollado cumple con los requisitos de interoperabilidad y rendimiento, asegurando una comunicación estable y rápida con el PLC. Este proyecto aporta una solución flexible y escalable para la comunicación industrial, ofreciendo un diseño modular que puede ser adaptado a diferentes aplicaciones y entornos industriales.

PALABRAS CLAVE: Modbus RTU, Interfaz RS-485, Trama de Datos, Baud Rate, Comunicación semidúplex.

ABSTRACT

This curricular integration paper focuses on the design and development of an acquisition and communication module based on the Modbus RTU serial fieldbus protocol. Modbus RTU is a communication protocol widely used in industrial systems for the interconnection of electronic control and data acquisition devices. The main goal is to create an analog signal acquisition module that, in turn, enables efficient and reliable communication between itself and a programmable logic controller.

The development of the module included several key stages: Modbus RTU protocol analysis, hardware design, firmware programming, and integration and validation testing. The hardware design covered aspects such as the selection of suitable microcontrollers, RS-485 interface and communication circuits, and the layout of components to minimize interference and maximize signal integrity. As for the firmware, functions for Modbus frame encoding and decoding, error handling, and real-time communication management were implemented.

Integration and validation tests were carried out in the industrial network laboratory of the National Polytechnic School, using simulation equipment and real devices. The results showed that the developed module meets the interoperability and performance requirements, ensuring a stable and fast communication with the PLC. This project provides a flexible and scalable solution for industrial communication, offering a modular design that can be adapted to different applications and industrial environments.

KEYWORDS: Modbus RTU, RS-485 Interface, Data Frame, Baud Rate, Half-Duplex Communication.

1 INTRODUCCIÓN

Actualmente es evidente un avance tecnológico en cualquier aspecto que una persona pudiese imaginar, desde carros inteligentes hasta dispositivos móviles como los que se usan a diario, no obstante, el mundo de las comunicaciones industriales también ha tenido un remarcado avance tecnológico que se viene dando desde los años 80, donde se implementaron los primeros bosquejos de lo que hoy por hoy se conocen como redes de comunicación industrial, ya para los años 90 muchos fabricantes apostaron por el diseño e implementación de buses industriales de comunicación basados en algunos estándares, garantizando así la intercomunicación entre sus productos, dejando de lado la transmisión analógica de datos, misma que utilizaba señales de corriente normalizadas en un rango de 4 a 20 mA, siendo caracterizadas por un constante cambio en la amplitud de la señal de corriente, lo que le otorgaba características muy vulnerables para los entornos industriales, tanto por el ruido presente como la limitación en la distancia [1], [2].

En los sistemas industriales modernos el intercambio de información entre los diferentes elementos, sistemas, procesos etc., que lo conformen, suponen el principal pilar para estar a la altura de la competitividad que exige el mercado de los productos finales, además son consideradas como el sistema nervioso central de cualquier arquitectura en los procesos de automatización, ya que le otorgan al sistema características como la flexibilidad y controlabilidad de la mayoría de dispositivos, así como una herramienta rápida y poderosa para el intercambio de datos [3], [4].

Con todo lo mencionado anteriormente, el presente documento pretende detallar e ilustrar paso a paso el diseño e implementación de un prototipo de tarjeta de comunicación para la lectura de una señal analógica proveniente de cualquier sensor, cuya salida entregue un valor de voltaje estandarizado en un rango de 0 a 10 voltios. Dicho prototipo hará uso del protocolo Modbus para unidades terminales remotas (RTU), suponiendo la integración de componentes electrónicas de bajo costo que le otorguen un cierto grado de robustez, y con ello, conviertan al prototipo en una alternativa accesible y escalable en el campo de los dispositivos de medición y comunicación, respetando los requerimientos del prototipo y los medios necesarios para que el intercambio de información sea exitoso y pueda ser, hasta cierto punto, didáctico para quien opere el prototipo.

1.1 Objetivo general

Diseñar e implementar un prototipo de módulo de comunicación industrial para la adquisición de un dato analógico, y su envío a través de un bus de campo serial propio del protocolo Modbus RTU.

1.2 Objetivos específicos

1. Revisar bibliografía relacionada con el protocolo de comunicación Modbus RTU, sus requerimientos, limitaciones, su funcionamiento, y los campos que constituyen la trama en el intercambio de información.
2. Analizar y seleccionar, según las necesidades del proyecto, componentes electrónicas que permitan una comunicación eficiente entre el PLC y el prototipo, y supongan una solución flexible y escalable.
3. Diseñar la circuitería y electrónica necesaria para la integración de los diferentes componentes que se requieren para su correcto funcionamiento y su vinculación con el usuario.
4. Desarrollar la programación necesaria para el intercambio de datos y la correcta comunicación, de acuerdo con la trama de datos y los requerimientos del protocolo de comunicación.
5. Evaluar el desempeño del módulo considerando aspectos como velocidad de transmisión, robustez y operabilidad por parte del usuario.

1.3 Alcance

El alcance de este proyecto se divide en varias fases, desde una fase teórica hasta una fase de pruebas, mismas que se detallan a continuación:

A. Fase Teórica

- Se propone una recopilación bibliográfica de los conceptos fundamentales referentes a las comunicaciones industriales, modos de transmisión y el protocolo Modbus propiamente, particularizando el protocolo RTU, con la finalidad de realizar un estudio y entendimiento de la composición de la trama, todos los campos e identificadores que supone dicho protocolo para la comunicación entre dispositivos.

- Se realizará una síntesis orientada a Modbus RTU con su modo de transmisión Half dúplex, ya que Modbus RTU supone un modelo request/response (solicitud/respuesta) para las redes maestro/esclavo que se levantan sobre este protocolo. Además del entendimiento de la interfaz RS-485 para la transmisión de información y la comprensión de la importancia de esta interfaz en entornos industriales.
- Se realizarán búsquedas de hardware y software que permitan el desarrollo del prototipo propuesto, así como un entendimiento y análisis del estado de arte para tarjetas de comunicación y adquisición de datos levantadas sobre Modbus RTU.
- Se seleccionarán las herramientas previamente buscadas tanto de hardware y software que permitan el diseño del prototipo mencionado, priorizando el beneficio económico en su elaboración, de esta forma se podrá realizar un análisis de entradas y salidas y con ello la selección de la tarjeta embebida o microcontrolador a utilizar, que además permita mejorar el prototipo a futuro.

B. Fase de diseño

- Se realizará el diseño de hardware del prototipo incluyendo las etapas de adquisición y acondicionamiento de la señal analógica a medir, alimentación general del prototipo y los diferentes módulos que se integren para la comunicación y la transmisión de datos.
- Se realizará la programación del software necesario para comandar el comportamiento del prototipo desarrollado el cual deberá contener el armado, envío y recepción de la trama Modbus RTU, así como lectura de la señal analógica proveniente del conversor analógico digital, y la comunicación con los diferentes módulos que se integren para la transmisión de información.
- Se realizarán los esquemáticos necesarios en cualquier software de diseño de placas de circuito impreso PCB para llevar el diseño del prototipo de protoboard a una placa electrónica, con la finalidad de reducir sus dimensiones y hacerla amigable con el operador.

C. Fase de implementación

- Con la placa PCB diseñada se realizará la programación necesaria en los PLC's del Laboratorio de Redes Industriales de la Escuela Politécnica Nacional, para comprobar el envío y transmisión de la trama, y con ello, verificar que se cumplan cada uno de sus identificadores gracias al chequeo de errores en el envío y recepción de datos.

- Se implementará la placa desarrollada junto a elementos que generen interferencias electromagnéticas (EMI's) como lo es el motor con el que cuenta el laboratorio para las prácticas de control de velocidad, con la finalidad de probar el desempeño de la tarjeta desarrollada.

D. Fase de pruebas/análisis de resultados

- Finalmente, se realizarán las pruebas de validación en la lectura de la señal analógica proveniente de cualquier sensor que genere una señal de voltaje normalizada en un rango de 0-10V, y el comportamiento de la tarjeta en ambientes con y sin EMI's para poder generar normas de uso.
- Analizar las lecturas obtenidas y poder establecer la precisión de la tarjeta de comunicación, y su sensibilidad en función de las lecturas contrastadas con elementos de medición.

1.4 Marco teórico

En esta sección se explicarán los fundamentos teóricos necesarios para el desarrollo del trabajo de integración curricular propuesto. Para facilitar la comprensión del proyecto, se resumirán los conceptos revisados y aplicados, comenzando con criterios generales y avanzando hacia casos específicos según las necesidades del proyecto. También se detallarán las consideraciones para el diseño de hardware, que, al combinarse con los criterios de software, permitieron la creación del prototipo descrito en este documento.

1.4.1. Protocolo de Comunicación

La definición de protocolo de comunicación no es más que la de un conjunto de normas, reglas, pautas o instrucciones formales, cuyo cumplimiento es de carácter mandatorio ya que guían las acciones para un exitoso y ordenado intercambio de información [5].

En este conjunto de normas se establece el formato de datos, la secuenciación, temporización, control de acceso y control de errores, entre otras, de forma que para que dos dispositivos puedan establecer comunicación deben utilizar el mismo protocolo.

La necesidad de conectar dispositivos llevó al desarrollo de protocolos de comunicación. Actualmente, existen protocolos que permiten la interoperabilidad entre dispositivos de distintos fabricantes, pero también hay protocolos específicos de cada fabricante para sus propios dispositivos. Por eso se creó el Modelo OSI, que establece marcos de referencia estandarizados para facilitar la conexión y compatibilidad entre diferentes sistemas [6], [7].

1.4.2. Estándar de Comunicación

La definición de estándar de comunicación no se aleja de la definición de protocolo de comunicación, ya que se define como reglas o acuerdos que contienen especificaciones y criterios técnicos para ser utilizados, con la única diferencia que estos acuerdos están documentados y normados para regular la transmisión de datos en los sistemas de comunicación [8].

Las reglas que definen cómo deben comunicarse los sistemas son establecidas por organismos de normalización como ISO, IEEE y ANSI. Estos protocolos determinan el formato lógico de la comunicación, mientras que los estándares especifican los requisitos necesarios para asegurar la compatibilidad, interoperabilidad, escalabilidad y seguridad entre diferentes sistemas. En resumen, un protocolo se enfoca en cómo se establece la comunicación, mientras que un estándar se encarga de definir las normas para garantizarla [9].

1.4.3. Protocolo Modbus

Modbus es un protocolo de comunicación empleado en sistemas de automatización industrial, operando en la capa de aplicación del modelo OSI. Permite la comunicación entre dispositivos, como controladores y sensores, mediante un esquema maestro-esclavo. Modbus facilita la transmisión de datos en forma de solicitudes y respuestas, permitiendo la lectura y escritura de datos a través de una red serie. Es conocido por su simplicidad, eficiencia y amplia adopción en la industria para redes de comunicación en sistemas de control [10].

Modbus fue desarrollado en 1979 por Modicon, actualmente Schneider Electric, como un protocolo de comunicación estándar no oficial. Aunque no estaba regulado por organismos de estandarización, su simplicidad y compatibilidad lo hicieron ampliamente aceptado y utilizado en la industria. Su estructura, basada en códigos de función, facilitó su adopción [10].

Modbus se ha destacado entre otros protocolos de comunicación por estar diseñado específicamente para aplicaciones industriales. Su carácter abierto y su sencilla implementación contribuyen a su popularidad. Aunque existen varias versiones de Modbus, muchas han sido descontinuadas con la llegada de internet, siendo las versiones de las capas 1, 2 y 7 del modelo OSI las más relevantes. La figura 1.1 ilustra cómo se relaciona Modbus con el modelo OSI [11].

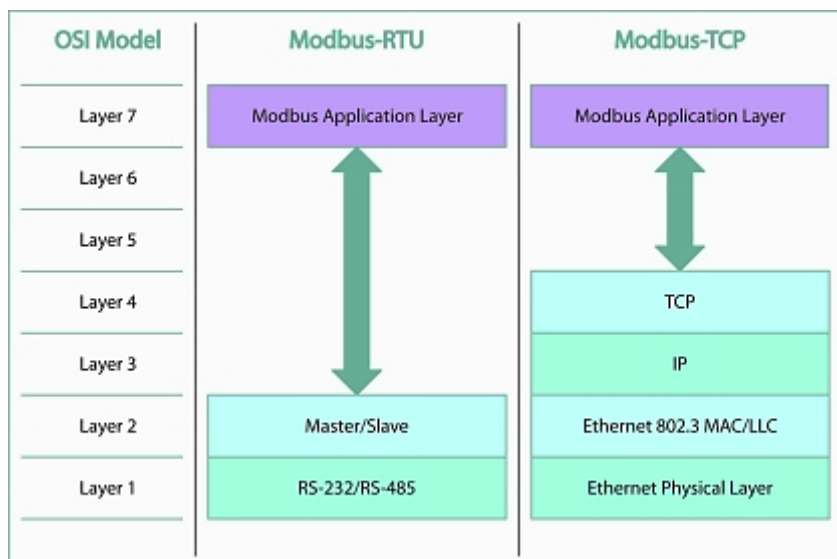


Figura 1.1 Modbus y el modelo OSI [12]

1.4.4. Tipos de Protocolo Modbus

Con el paso del tiempo Modbus se ha convertido en un protocolo frecuentemente usado por muchas industrias, por lo cual fueron requeridas varias adaptaciones del protocolo según las necesidades de los sistemas de automatización industrial, de estas adaptaciones surgieron cuatro versiones más conocidas [11].

- **Modbus RTU:** Este protocolo destaca de las otras versiones porque fue diseñado para aplicaciones de comunicación inalámbrica, de las señales emitidas por los dispositivos de control e instrumentación a un sistema de adquisición y control de datos SCADA. Utiliza lenguaje binario y posee una sólida verificación de errores conocida como CRC o chequeo de redundancia cíclica, esta versión es la más utilizada en aplicaciones de campo ya que está estrechamente ligada a la producción automatizada [11].
- **Modbus TCP:** Esta versión de Modbus se introdujo para aprovechar las ventajas de la arquitectura LAN, aumentando la cantidad de dispositivos que pueden ser integrados a la red, debido a que el número de dispositivos y la distancia de conexión es una gran limitante en las otras versiones. Una de sus características es que Modbus TCP utiliza el puerto 502 pudiendo reasignarse de ser necesario [11].

Otra característica importante de Modbus TCP es que tiene un modo de comunicación lógico de tipo multicast, lo que significa que en esta versión del protocolo puede existir más de un cliente y más de un esclavo, considerando al esclavo como servidor y al maestro como cliente [11].

- **Modbus ASCII:** Modbus ASCII es la versión más antigua desarrollada por Modbus, los elementos de su trama son los mismos que los del protocolo RTU, con la diferencia que estos están expresados en código ASCII, es decir, caracteres hexadecimales de 4 bits cada uno imprimibles. Este protocolo facilita la monitorización de la actividad en uno o varios puertos seriales, no obstante, este protocolo se considera obsoleto ya que ahora el protocolo RTU posee las mismas funcionalidades que Modbus ASCII [11].
- **Modbus PLUS:** Modbus +, es habitualmente considerado como un protocolo a pesar de ser un sistema completo de alta velocidad entre pares, basado en la comunicación por tokens o turnos, posee un medio predefinido y se implementa en la capa física o capa 1 del modelo OSI, al ser de alta velocidad utiliza la arquitectura de red LAN para aplicaciones industriales [11].

1.4.5. Modbus RTU

Como ya se mencionó anteriormente, Modbus RTU es el protocolo escogido sobre el cual estará basado el prototipo propuesto, de esta manera, este protocolo utiliza la arquitectura maestro-esclavo para la comunicación de dispositivos montados sobre una interfaz RS-485. La comunicación maestro-esclavo es de tipo solicitud/respuesta en donde el maestro inicia la transmisión de datos en el bus de comunicación o iniciando la acción de solicitud, y donde únicamente el dispositivo o esclavo que contenga la información de la solicitud responderá, mientras que, los otros dispositivos ignorarán dicha solicitud [13].

Cada dispositivo en la red Modbus tiene una dirección única que debe estar entre 1 y 247, siempre que no haya repetidores en la red. Esta dirección es un parámetro crucial en la trama, ya que cuando el maestro inicia la comunicación, incluye la dirección del dispositivo al que se dirige la solicitud. Solo el dispositivo con esa dirección específica responderá. Aunque el protocolo permite que el maestro envíe mensajes de broadcast a toda la red, los dispositivos esclavos solo responden a solicitudes dirigidas individualmente [13].

Cabe aclarar que el estándar RS-485 permite conectar hasta 32 dispositivos en un mismo bus de comunicación, debido a la carga eléctrica que cada dispositivo impone al bus, conocida como unidad de carga (UL). No obstante, cuando se utiliza el protocolo Modbus RTU sobre una red RS-485, es posible direccionar hasta 247 dispositivos esclavos gracias a su esquema de direccionamiento. Sin embargo, esta capacidad de direccionamiento no modifica la limitación física establecida por el estándar RS-485. Por lo tanto, Modbus RTU puede integrar hasta 247 dispositivos utilizando técnicas como repetidores, dispositivos de alta impedancia o segmentación de la red [14].

1.4.6. Interfaz RS-485

El estándar TIA/EIA-485, o RS-485, define una interfaz de comunicación diferencial que permite conectar hasta 32 dispositivos, cada uno funcionando como una unidad de carga. Algunos dispositivos de bajo consumo pueden permitir conectar más de 32 unidades. El medio físico más común es el par trenzado, que facilita la transmisión de datos mediante un solo par de cables. En modo half-dúplex, cada dispositivo activa su transmisor solo cuando envía datos, permitiendo que otros dispositivos usen la línea en otros momentos. Con dos pares trenzados, la red puede operar en modo full-dúplex y es compatible con el estándar RS-422 [15].

RS-485 utiliza un medio físico de comunicación diferencial conocido como "par trenzado". En esta interfaz, los transmisores y receptores envían datos mediante la diferencia de tensión entre los conductores del par trenzado. Los niveles binarios se determinan por la polaridad de esta diferencia de tensión: si la tensión en el conductor "+" es mayor que en el conductor "-", se interpreta como un "1" lógico; y si es al revés, se interpreta como un "0" lógico, como se muestra en la figura 1.2.

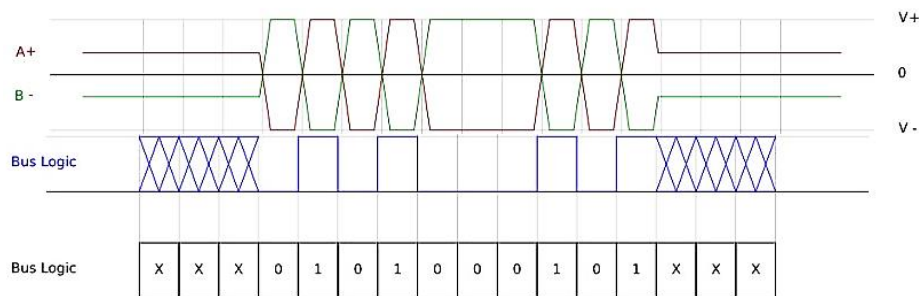


Figura 1.2. Forma de onda del estándar RS-485 [16]

Se establece un margen de ruido de $\pm 0,2$ V para mejorar la tolerancia a las interferencias. Esta técnica cancela los ruidos inducidos en el medio de transmisión, ya que cualquier ruido afectaría ambos conductores por igual, manteniendo constante la diferencia de tensión y preservando la información. Además, la interferencia electromagnética emitida por un bus de comunicación diferencial es menor en comparación con los buses no diferenciales [15].

Resistencia de Terminación

La teoría de las comunicaciones indica que es esencial terminar las líneas de comunicación con una impedancia adecuada para reducir reflexiones que distorsionan los datos, mejorando la velocidad y el alcance de la red. La terminación correcta para esta interfaz se ilustra en la figura 1.3 [15].

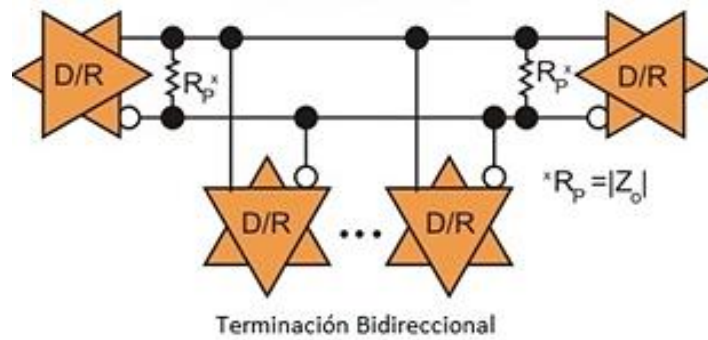


Figura 1.3. Tipos de terminaciones según el estándar RS-485 [15]

- **Terminación bidireccional:** Asegura una buena integridad de señal, permitiendo colocar controladores en cualquier lugar de la red, aunque incrementa el consumo de energía [15].

La impedancia característica ideal para la terminación en un par trenzado es aproximadamente de 120 ohmios [15].

Topología

La topología de una red define cómo se configuran los nodos y las líneas para transmitir datos. Con RS-485, a velocidades y distancias moderadas, la topología afecta poco al rendimiento. Sin embargo, para minimizar reflexiones y errores de comunicación, una topología simple como la "Daisy Chain", donde los dispositivos se conectan directamente en la línea principal, es eficaz. Esto se muestra en la figura 1.4 [15].

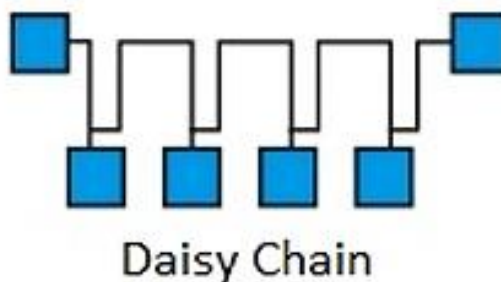


Figura 1.4. Topología en el estándar RS-485 [15]

1.4.7. Trama Modbus

El protocolo Modbus establece una unidad de datos de protocolo, llamada PDU, que es simple y no depende de las capas de comunicación subyacentes. La asignación del protocolo Modbus a diferentes buses o redes específicas puede introducir algunos campos adicionales, estos campos están contenidos en la unidad de datos de aplicación (ADU) tal como se muestra en la figura 1.5 [10].

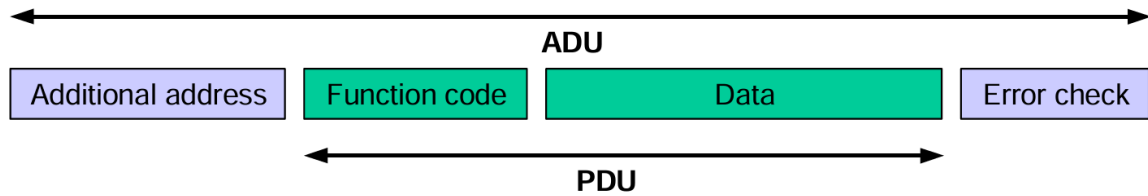


Figura 1.5. Trama general Modbus [10]

- **PDU:** El Protocol Data Unit (PDU) es el mensaje principal intercambiado entre el cliente y el servidor. Consiste en un código de función y los datos correspondientes, y su estructura es uniforme, independientemente del tipo de red utilizada para el transporte de datos. En general, el PDU incluye un código de función, que es un byte que indica la operación a realizar (como leer o escribir), y los datos necesarios para ejecutar esa operación [10].
- **ADU:** El Application Data Unit (ADU) es la unidad completa de datos que incluye el Protocol Data Unit (PDU) junto con información adicional específica del medio de comunicación utilizado. En otras palabras, el ADU es el PDU más ciertos componentes adicionales necesarios para la transmisión a través de una red particular. La estructura del ADU varía según el tipo de red e incluye típicamente un encabezado con información específica del medio, como la dirección del dispositivo o el identificador de transacción en TCP/IP, además de un campo de verificación de errores que varía según la red, como el CRC en conexiones seriales [10].

La unidad de datos de aplicación Modbus es generada por el maestro al iniciar una transacción. La función en el mensaje indica al cliente la acción a realizar. Modbus codifica el campo de función en un byte, con códigos válidos del 1 al 255; sin embargo, los códigos del 128 al 255 se reservan para respuestas de excepción. El campo de función en el mensaje del maestro al esclavo especifica la acción requerida. El código "0" no es válido, y algunos códigos incluyen subfunciones para acciones múltiples [10].

En los mensajes del maestro al esclavo, el campo de datos incluye información adicional como direcciones y cantidades necesarias para la acción indicada por el código de función. En algunos casos, este campo puede estar vacío si el código de función es suficiente. Si no hay errores, la respuesta del esclavo contiene los datos solicitados; si hay errores, incluye un código de excepción que indica la próxima acción a tomar. El maestro puede leer estados ON/OFF o leer/escribir datos en registros. El esclavo usa el campo de código de función para señalar si la respuesta es normal o si hubo un error. En respuestas normales, el esclavo replica la solicitud con el código de función original. La figura 1.6 resume estas transacciones Modbus [10].

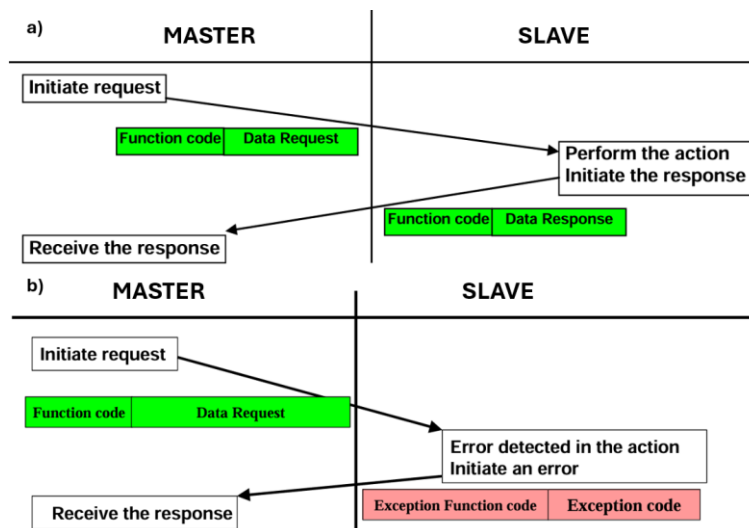


Figura 1.6. a) Transacción Modbus sin error, b) Transacción Modbus con error [10]

De esta forma, para un mejor entendimiento se explica la trama del protocolo Modbus RTU por parte del maestro y del esclavo por separado.

Trama del Maestro

Como ya se mencionó antes, la trama de este protocolo incluye el PDU que a su vez este contiene el código de función y los datos a transmitir, particularmente para el protocolo RTU, el ADU incluye el campo de dirección, utilizado para identificar a los diferentes esclavos, y el campo de CRC (chequeo de redundancias cíclicas), utilizado para el chequeo de errores, el envío de solicitudes se realiza a nivel de bits como se ilustra en la figura 1.7.

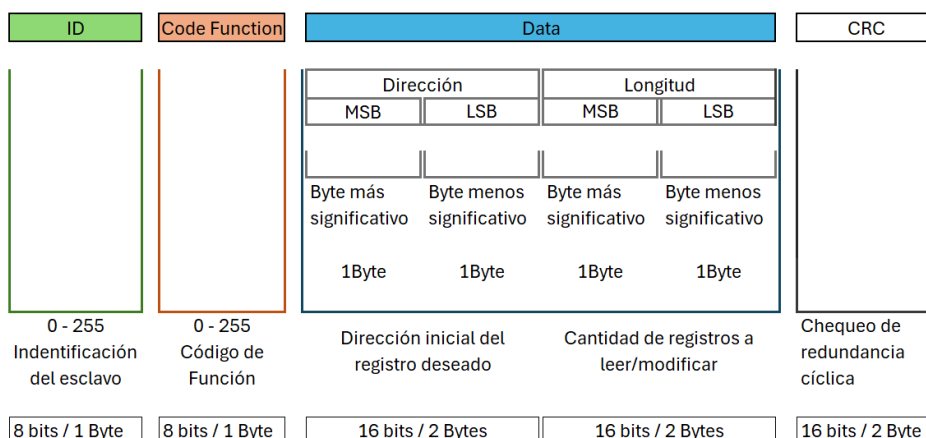


Figura 1.7. Trama de datos del maestro para solicitud al esclavo

Es importante destacar que, al manejar tramas a nivel de bit, los diferentes campos se agrupan en bytes, permitiendo hasta 256 posibilidades de direccionamiento o funciones. Sin embargo, en el campo ID, los esclavos a los que se envían solicitudes deben estar en el rango de 1 a 247.

La dirección cero está reservada para el maestro, y las direcciones del 248 al 255 están reservadas para propósitos especiales o futuros, según la especificación Modbus [17].

Trama del Esclavo

Una vez conocida la trama que el maestro coloca en el bus de datos para iniciar la comunicación con los dispositivos esclavos, es preciso conocer la forma en la que estos responden. Para el caso de los esclavos, el campo ID y función se mantiene igual que en el maestro, el cambio en la trama se da en el campo DATA, ya que el esclavo debe indicar el número de bytes que necesita para poder responder la solicitud del maestro, seguido de los dos bytes para el chequeo de redundancia cíclica (CRC), con todo lo mencionado se ilustra la figura 1.8.

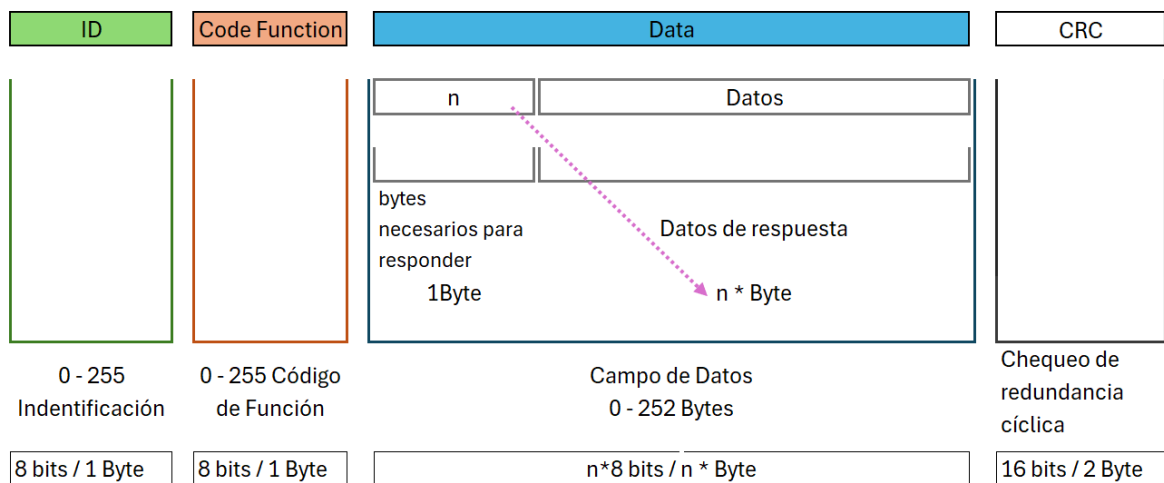


Figura 1.8. Trama de datos de los esclavos para solicitud al esclavo

Mensajes Modbus

Después de comprender las tramas del maestro y del esclavo, es importante entender cómo se relacionan. El dispositivo maestro envía un mensaje, que es una solicitud con la trama Modbus, a través de un bus serial. Este mensaje tiene un punto inicial y final, lo que permite a los dispositivos esclavos identificar el comienzo y el fin del mensaje. En el modo RTU, las tramas de mensajes están separadas por un intervalo de silencio de al menos $t_{3.5}$, que equivale a 3.5 veces el tiempo de transmisión de un carácter completo, como se muestra en la figura 1.9 [17].

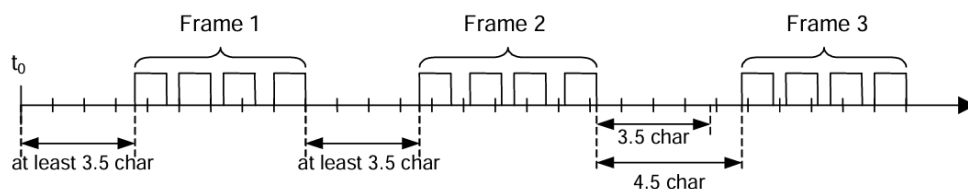


Figura 1.9. Trama de mensajes RTU [17]

El tiempo de inactividad ($t_{3.5}$) se basa en el tiempo requerido para transmitir un carácter completo, que consta de 1 bit de inicio, 8 bits de datos, 1 bit de paridad (si se usa) y 1 bit de parada, totalizando 11 bits por carácter. El tiempo de transmisión de un carácter depende de la velocidad de transmisión (baud rate) y se calcula con la siguiente fórmula [17].

$$T_{CP} = \frac{11 \text{ bits}}{\text{Baud Rate}}$$

Ecuación 1.1: Velocidad de transferencia de datos [18]

Por lo tanto, el tiempo de inactividad $t_{3.5}$ se calcula como:

$$t_{3.5} = 3.5 \cdot T_{CP} = 3.5 \cdot \frac{11 \text{ bits}}{\text{Baud Rate}}$$

Ecuación 1.2: Tiempo de inactividad para Modbus RTU [17]

Este intervalo asegura que los dispositivos en la red puedan distinguir el final de un mensaje y el comienzo del siguiente, garantizando una comunicación efectiva. La trama del mensaje debe transmitirse como un flujo continuo de caracteres; si hay un intervalo de silencio superior a 1,5 veces el tiempo de un carácter entre dos caracteres, la trama se considera incompleta y debe ser descartada por el maestro, como se muestra en la figura 1.10 [17].

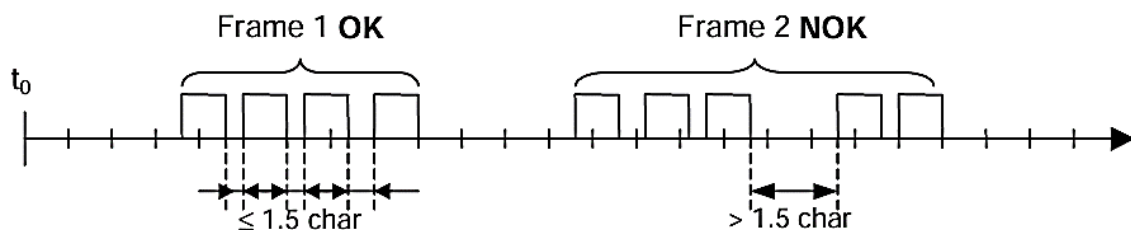


Figura 1.10. Esquema de comunicación sin y con errores respectivamente [17]

Chequeo de Redundancia Cíclica CRC

El CRC (Cyclic Redundancy Check) en Modbus RTU es un algoritmo de verificación de errores diseñado para asegurar la integridad de los datos en las comunicaciones. Este algoritmo detecta errores en la transmisión generando un código de verificación de 16 bits, que se adjunta al final de cada mensaje [17].

Al recibir un mensaje, el dispositivo receptor recalcula el CRC y lo compara con el valor recibido. Si los valores coinciden, el mensaje se considera válido, garantizando la fiabilidad y precisión en la transferencia de información en Modbus RTU [17].

El cálculo del CRC en Modbus RTU utiliza un algoritmo del fabricante con un polinomio generador de 16 bits. Comienza con un registro CRC de 16 bits inicializado en 65535 (0xFFFF en hexadecimal). Para cada byte del mensaje, se realiza un XOR con el registro CRC y luego un desplazamiento a la derecha bit a bit, repitiendo el proceso 8 veces. Si el bit menos significativo es 1, se realiza un XOR adicional con el valor constante hexadecimal 0xA001, conocido como dato polinomial, para asegurar que el CRC detecte cambios en los datos transmitidos [17].

1.4.8. Funciones Modbus

En Modbus, el campo de función puede abarcar valores del 0 al 255, teóricamente permitiendo hasta 256 funciones. Sin embargo, solo los códigos del 1 al 127 son válidos según la especificación estándar de Modbus. Los códigos del 128 al 255 están reservados para usos específicos o no estándar. Así, aunque se pueden tener hasta 256 valores distintos, la mayoría de las implementaciones de Modbus utilizan las funciones estándar definidas en la especificación oficial, como se muestra en la figura 1.11 [19].

				Function Codes		(hex)	Section
				code	Sub code		
Data Access	Bit access	Physical Discrete Inputs	Read Discrete Inputs	02		02	6.2
		Internal Bits Or Physical coils	Read Coils	01		01	6.1
			Write Single Coil	05		05	6.5
			Write Multiple Coils	15		0F	6.11
	16 bits access	Physical Input Registers	Read Input Register	04		04	6.4
			Read Holding Registers	03		03	6.3
		Internal Registers Or Physical Output Registers	Write Single Register	06		06	6.6
			Write Multiple Registers	16		10	6.12
			Read/Write Multiple Registers	23		17	6.17
		Mask Write Register	22		16	6.16	
		Read FIFO queue	24		18	6.18	
	File record access	Read File record		20		14	6.14
		Write File record		21		15	6.15
	Diagnostics	Read Exception status		07		07	6.7
		Diagnostic		08	00-18,20	08	6.8
Get Com event counter		11		0B	6.9		
Get Com Event Log		12		0C	6.10		
Report Server ID		17		11	6.13		
Read device Identification		43	14	2B	6.21		
Other	Encapsulated Interface Transport		43	13,14	2B	6.19	
	CANopen General Reference		43	13	2B	6.20	

Figura 1.11: Funciones Modbus públicas [10]

La simplicidad de estas funciones públicas es que están pensadas para manejar únicamente 4 tipo de datos, los cuales se agrupan de la siguiente manera:

- **COILS:** Salidas discretas o digitales (bobinas) de 1 bit de tamaño, que pueden ser forzadas, y son utilizados para representar estados de encendido/apagado de actuadores asociados a las salidas de un PLC [20].

- **CONTACTS:** Entradas discretas o digitales de 1 bit de tamaño, que pueden ser forzadas, y son utilizados para la lectura de sensores discretos, estados de interruptores (abierto/cerrado) asociados a las entradas de un PLC [20].
- **INPUT REGISTERS:** Entradas a nivel de registro, es decir, agrupación de 8 bits completos, de 2 bytes de tamaño utilizado para lectura de datos provenientes de otro PLC o dispositivo esclavo, únicamente son registros de lectura [20].
- **HOLDING REGISTERS:** Este tipo de dato se asemeja a un registro de entrada como valor de retención, se diferencia del Input Register en su uso, ya que comúnmente son utilizados para almacenar configuraciones, estados, consignas y además estos datos pueden ser de escritura y posee un tamaño de 2 bytes [20].

En términos formales, tanto los COILS como los HOLDING REGISTERS son datos de salida del maestro, lo que significa que son bits y registros que el maestro puede escribir en el esclavo. Aunque el maestro también puede leer estos datos para conocer su valor actual, esta acción no es su función principal, para un mejor entendimiento se presenta la figura 1.12 correspondiente al tipo de registro de Modbus [20].

Tablas primarias	Tipo de objeto	Tipo de
Entrada Discreta	Un bit	Sólo lectura
Bobinas	Un bit	Lectura- Escritura
Registros de entrada	Palabra de 16 bits	Sólo lectura
Registros de retención	Palabra de 16 bits	Lectura- Escritura

Figura 1.12. Tipos de registro para Modbus [10]

1.4.9. Modos de Transmisión

Los modos de transmisión se pueden definir como la forma en la que se transmiten los datos de los sistemas electrónicos de comunicaciones, estos modos pueden ser en una sola o en ambas direcciones, solo una o varias a la vez y la combinación entre estas. Existen 3 tipos de modos de transmisión posibles: simplex, semidúplex o half dúplex y full dúplex, sin embargo, se mencionará únicamente el modo sobre el cual funciona Modbus RTU.

Transmisión Semidúplex

En modo de transmisión semidúplex o también conocida como half dúplex o sistemas “cambio y fuera”, las transmisiones se pueden realizar en varias direcciones a varios dispositivos, pero no a la vez. En contraste con el modo de transmisión anterior, ahora cualquier dispositivo puede ser transmisor o receptor, pero no al mismo tiempo. Un claro ejemplo de esto son los sistemas de radio de tipo PTT (Push to talk) o walkie talkie, utilizados por servicios de transporte, emergencia, seguridad y demás, como ejemplo de lo antes mencionado se evidencia en la figura 1.13 un diagrama de este modo de transmisión [5].

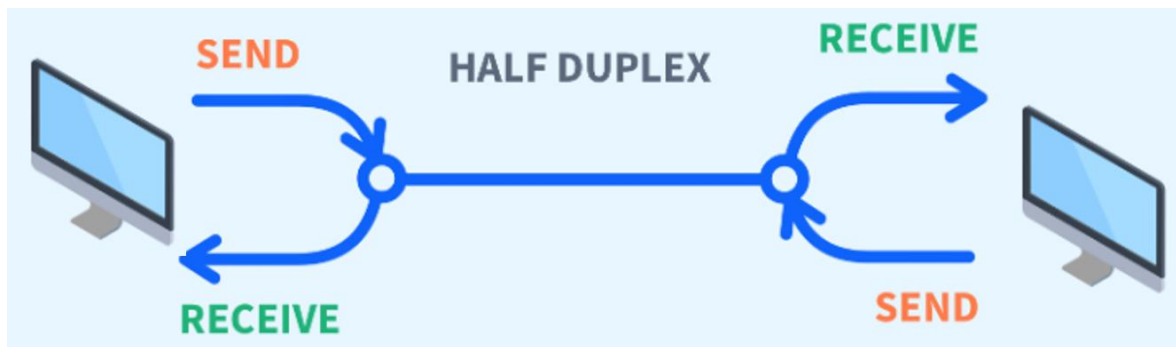


Figura 1.13. Modo de transmisión tipo semidúplex [21]

2 METODOLOGÍA

Como se mencionó en el capítulo anterior, el protocolo Modbus RTU está diseñado para dispositivos remotos en campo. Aunque el módulo propuesto no estará en campo, emulará este comportamiento al estar ubicado en el laboratorio de redes industriales de la Escuela Politécnica Nacional. Así, el desarrollo de este componente electrónico sigue una metodología experimental, específicamente de campo, ya que se realizarán pruebas bajo condiciones específicas.

El enfoque del proyecto será cualitativo, evaluando características como el desempeño en entornos desfavorables, el intercambio de información y cómo estos factores afectan el rendimiento. En este capítulo se detallará el diseño e implementación del prototipo de comunicación para un sensor analógico basado en Modbus RTU. Se realizará un análisis de las necesidades a nivel de hardware y software, priorizando criterios de ingeniería para seleccionar componentes en función de su durabilidad, costo comercial y robustez del producto final.

2.1 Requerimientos del prototipo

El proyecto debe cumplir con las siguientes características: leer un valor analógico en un rango de 0 a 10V, enviado por un sensor con esa salida de voltaje. La señal será capturada por un microcontrolador mediante un conversor analógico-digital, que luego construirá la trama del protocolo y enviará los datos de la conversión a través de una interfaz RS-485 utilizando el protocolo Modbus RTU.

Los requerimientos nacen a partir del diseño de cualquier componente electrónico, mismos que se detallan a continuación:

- **Alimentación:** Como cualquier otro componente electrónico, para su funcionamiento requiere una fuente de alimentación, pudiendo ser esta de corriente alterna AC o corriente continua DC, generalmente los elementos de medición ya sean analógicos o digitales suelen situarse en campo, si se particulariza a los sensores digitales, estos requieren un flujo de energía para su funcionamiento.
- **Interfaz de Comunicación:** Modbus RTU está basado en el estándar de comunicación definido por la norma TIA/EIA-485-A, siendo ampliamente utilizada en aplicaciones industriales debido a su robustez ya que utiliza un par trenzado a nivel diferencial, lo que evita la pérdida de datos por interferencias electromagnéticas pudiendo extenderse hasta 1200m utilizando hasta 32 dispositivos sin repetidores a una velocidad de 100kbps [10].

- **Velocidad de transferencia:** RS-485 puede operar a una velocidad de hasta 115.2kbps adaptable a velocidades de los controladores más comerciales, de forma particular esta velocidad estará condicionada por el PLC que figure como maestro de la red Modbus, y los convertidores adicionales que utiliza el laboratorio de redes industriales para establecer comunicación [15].
- **Compatibilidad:** El producto final debe ser capaz de conectarse a cualquier dispositivo que se encuentre en la interfaz RS-485, ya sea directamente por los puertos que permitan establecer comunicación Modbus RTU o mediante dispositivos conversores, sin afectar la adquisición de datos y chequeo de errores de tipo *checksum* o *Timeout* por falta de respuesta.
- **Interacción con el usuario:** Si bien muchas tarjetas de comunicación no permiten conocer el estado de la transmisión de datos, se plantea utilizar medios de visualización adicionales, por ejemplo, indicadores led como en los conectores RJ45 cuyo parpadeo a gran velocidad indica la transmisión y recepción de información, además de una pantalla que permita conocer el valor de la señal adquirida y otras configuraciones propias del protocolo.

2.2 Selección de equipos y herramientas

Una vez establecidas las necesidades del prototipo, se presenta a continuación, la selección de componentes que podrían ser integradas y que supongan una alternativa viable, escalable y accesible que cumpla con los objetivos propuestos. A nivel de hardware, para la alimentación se plantean utilizar valores DC por lo cual se proponen las siguientes alternativas.

Regulador lineal de voltaje

Un regulador de voltaje lineal positivo es un dispositivo electrónico diseñado para mantener una salida de voltaje constante y estable desde un voltaje variable, mayor que el de la salida. Este tipo de regulador, como el que se muestra en la figura 2.1, se utiliza a menudo en aplicaciones donde se requiere energía precisa y confiable, como circuitos analógicos, sistemas de comunicación, y equipos electrónicos sensibles. Estos reguladores tienen un diseño simple, bajo ruido de salida y capacidad para proteger circuitos contra sobrecargas y cortocircuitos [22].

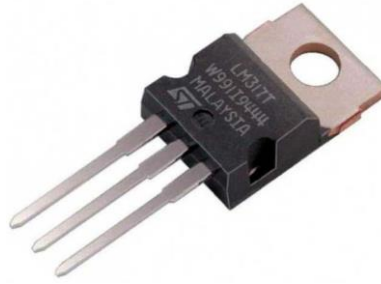


Figura 2.1. Regulador lineal de voltaje positivo [22]

Convertidores Buck

Un convertidor reductor, es un tipo de convertidor de potencia que disminuye un voltaje de entrada a un nivel más bajo de voltaje de salida. Este dispositivo es esencial en aplicaciones donde se necesita un voltaje de salida inferior al voltaje de entrada disponible, como en la regulación de voltaje para microcontroladores, circuitos integrados y dispositivos portátiles. Estos convertidores se caracterizan por su alta eficiencia y capacidad de proporcionar un voltaje de salida estable y ajustable [23].

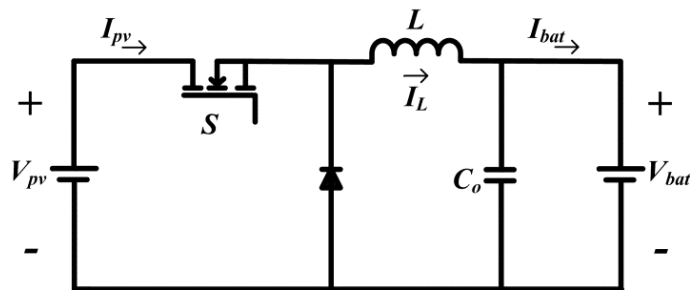


Figura 2.2. Convertidor Buck [23]

Ahora, considerando el microcontrolador que armará la trama Modbus RTU se tiene:

ATMega 328P

El ATmega328P es un microcontrolador de 8 bits basado en la arquitectura AVR. Debido al equilibrio entre funcionalidad y simplicidad, este microcontrolador es ampliamente conocido y utilizado en proyectos electrónicos y sistemas integrados. Tiene 32 KB de memoria flash programable, 1 KB de EEPROM y 2 KB de SRAM, lo que proporciona suficiente espacio para almacenar y ejecutar programas complejos [24].

Opera a frecuencias de reloj de hasta 20 MHz y tiene 23 líneas de E/S digitales (entrada/salida), 6 de las cuales pueden usarse como salidas PWM (modulación de ancho de pulso) y 6 como entradas analógicas. Además, integra funciones avanzadas como un convertidor analógico a digital (ADC) de 10 bits, comunicación serie a través de UART, SPI e I2C y múltiples modos de ahorro de energía, lo que lo hace ideal para aplicaciones de bajo consumo [25].

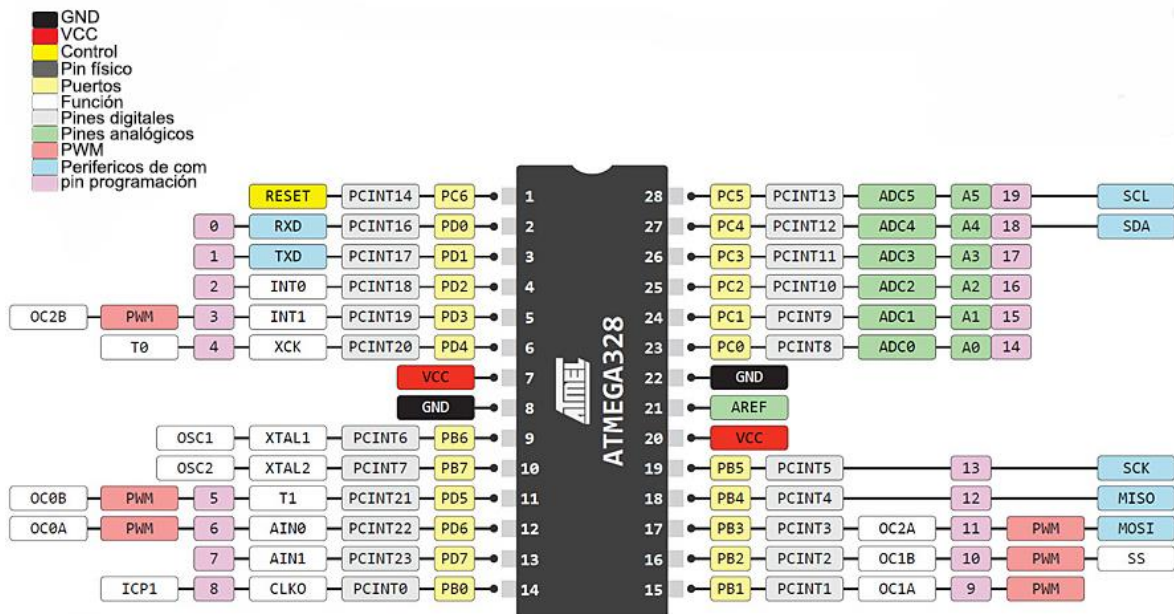


Figura 2.3. Distribución de pines del ATmega 328P [25]

Placa de desarrollo ESP32

La ESP32-WROOM es un módulo de microcontrolador que proporciona conectividad inalámbrica y capacidades de procesamiento, se basa en el chip ESP32 e incluye un procesador de doble núcleo que funciona hasta 240 MHz, lo que proporciona un gran rendimiento para tareas complejas. La ESP32 cuenta con 520 KB SRAM y 4 MB de memoria flash, que pueden almacenar grandes cantidades de programas y datos. El módulo cuenta con conectividad inalámbrica Wi-Fi 802.11 b/g/n y Bluetooth 4.2, para aplicaciones de Internet de las cosas (IoT), redes de sensores y dispositivos portátiles. También posee interfaces de E/S, como GPIO, ADC, DAC, SPI, I2C, UART y PWM. Su bajo consumo de energía lo hacen adecuado para aplicaciones que requieren eficiencia energética [26].

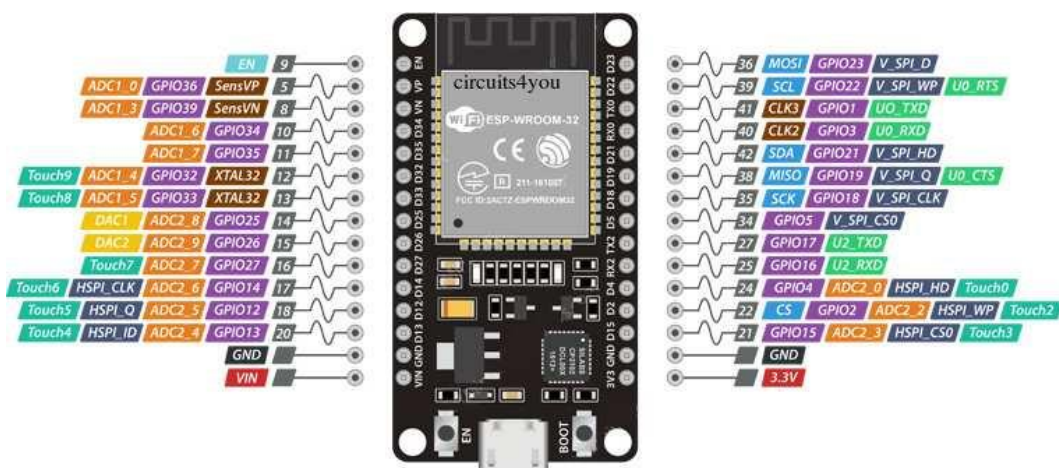


Figura 2.4. Distribución de pines de la ESP32 [26]

Dentro de los módulos para manejar la interfaz de comunicación RS-485, se plantean los siguientes circuitos integrados:

Módulo Max-485

El módulo MAX485 es un transceptor de comunicación que se utiliza para implementar la comunicación serial RS-485, su capacidad para transmitir datos a largas distancias y en entornos con ruido eléctrico, permite la transmisión y recepción de datos diferenciales, lo que mejora la inmunidad al ruido y permite la comunicación bidireccional sobre un solo par de cables. Este módulo es capaz de operar a velocidades de hasta 2.5 Mbps y soporta hasta 32 dispositivos conectados en un mismo bus de comunicación [27].

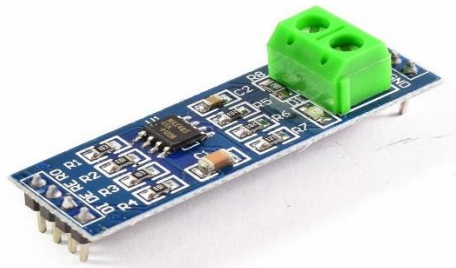


Figura 2.5. Módulo Max 485 [27]

SN75176

El SN75176 es un transceptor de comunicación diseñado para implementar los estándares RS-485 y RS-422, utilizados en sistemas industriales y de automatización para la transmisión de datos a largas distancias y en entornos con interferencias eléctricas. Este dispositivo permite la comunicación diferencial bidireccional sobre un par de cables, mejorando la robustez y la inmunidad al ruido.

La distancia máxima de transmisión depende de la velocidad de datos y la calidad del cableado utilizado, siendo recomendable el uso de cables de par trenzado y blindaje para optimizar el rendimiento en condiciones adversas [28].

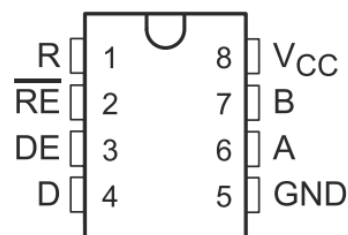


Figura 2.6. Circuito integrado SN75176 [28]

Por último, en la etapa de hardware, se considerará el uso de elementos de visualización dentro de los cuales se destaca:

LCD Oled 128*64

La pantalla OLED de 0,96 pulgadas es una pantalla pequeña y eficiente que se utiliza para mostrar imágenes y texto en dispositivos electrónicos. Puede producir colores brillantes y negros oscuros, y se conecta fácilmente a otros dispositivos como computadoras. Es ideal para cosas como relojes inteligentes, dispositivos domésticos inteligentes y cualquier otro dispositivo que necesite una pantalla pequeña con buena iluminación [29].

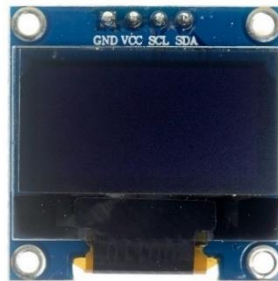


Figura 2.7. Pantalla Oled 128*64 [29]

LCD 16*2

La pantalla LCD de 16x2 es una pantalla que muestra palabras y números en dispositivos electrónicos. Puede mostrar hasta 16 caracteres (letras o números) por línea y posee dos líneas para mostrar información. Esta pantalla utiliza niveles de energía para los diferentes caracteres que se visualicen en ella. Son altamente recomendadas porque no consumen mucha energía y pueden funcionar fácilmente con otros componentes electrónicos. Aunque no pueden mostrar imágenes sofisticadas, son excelentes para mostrar información importante con claridad, como alarmas, configuraciones, etc [30].

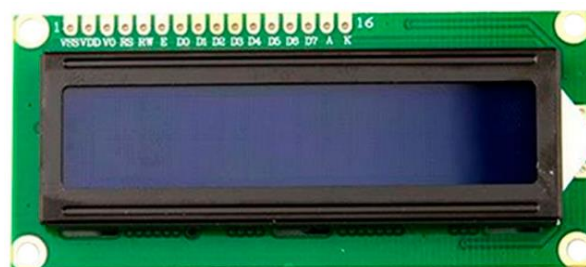


Figura 2.8. Pantalla Oled 128*64 [30]

Ahora bien, para solventar los requerimientos de software se utilizarán programas para la codificación del microcontrolador con el que se trabaje, si se opta por el AVR se puede optar por los softwares Microchip Studio o Arduino IDE, por otro lado, si se opta por la tarjeta de desarrollo ESP32 se utilizará Arduino IDE o Thonny para Microphyton. Además, se debe considerar programas para la elaboración de placas y esquemas electrónicos como lo son Tinkercad, Proteus, Eagle, Fritzing, entre otros.

2.3 Diseño del prototipo

En este subcapítulo se detallarán los procedimientos seguidos para el desarrollo propuesto, al ser un prototipo de tarjeta de comunicación se enmarca en una metodología experimental. Con los elementos descritos en el subcapítulo anterior y añadiendo elementos de protección y para el correcto funcionamiento de los componentes, se podrá establecer comunicación basada en el protocolo Modbus RTU y todo lo que conlleva, como su interfaz y características tanto lógicas como físicas.

A partir de la metodología planteada, el proyecto se divide en varias etapas:

- Etapa de alimentación
- Etapa de adquisición
- Etapa de control
- Etapa de visualización
- Etapa de configuración
- Etapa de comunicación

Cada una de las etapas aquí mencionadas se detallarán en el orden que fueron realizadas durante el desarrollo del proyecto, debidamente justificadas y con las bases teóricas sobre la elección o cambio de componentes y los criterios escogidos para dichas selecciones.

2.3.1. Etapa de Alimentación

Una buena práctica en el desarrollo de proyectos electrónicos es considerar el tipo de alimentación requerida, para ello se deben contemplar particularidades como; la demanda energética total del proyecto y de los diferentes elementos por separado, la aplicación, el lugar, entre otros. Para este caso, dado que el proyecto tiene como destino final el laboratorio de redes industriales de la Escuela Politécnica Nacional cuya aplicación servirá para las diferentes prácticas asociadas a la red de controladores Modbus, se utilizará una alimentación de $24V_{DC}$ ya que este es el potencial comúnmente usado para alimentar varios de los PLC's de dicho laboratorio, adicionalmente es importante resaltar que el valor de alimentación escogido es un valor estandarizado para tableros industriales.

El laboratorio cuenta ya con módulos cuya salida es $24V_{DC}$, pero los componentes electrónicos que se utilizarán funcionan a niveles lógicos de dispositivos TTL, es decir a $5V_{DC}$, así la selección de un regulador de tensión se detalla en el siguiente apartado.

2.3.1.1. Selección del regulador de voltaje:

Inicialmente se propuso usar un único regulador que reduzca el voltaje de 24V a 5V, para ello, investigando sobre los diferentes componentes que existen y que suponen una ventaja económica se propuso usar el regulador de voltaje LM7805, este regulador es un circuito integrado que mantiene un voltaje fijo en su salida, independientemente del voltaje aplicado en la entrada y no conlleva elementos adicionales.

Entre las características relevantes de este regulador, se tiene su salida fija de 5V con corriente máxima de hasta 1A, un encapsulado tipo TO220, con un rango de alimentación a su entrada de 2 a 35 V_{DC}. Además, posee un sistema de protección ya que internamente cuenta con un limitador de corriente contra cortocircuitos y un limitador de temperatura, mismo que puede reducir el nivel de corriente suministrado.

Según la hoja de datos proporcionada por uno de los fabricantes de este tipo de reguladores, la conexión para tener una salida fija a 5V sugerida se muestra en la figura 2.10.

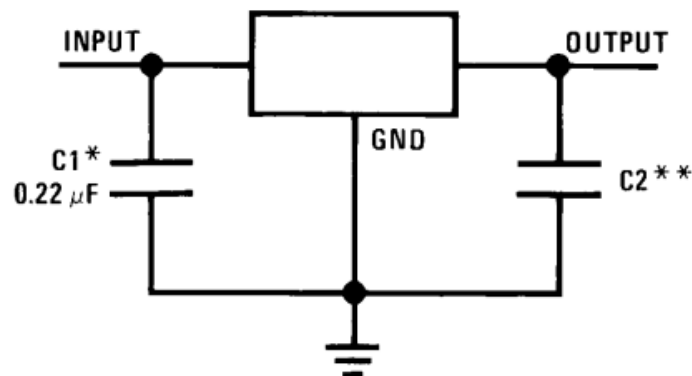


Figura 2.10. Esquema para regulador de voltaje de salida fija

Como se puede observar en la figura 2.10, el fabricante especifica el valor de capacitancia de entrada, pero no de salida, sin embargo, se indica que, si bien el circuito puede prescindir del condensador de salida para su funcionamiento, se recomienda incluir un capacitor de 0.1μF para la estabilidad ante posibles transitorios en la respuesta del circuito integrado.

2.3.1.2. Diseño de la etapa de alimentación

A partir de las sugerencias del fabricante, se diseñó la etapa de entrada del circuito, incluyendo elementos pasivos pensando en la seguridad del usuario a utilizar el prototipo y en la seguridad de cada componente que lo integra, de esta forma la figura 2.11 ilustra la etapa de alimentación que fue implementada.

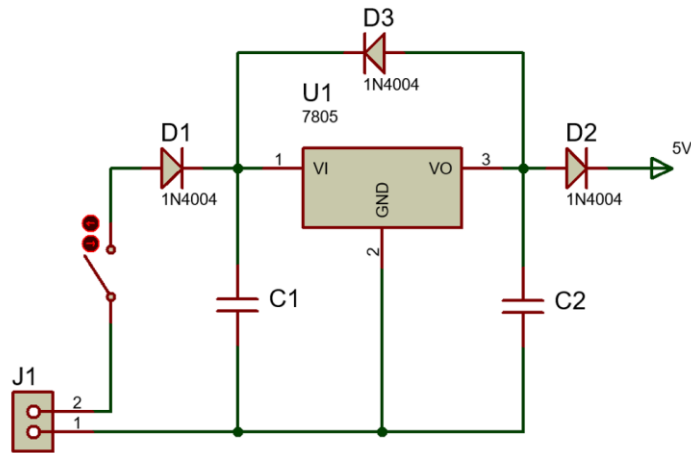


Figura 2.11. Etapa inicial de alimentación.

El esquema de la figura 2.11, representa la etapa de alimentación que inicia con un conector de dos borneras donde se alimentará a 24V todo el prototipo, después se añade un interruptor que permitirá el paso de energía hacia el circuito integrado, en serie al interruptor se integran a la entrada y a la salida del regulador diodos en cada pin, para prevenir una incorrecta polarización del circuito, así, si la alimentación en las borneras se conecta inversamente al circuito los diodos no permitirán el paso de energía al resto de componentes.

Por facilidad de implementación, ensamblaje y ventajas económicas, se generalizan todos los diodos al modelo 1N4004, se escoge este diodo ya que se ajusta a las necesidades de la etapa de diseño, siendo así la principal característica la corriente directa de 1A, la máxima corriente que puede proporcionar el regulador de voltaje.

Durante esta primera etapa, es evidente la necesidad de calcular la potencia de disipación del circuito integrado LM7805, pese a que tiene una protección térmica que reduce el suministro de corriente proporcionado por dicho circuito integrado, este es susceptible a disipar calor de forma poco conveniente, por lo que se necesita el cálculo de disipación del regulador mencionado, si se considera la ecuación referente a potencia se tiene:

$$P_D = \Delta V \cdot I [W]$$

Ecuación 2.1. Potencia de Disipación

Donde

- P_D : Potencia Disipada
- ΔV : Variación de voltaje
- I: Corriente

Si se toma el voltaje de entrada igual a 24 V_{DC}, el voltaje de salida 5 V_{DC} y asumiendo el caso donde la demanda de corriente llegue al límite de 1A, se tiene:

$$P_D = (24 - 5) \cdot (1) = 19 [W]$$

Cabe mencionar que la potencia disipada en forma de calor es extraída de las juntas hacia el empaquetado del regulador, y posteriormente del empaquetado al ambiente. Considerando que el regulador operará al aire libre, es decir, sin dispositivos adicionales para su enfriamiento o métodos de ventilación forzada, la ecuación que describe dicho fenómeno en estado estable es:

$$T_J = T_A + \theta_{JA} \cdot P_D$$

Ecuación 2.2. Relación entre temperatura y potencia disipada

Donde:

T_J : Temperatura de junta

T_A : Temperatura ambiente

θ_{JA} : Resistencia térmica junta – ambiente

Dependiendo el fabricante los valores de resistencia térmica junta-carcasa y junta-ambiente varía, pero teniendo un valor aproximado común entre fabricantes, se puede establecer en 5 y 50°C/W respectivamente, considerando una temperatura ambiente de 25°C se tiene:

$$T_J = 25 + (55) \cdot 19$$

$$T_J = 1070 \text{ } ^\circ\text{C}$$

El valor de temperatura indicado es irreal, sugiriendo que el circuito integrado se quemaría bajo los parámetros establecidos. Sin embargo, en la práctica, el circuito podría funcionar, aunque con una elevada disipación de calor, lo que reduciría significativamente su vida útil si se opera a esa temperatura durante períodos prolongados. Además, la revisión de la hoja técnica muestra que la prueba del circuito integrado se realizó con un voltaje de entrada de 10V, resultando en una potencia disipada de 5W, aproximadamente 3 veces menor que la calculada.

En función de los test de prueba de los diferentes circuitos integrados surge la necesidad de incluir una etapa adicional de regulación de voltaje, teniendo como finalidad que la diferencia entre voltaje de entrada y salida sea cercana a la diferencia entrada-salida con los valores para los que fueron testeados los circuitos integrados. De manera similar a la etapa anterior, es necesario seleccionar el regulador para la etapa de alimentación. Se utilizará otro regulador de voltaje, como se muestra en la figura 2.12.

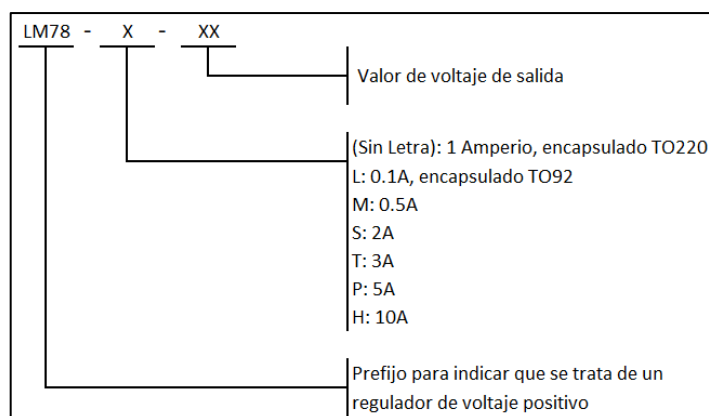


Figura 2.12. Especificaciones para la selección de reguladores de voltaje

Según la figura 2.12, además del regulador LM7805, se incluirá el LM7812 en el diseño. Este regulador proporciona un voltaje de 12VDC, cercano al valor de entrada para el cual fue probado, y puede ser útil para futuras mejoras o adaptaciones, como la inclusión de amplificadores operacionales u otros componentes que requieran ese voltaje estandarizado. La potencia disipada se cuantifica en la tabla 2.1, basada en los reguladores utilizados.

Tabla 2.1. Comparación de potencia disipada teórica vs aplicada con un solo regulador

	V_{in}	V_{out}	P_D
Valor Teórico	10	5	<15 W
Valor Aplicado	24	5	19 W

Por otro lado, al utilizar dos circuitos integrados es lógico esperar que la potencia de disipación se divida entre los dos reguladores, lo cual se evidencia en la tabla 2.2, dicha tabla muestra que en ambos circuitos integrados no se excede la potencia de disipación.

Tabla 2.2. Comparación de potencia disipada teórica vs aplicada con dos reguladores

		V_{in}	V_{out}	P_D
Valor Teórico	LM7812	19	12	<15 W
	LM7805	10	5	<15 W
Valor Aplicado	LM7812	24	12	12 W
	LM7805	12	5	7 W

El valor de potencia disipada teórica que se muestran en las tablas 2.2 y 2.1 son datos proporcionados por la hoja de datos de cualquier fabricante, de forma particular, utilizando la hoja de datos del fabricante Fairchild Semiconductor muestra, entre otras cosas, el valor de voltaje en la salida que se muestran en la figura 2.13 para el LM7805 y en la figura 2.14 para el LM7812.

Parameter	Symbol	Conditions	MC7805/LM7805			Unit
			Min.	Typ.	Max.	
Output Voltage	V _O	T _J = +25 °C	4.8	5.0	5.2	V
		5.0mA ≤ I _O ≤ 1.0A, P _O ≤ 15W V _I = 7V to 20V	4.75	5.0	5.25	

Figura 2.13. Voltaje de salida en el regulador lineal LM7805 [31]

Como se puede observar en esta figura, dentro de las condiciones para tener una salida de 5V_{DC} es que la potencia de disipación sea menor a 15W, para ello es mandatorio que el voltaje de entrada esté entre 7 a 20V_{DC}, por lo cual se sustenta y se reafirma la imposibilidad de utilizar únicamente este regulador de voltaje. Por otro lado, considerando ahora el segundo regulador de voltaje.

Characteristic	Symbol	Test Conditions	LM7812I			LM7812			Unit
			Min	Typ	Max	Min	Typ	Max	
Output Voltage	V _O	T _J = +25 °C	11.5	12	12.5	11.5	12	12.5	V
		5.0mA ≤ I _O ≤ 1.0A, P _D ≤ 15W V _I = 14.5V to 27V				11.4	12	12.6	
		V _I = 15.5V to 27V	11.4	12	12.6				

Figura 2.14. Voltaje de salida en el regulador lineal LM7812 [31]

Para este regulador, el voltaje de entrada escogido si satisface la condición del rango admitido de voltaje de entrada, y con lo expuesto en la tabla 2.2 se evidencia el cumplimiento de la condición para la potencia de disipación.

Una vez establecida la base teórica sobre la cual se incluye un segundo regulador de voltaje, se presenta el esquema final de esta etapa, en la cual a la salida se incluye una resistencia y un led como indicador de encendido para el usuario, de igual manera que el esquema presentado en la figura 2.11, el regulador contará con los capacitores en sus pines de entrada y el arreglo de diodos previamente mencionados, obteniendo así lo que se evidencia en la figura 2.15.

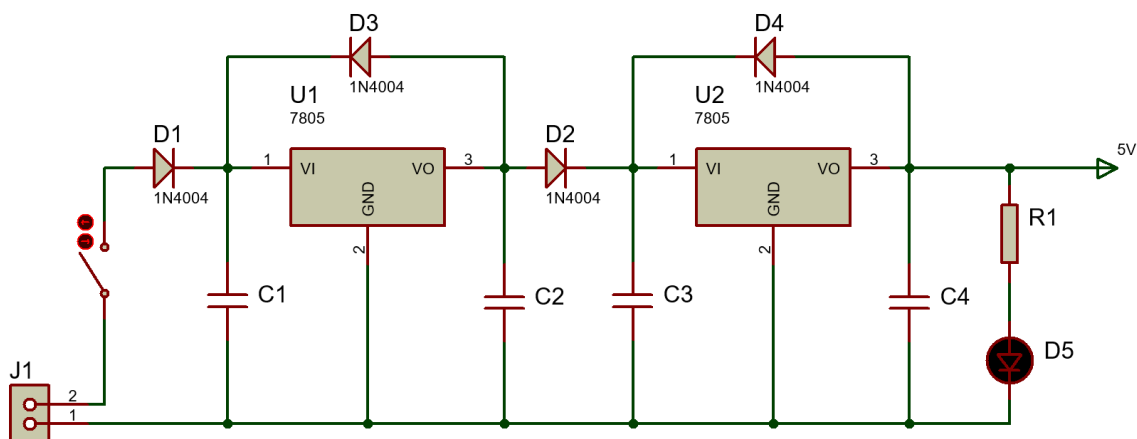


Figura 2.15. Etapa final de alimentación.

2.3.2. Etapa de adquisición

El diseño de esta etapa abarca desde la salida de voltaje del sensor (0 a 10V) hasta la entrada del convertor analógico-digital del microcontrolador. Dado que el Atmega 328P soporta un máximo de 5V en sus entradas, es necesario atenuar la señal del sensor a la mitad. Para ello, se utilizará un circuito con un diodo de polarización para asegurar la correcta circulación de energía y un potenciómetro de precisión para ajustar la relación 2:1 de voltaje mediante un divisor de tensión, como se muestra en la figura 2.16. Este elemento pasivo facilita las calibraciones en la lectura de voltaje.

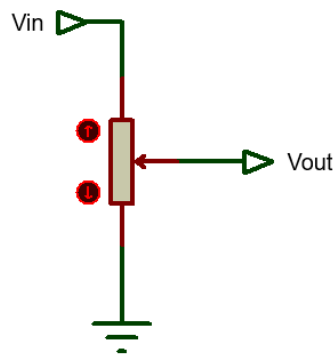


Figura 2.16. Divisor de Voltaje con potenciómetro

Un potenciómetro es un componente electrónico que actúa como una resistencia variable, tiene tres terminales, razón por la cual un divisor de voltaje utilizando un potenciómetro se basa en la resistencia variable del potenciómetro para dividir un voltaje de entrada en dos partes proporcionales, como se indica en la siguiente ecuación:

$$V_{out} = |R_T - R_{out}| \cdot V_{in}$$

Ecuación 2.3. Voltaje de salida en un divisor de tensión

Donde

V_{out} : Voltaje de Salida

V_{in} : Voltaje de Entrada

R_T : Resistencia total entre V_{in} y GND

R_{out} : Resistencia entre V_{out} y GND

Por último, este circuito integra un diodo Zener de 5.1 voltios en paralelo a la salida variable del potenciómetro y GND, este elemento tiene como única finalidad la protección del convertor analógico digital del microcontrolador, de esta forma la figura 2.17 muestra el esquema que combina los elementos antes mencionados.

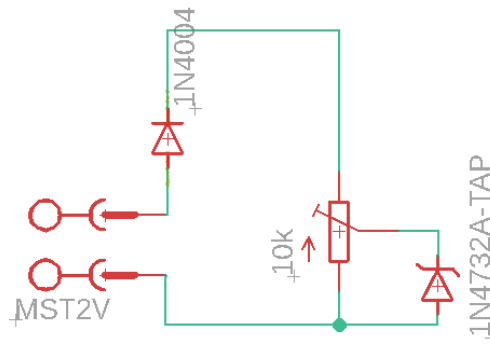


Figura 2.17. Etapa de adquisición

2.3.3. Etapa de comunicación

Esta etapa integra el módulo de comunicación MAX 485, el cual posee 8 pines como los que se ven en la figura 2.18 y su correspondencia entre los pines del microcontrolador Atmega 328P y dicho módulo se muestra en la tabla 2.3, cabe mencionar que este módulo es el que permite la conversión entre datos de nivel TTL a su equivalente al estándar RS-485, teniendo como salida del módulo, las terminales A y B que son el par diferencial de comunicación [27].

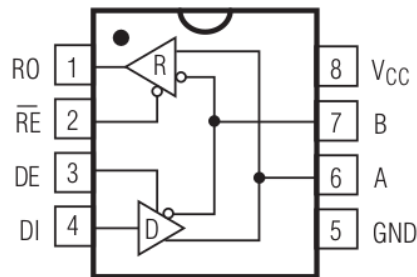


Figura 2.18. Etapa de adquisición

La correspondencia entre los pines del módulo que integra al transceptor de comunicación y el microcontrolador ATmega 328P, se muestra a continuación.

Tabla 2.3. Correspondencia de pines entre Arduino y Max 328P

Módulo MAX 485	ATMega 328P	Descripción
RO	RX	Recepción
DE	PIN 2	Pin de control para habilitar y deshabilitar transmisión y recepción
RE		
DI	TX	Transmisión
VCC	+5V	Alimentación
A	-	Línea diferencial
B	-	Línea diferencial
GND	GND	Tierra

Una acotación importante surge debido al esquema del módulo MAX 485, este módulo ya integra una resistencia de 120Ω entre las líneas diferenciales A y B como se muestra en la figura 2.19, esta resistencia, como ya se revisó en el capítulo uno, tiene como funciones la reducción de interferencias de alta frecuencia, pérdidas en los conductores, problemas de desacople y reflexión en la comunicación.

Si bien es cierto que este elemento tiene protagonismo cuando la red se extiende distancias más allá de los 100m y opera a velocidades mayores a 19200 bps, por facilidad en la integración del prototipo con la red Modbus RTU con la que cuenta el laboratorio, la resistencia en mención debe ser removida para no inhabilitar el funcionamiento y comunicación de los demás dispositivos que pertenecen a la red.

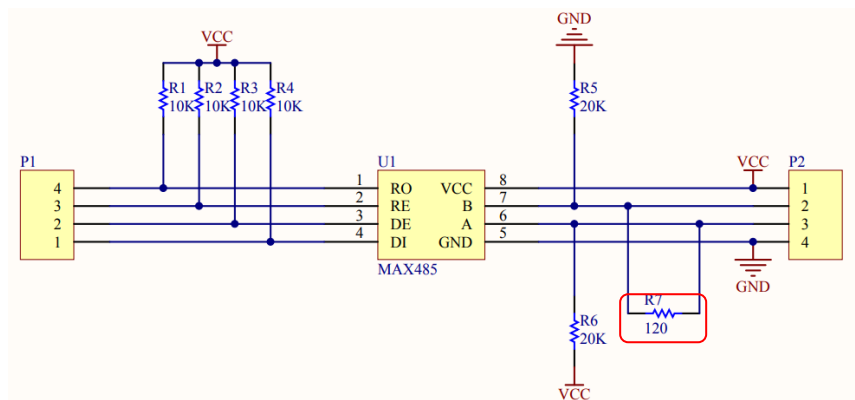


Figura 2.19. Esquema del módulo Max 485 [27]

La remoción de la resistencia en el módulo se muestra en la figura 2.20.

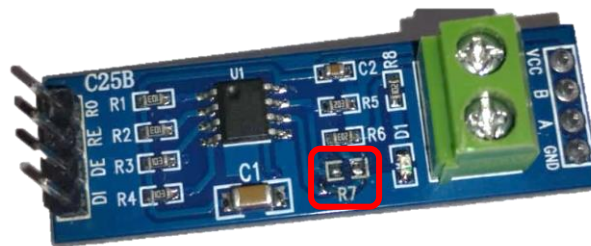


Figura 2.20. Módulo Max 485 sin resistencia terminadora

2.3.4. Etapa de visualización

Para esta etapa se va a integrar, por su versatilidad y menor cantidad de pines, una pantalla Oled de 128×64 pixeles puesto que es ideal para aplicaciones de bajo consumo energético. Se escoge esta pantalla, ya que a nivel de programación se pueden proyectar mayor cantidad de datos con un menor consumo de recursos, resultando en un ahorro significativo en las líneas de código y recursos de procesamiento del microcontrolador [29].

Esta pantalla utiliza la interfaz de comunicación I2C por lo cual requiere únicamente dos pines, SDA (Datos del sistema) y SCL (Reloj del sistema), además de dos pines para alimentación. La esquematización de esta etapa constituida por un conjunto de 4 pines; 5V_{DC} y GND, proporcionados por la fuente de alimentación, y los dos pines del bus de comunicación que se conectan al microcontrolador, se muestra en la figura 2.21.

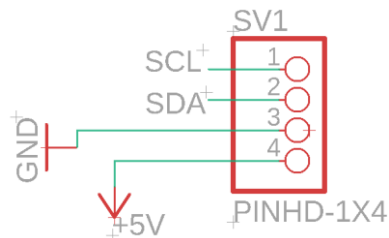


Figura 2.21. Esquema de la etapa de visualización

2.3.5. Etapa de configuración

Al plantear un prototipo que sea amigable con el usuario capaz de permitirle conocer parámetros importantes en el intercambio de información, y de forma particular, direcciones Modbus y los datos que se almacenen en estas, se propone la integración de tres botones que servirán para navegación y modificación de parámetros del prototipo. Uno de los botones cumplirá la función de acceso a un menú de parámetros configurables, mientras que los otros botones cumplirán funciones de desplazamiento y de aumento, disminución y selección de valores en los parámetros configurables.

De esta forma, el prototipo no solo permite al usuario el intercambio de información, sino que también le permite de forma rápida y eficaz la modificación de registros Modbus, sin la necesidad de modificar la programación del prototipo volviéndola transparente al usuario, dentro de la etapa de configuración, y según las necesidades del protocolo, se plantea los siguientes parámetros como configurables:

- **Esclavo ID:** Este campo efectuará cambios en el identificador que tendrá el prototipo en la red Modbus rtu.
- **Input Register address:** Este campo contendrá la dirección donde se almacena la conversión analógica digital del valor de voltaje que será enviada hacia el maestro
- **Holdig Register address:** Este campo apunta a la dirección donde se almacena los registros de retención enviados por el maestro al prototipo.
- **Baud Rate:** Este campo permitirá la configuración de la velocidad de transmisión serial, se prevé que las opciones más comunes son: 4800, 9600, 19200, 38400.

2.3.6. Etapa de control

Hardware

Esta etapa cohesiona el control de los diferentes periféricos antes mencionados, como el módulo MAX485, la pantalla Oled y botones de configuración, y la programación necesaria para la generación de la trama y los requerimientos que harán posible el intercambio de información con otros dispositivos mediante el uso del protocolo Modbus RTU.

De esta forma, la figura 2.22 muestra el conexionado físico de los diferentes elementos con los pines del microcontrolador ATmega 328P.

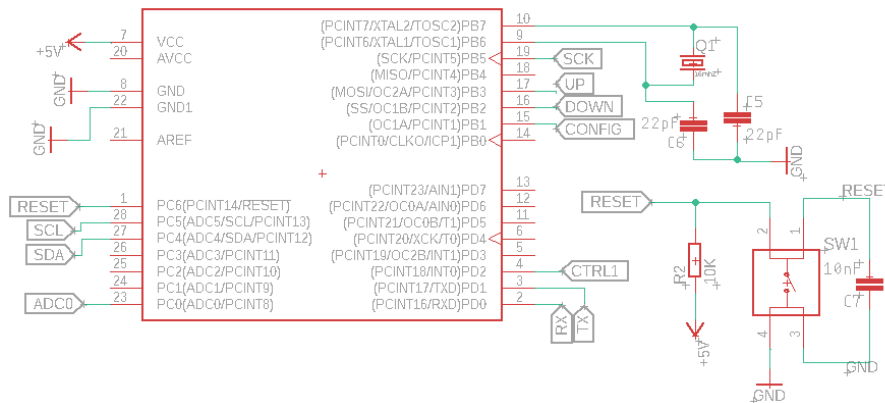


Figura 2.22. Conexionado del microcontrolador ATmega 328P

Es evidente que el microcontrolador contiene más conexiones de las que se han justificado, no obstante, la tabla 2.4 contiene una breve descripción de la etiqueta asociada al pin del microcontrolador y el componente con el que se conecta, emulando un mapa de memoria.

Tabla 2.4. Relación de los pines del microcontrolador

Etiqueta	Etapa	Descripción
5V _{DC}	Alimentación	Pin de alimentación de 5V _{DC}
GND		Pin de tierra
SCL	Visualización	Pin de reloj para comunicación I2C
SDA		Pin de datos del sistema para comunicación I2C
ADC0	Adquisición	Pin al cual llega el valor de 0 a 5V para su posterior conversión digital
UP	Configuración	Pines para navegación/modificación de parámetros
DOWN		
CONFIG		Pin para acceso al menú de parámetros configurables
CTRL1	Comunicación	Pin para habilitación de transmisión o recepción del módulo Max485
TX		Pin para transmisión de datos
RX		Pin para recepción de datos
SCK		Indicador de programación

Adicionalmente, el esquema muestra algunas conexiones adicionales, estas condiciones son producto de buenas prácticas en la elaboración de circuitos embebidos, de esta forma se incluye un oscilador externo y un circuito de reseteo. La inclusión de un oscilador externo resulta crucial en este tipo de prototipos debido a que maneja, en primera instancia, comunicación serial. Por esta razón, requiere una mayor precisión en la frecuencia de su reloj, además, estos componentes son significativamente más estables a lo largo de un rango de temperaturas y voltajes de operación.

Por último, si bien la programación del prototipo no será de conocimiento del usuario, dicha codificación se plantea mediante el uso de la tarjeta embebida Arduino UNO, esta tarjeta posee un oscilador externo de 16MHz, razón por la cual el microcontrolador debe igualar esta frecuencia, caso contrario la sincronización de datos resultará fallida, la inclusión de los componentes en mención se evidencia en la figura 2.23.

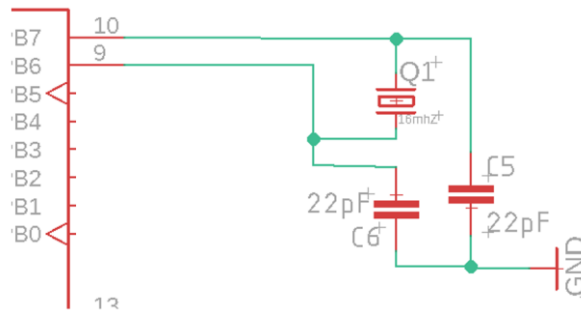


Figura 2.23. Circuito de oscilación externa

El circuito de reseteo será implementado con componentes pasivas, de acuerdo con los esquemas típicos como el que se muestra en la figura 2.24, el pin de reseteo será configurado ya en la programación como un pin de entrada con pull-up activo, de forma que con una resistencia de 10kΩ se limita el paso de corriente hacia el pin de reseteo y se evitan cortocircuitos con la alimentación del esquema, mientras que el botón asociado sirve para cerrar el circuito, adicionalmente se incluye un capacitor de 10nF en paralelo al pulsante para evitar rebotes en la señal de reseteo, de la siguiente forma.

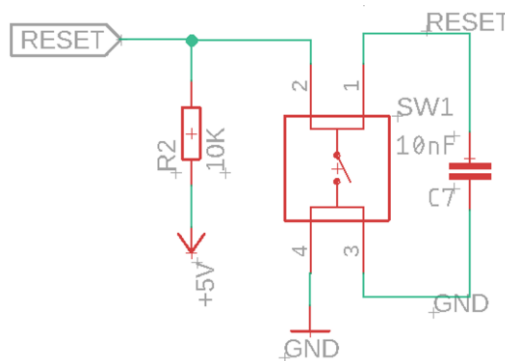


Figura 2.24. Circuito de reseteo

Software

Ahora bien, para la parte de software debido a la integración de los periféricos se sugiere el trabajo por subrutinas, ya que al manejar buses de comunicación distintos como lo es el bus UART y el bus I2C, el microcontrolador debe priorizar la comunicación, por ello se plantea también una programación parecida a la utilizada en máquina de estados, con banderas que permitan la relación entre subrutinas.

El programa principal busca ser relativamente corto y que este sea quien llame a las diferentes subrutinas para la integración de las diferentes componentes antes descritas, y el correcto desempeño del prototipo en la transmisión y recepción de datos, de esta manera la figura 2.25 muestra el diagrama de flujo asociado a la programación principal del microcontrolador, para tal acción se utilizará el lenguaje C que es el lenguaje admitido por el software Arduino IDE, esto ya que la programación análoga a máquina de estados utiliza sentencias if, while, switch así como las diferentes acciones asociadas al menú de parámetros configurables.

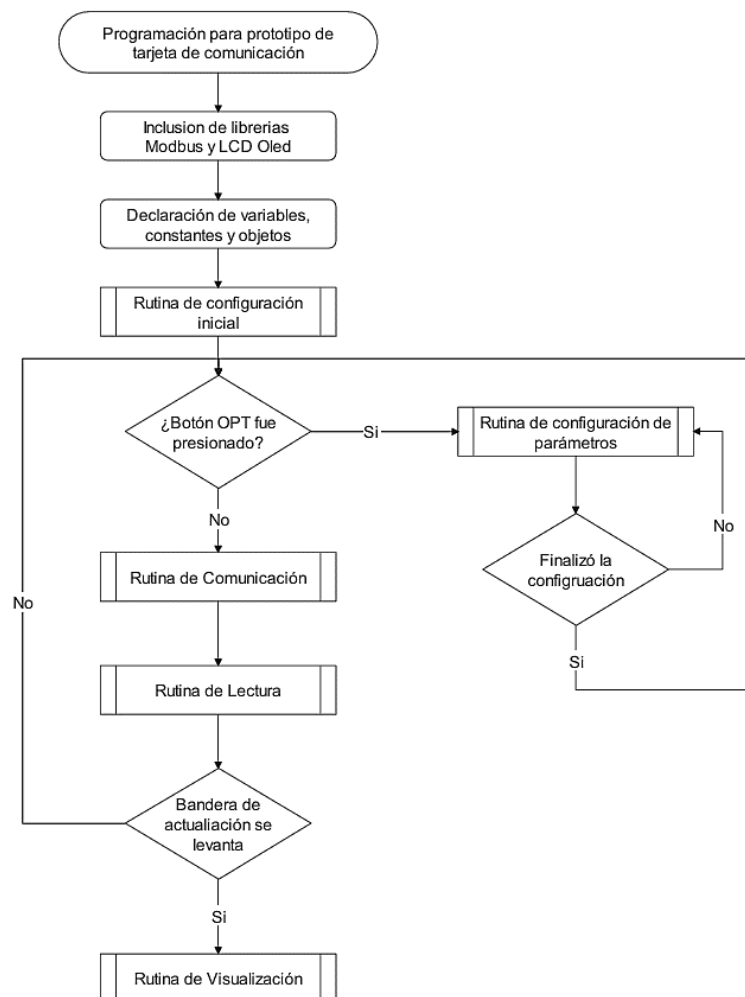


Figura 2.25. Diagrama de flujo principal del programa

Subrutina de configuración inicial

Esta subrutina se encargará de la inicialización de los objetos, particularmente del objeto Modbus que llama a la librería que se encarga del armado de la trama, así como también del objeto oled al cual se le asignan atributos mediante sus constructores. Por otro lado, en esta subrutina se inicializan los pines asociados a los botones para la navegación y configuración de parámetros, la figura 2.26 resume todo lo antes expuesto en su diagrama de flujo.

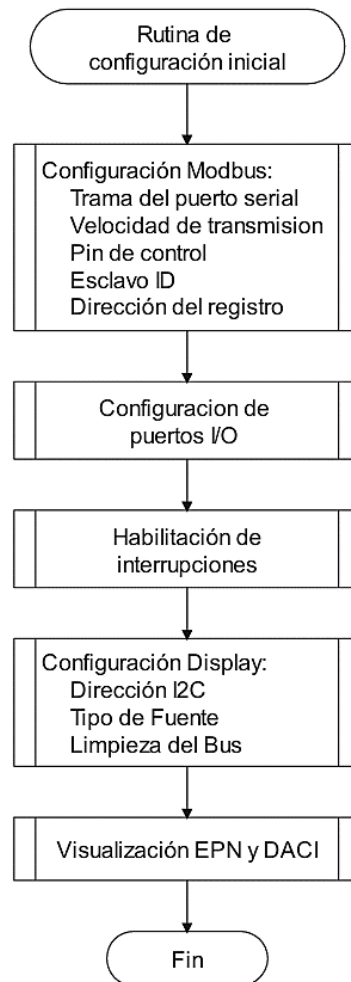


Figura 2.26. Diagrama de flujo rutina de configuración inicial

Subrutina de configuración de parámetros

Esta rutina contiene toda la lógica para el cambio de parámetros que son de interés en la comunicación y propios del protocolo Modbus RTU, de esta forma este menú permite el cambio del identificador del esclavo, la dirección del registro de entrada (Input Register) y la del registro de retención (Holding Register), así como también la velocidad de transmisión de comunicación.

Estos parámetros resultan cruciales ya que en el caso de que uno de ellos no este correctamente configurado la comunicación no podrá establecerse, la lógica de configuración de parámetros es totalmente transparente al usuario, ya que este solo interactuará a través de los botones de selección y navegación e internamente se realizarán los cambios solicitados. Esta codificación conlleva ciertas redundancias que impiden situaciones extrañas de configuración más allá de las que fueron planteadas, de esta forma se presenta la figura 2.27.

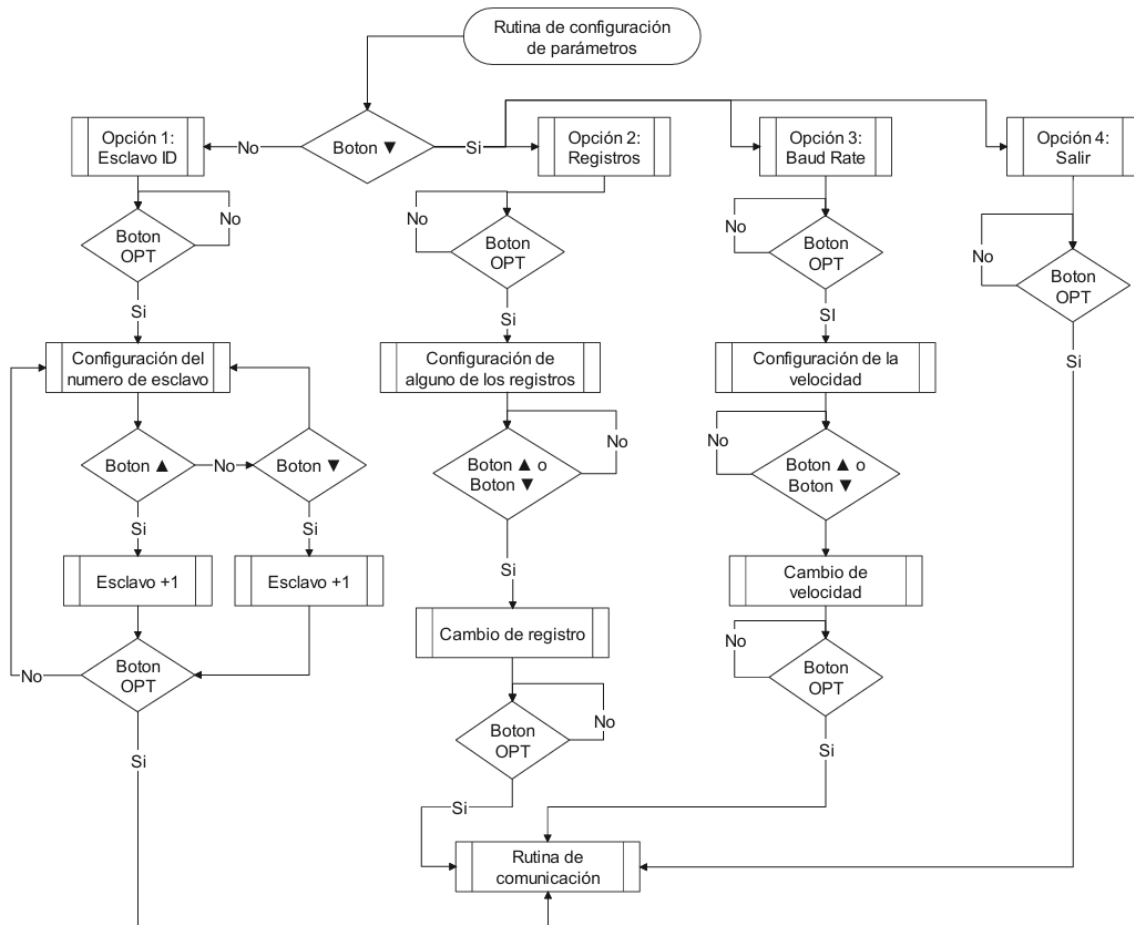


Figura 2.27. Diagrama de flujo rutina de configuración de parámetros

Subrutina de comunicación

Esta subrutina puede definirse como el corazón de este prototipo, ya que, se encarga del armado y envío de la trama con los datos de interés por parte del esclavo, y de igual manera, se encarga de la recepción y decodificación de la estructura de la solicitud enviada por el maestro. De esta forma, es preciso resaltar que se utilizará una librería desarrollada por un programador ajeno a este proyecto, con la finalidad de utilizar un recurso previamente probado y debidamente validado que satisfaga las necesidades de comunicación antes mencionadas.

La librería desarrollada por Vengen W., disponible en GitHub, mejora la librería de Andrés Sarmiento. Incluye subcarpetas para diferentes implementaciones del protocolo, como Modbus IP, Modbus UDP y Modbus Serial, siendo esta última la relevante para el prototipo. La programación de estos elementos se basa en un enfoque orientado a objetos, con dos archivos principales: un archivo de cabecera (.h) que declara atributos y métodos, y un archivo .cpp que utiliza estos atributos mediante constructores para construir la trama según la implementación del protocolo.

Cabe mencionar que cada uno de estos archivos contiene la programación necesaria para que el mensaje Modbus pueda ser receptado o transmitido de forma correcta, acatando los requerimientos del protocolo y los campos del ADU para cada implementación, es por ello, que de forma general se debe incluir un archivo principal llamado Modbus, que de igual manera tiene un archivo cabecera (.h) y un archivo de función (.cpp) tal como se muestra en la figura 2.28.

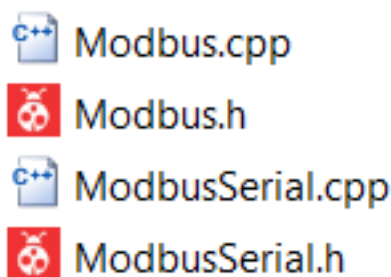


Figura 2.28. Inclusión de librerías Modbus

Es importante enfatizar en que el programa principal, desarrollado en el IDE de Arduino, no podrá funcionar si no se incluyen estos archivos en la misma ruta donde se ubica el archivo .ide, correspondiente al programa del prototipo. De esta forma, la programación realizada en esta sección, por parte del autor, se limita a la declaración de los registros de entrada y retención (input y holding register respectivamente) y a la asociación con las diferentes subrutinas anterior y posteriormente expuestas.

La figura 2.29 es una aproximación de lo que realiza esta librería para los fines pertinentes ya que la librería contiene funciones más complejas, mismas que la convierten en una poderosa y completa herramienta. No obstante, únicamente se hace uso de parámetros de lectura y escritura de registros por lo que no se esquematiza en el diagrama de flujo todas las funcionalidades, sino las que este prototipo requiere.

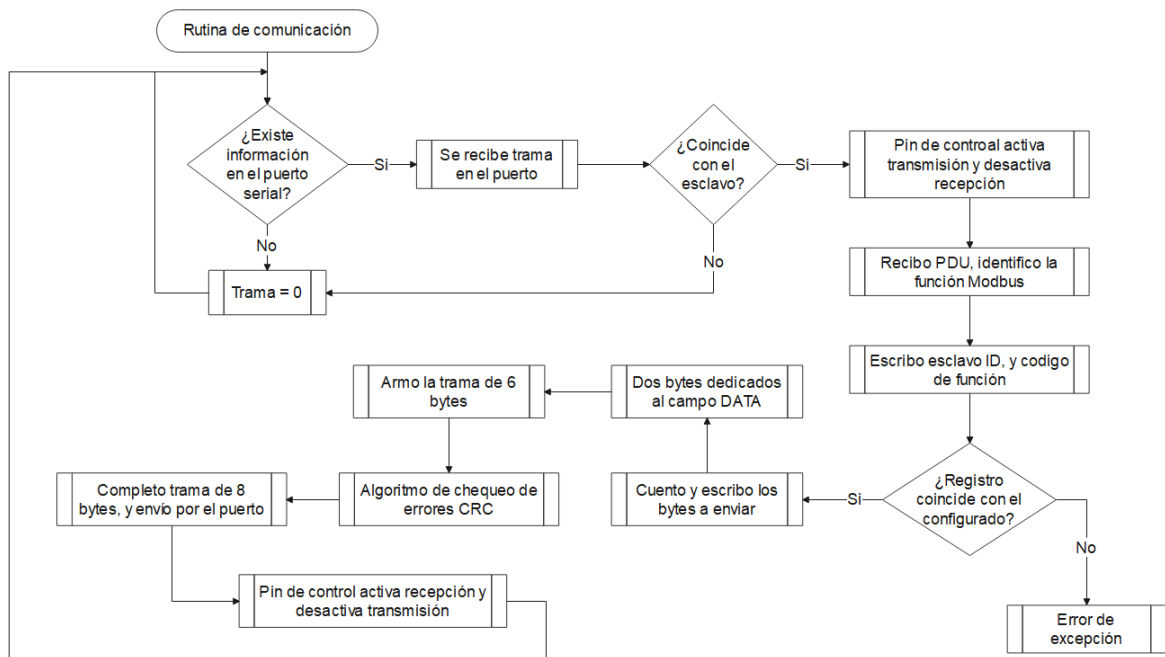


Figura 2.29. Diagrama de flujo rutina de comunicación

Subrutina de lectura

Esta rutina se encarga únicamente de la adquisición del dato del sensor analógico, en esta rutina se implementa un filtro pasa bajos por software, este filtro tiene como característica la atenuación de frecuencias altas dejando pasar así frecuencias bajas, lo que lo convierte en una gran herramienta en la medición de lecturas de sensores cuyos valores no son estables, o se encuentren en entornos poco amigables para los componentes electrónicos. Para la codificación de este filtro se utiliza la fórmula que describe la ecuación 2.4 [32].

$$Dato_{Filtrado} = (\alpha * Dato_{Leido}) + (1 - \alpha)Dato_{Filtrado}$$

Ecuación 2.4. Filtro pasa bajos por software

Donde

$Dato_{Filtrado}$: Dato que ya fue procesado

$Dato_{Leido}$: Dato que ingresa sin filtro, lectura sin filtrar

α : Es una constante con la que se determina el grado de filtrado, cuando vale 1 el dato no se filtra y cuando es 0 el dato se filtra totalmente, entre menor sea el valor de esta constante mayor será el filtrado y mejorará la estabilidad en la lectura.

De esta forma la rutina que describe este comportamiento se detalla en la figura 2.30 a continuación.

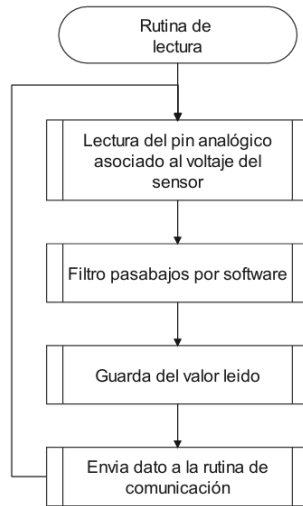


Figura 2.30. Diagrama de flujo rutina de lectura

Subrutina de visualización

Para esta rutina únicamente se maneja la proyección de caracteres en la pantalla para una mejor interacción del usuario con el prototipo, no obstante, es oportuno mencionar que debido al manejo de dos buses de datos, I2C y UART, inicialmente se tenía conflicto en la comunicación, ocasionando que a cualquier velocidad de transmisión la interacción solicitud respuesta sea de aproximadamente 5 segundos aun cuando la comunicación era 1 a 1, es decir un maestro a un solo esclavo, en este caso el prototipo, es por ello por lo que esta rutina tiene una tasa de actualización, controlada por una bandera en función de un tiempo configurable, puesto que ya se mencionó que los intereses de este prototipo recaen en la comunicación, para resumen de todo lo antes mencionado se presenta la figura 2.31.

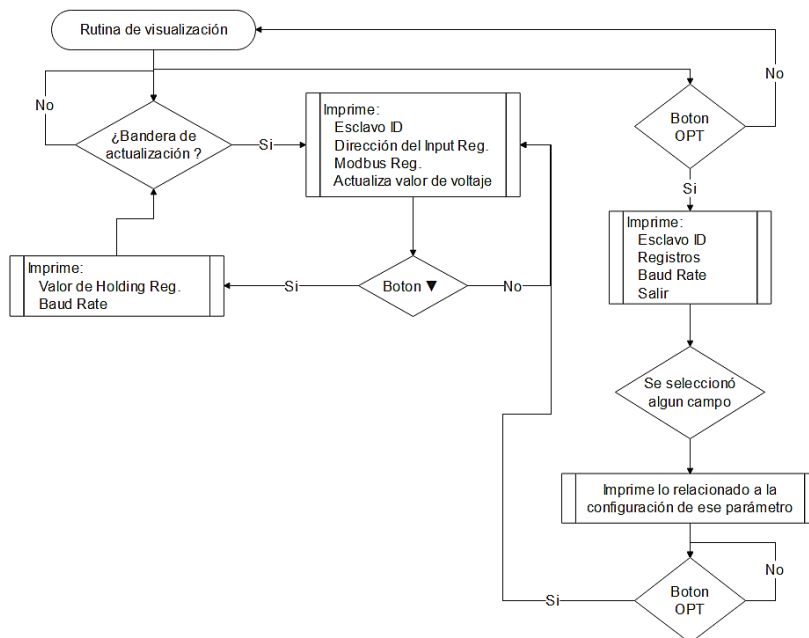


Figura 2.31. Diagrama de flujo rutina de visualización

2.4 Implementación del prototipo

Una vez detallada la esquematización por partes de todo el prototipo, se realizará su implementación de forma manual en una placa de pruebas o protoboard, con la finalidad de poder comprobar la estrecha correlación entre los diferentes componentes antes propuestos, de esta forma la figura 2.32, evidencia dicha implementación.

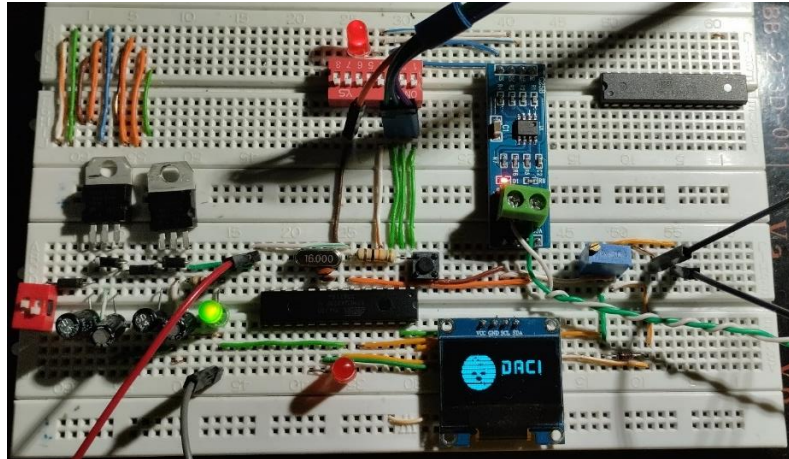


Figura 2.32. Implementación manual del prototipo

El diseño de la placa de circuito impreso PCB donde cada una de las etapas de hardware proyectadas en la figura 2.32 se cohesionarán para así poder relacionarse y satisfacer las necesidades del prototipo. De esta manera, utilizando el software de diseño Eagle se implementarán los componentes y las uniones de estos, gracias a su versatilidad y herramientas como lo son el DRC (Diálogo de revisión de reglas de diseño) y el ERC (Revisión de errores). El esquemático final se presenta en la sección de anexos como el anexo 1, mientras que el PCB implementado, se presenta en la figura 2.33.

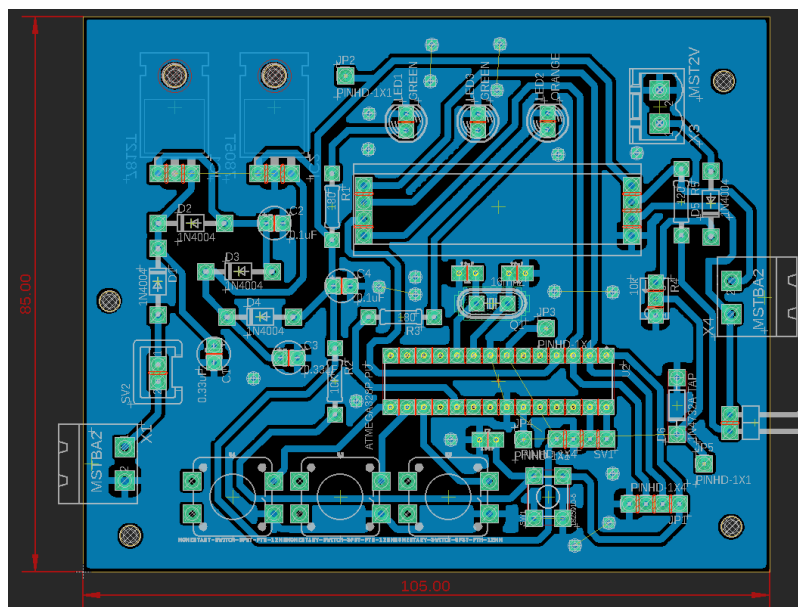


Figura 2.33. PCB diseñada

Dentro de las consideraciones más relevantes en este diseño fue el entendimiento de la funcionalidad de la malla de tierra, y la utilidad en la disipación de calor cuando los reguladores están unidos al área de cobre de la placa. Adicionalmente, por intereses económicos los diferentes periféricos se conectarán a la placa mediante pines, ya que el costo de fabricación de este tipo de placas usando una máquina CNC (máquina de control numérico por computadora) es por cm^2 , por lo que la inclusión de estos elementos repercutiría significativamente, de esta forma la figura 2.34 muestra la placa desarrollada.

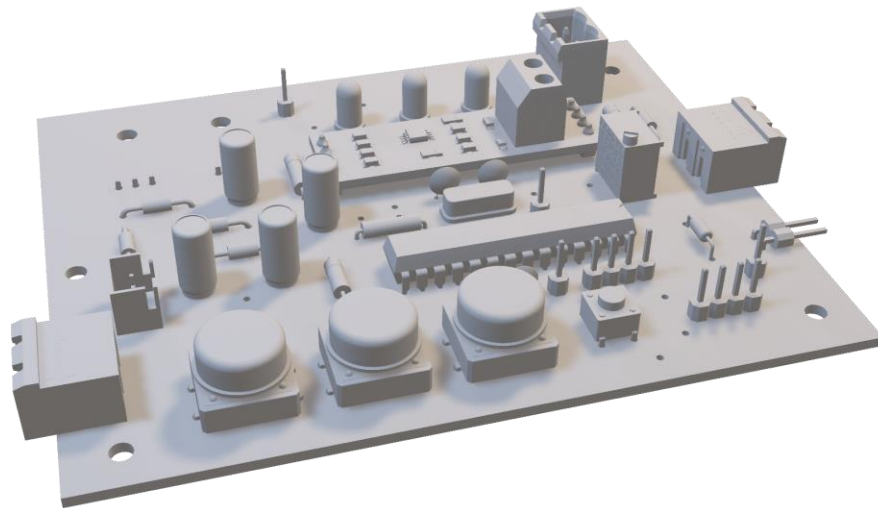


Figura 2.34. PCB diseñada

Adicionalmente en la sección de anexos, como anexo 2, se presenta una ilustración detallada de la placa y las diferentes etapas que la conforman. Otra de las ventajas por las que se diseñó con el software Eagle es su relación con el software Fusion 360, ambas de la empresa Autodesk, esta herramienta permite la virtualización 3D de los elementos, facilitando el análisis de componentes, lo que en consecuencia, permite minimizar errores en cuanto a las dimensiones estos o incluso considerar si los footprints están acorde a los componentes escogidos.

Por otro lado, si bien los intereses de este prototipo no van por el lado de diseño en 3D, es oportuno mencionar el diseño de una carcasa que encierre a la placa, esto para seguridad de los componentes ante malas prácticas por desconocimiento de operación, limitando así su mal o uso equivoco, por lo que la figura 2.35, se muestra la carcasa diseñada que contendrá la placa, esta carcasa fue diseñada en el software fusión 360 relacionándolo con las guías de la placa de la figura 2.34, de igual manera en la sección de anexos se presentan las diferentes vistas del producto final como anexo 3.

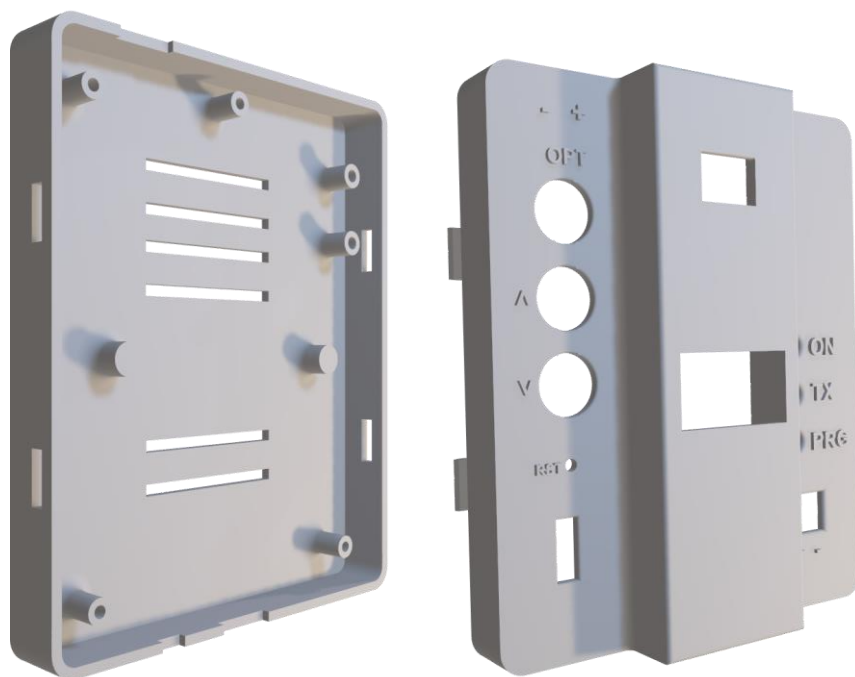


Figura 2.35. Carcasa diseñada para protección de la placa

3 RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

En este capítulo se presentará la implementación completa del prototipo desarrollado y con ello las pruebas y resultados obtenidos. Para ello se utilizará inicialmente un software que permitirá el monitoreo de la trama Modbus, tanto en la transmisión como en la recepción, y posterior a ello se realizarán pruebas del prototipo comunicándose con la red RS-485 para Modbus RTU del laboratorio al que está destinado dicho prototipo.

3.1 Resultados

En esta sección se presentarán las evidencias de funcionamiento del prototipo, como se mencionó en el capítulo 2 se priorizará la comunicación del dato analógico y en un segundo plano tendrá lugar el manejo de los periféricos. Para ello, se escalará desde la generación de la señal del mensaje hasta su comportamiento en la red propiamente.

3.1.1. Presentación del Prototipo

El prototipo final de la tarjeta de comunicación para la adquisición de una señal analógica, constituida por una PCB y una carcasa para su protección se muestra en la figura 3.1. Este prototipo permitió simplificar significativamente el espacio entre los componentes, y minimizar las interferencias electromagnéticas ocasionadas por los cables de conexión en el protoboard de la figura 2.32, convirtiendo así al prototipo en un dispositivo portable de fácil almacenamiento capaz de enviar mensajes Modbus, con su respectiva trama, a dispositivos que así lo requieran.

La figura 3.1 en la izquierda muestra la disposición de la PCB en la carcasa, mientras que a la derecha se muestra la presentación manipulable por el usuario protegiendo así los componentes electrónicos. Es oportuno mencionar que si bien, inicialmente el diseño de la carcasa estaba pensado para asegurarse mediante grapas derivadas del case como se mostró en la figura 2.35, por consideraciones ajenas al proyecto tales como: tolerancia de impresión, rigidez y dureza del material y el tipo de material propiamente, el uso de estos mecanismos fue descartado, y en su lugar se utilizaron elementos mecánicos de fijación como lo son bisagras y herrajes.



Figura 3.1. Prototipo de tarjeta de comunicación

3.1.2. Prueba de envío de mensaje

Para la verificación del correcto funcionamiento del prototipo fue necesario la solicitud de un mensaje para que el dispositivo esclavo pueda responder, de esta manera, utilizando en primera instancia un software que emula el comportamiento de un maestro Modbus RTU, mismo que se detallará en la siguiente subsección, se procedió a hacer una y múltiples solicitudes al prototipo, y mediante un osciloscopio, se evidenció la respuesta de este.

Como ya se mencionó en el capítulo uno, Modbus RTU es una trama que se maneja a niveles de bit, por lo que la respuesta tendrá una forma de onda propia del estándar RS-485 (figura 1.2), la forma para poder visualizar el mensaje en el osciloscopio es conectando la terminal positiva del osciloscopio a la línea diferencial A y en consecuencia la terminal negativa a la línea diferencial B, obteniendo así lo que se muestra en la figura 3.2.

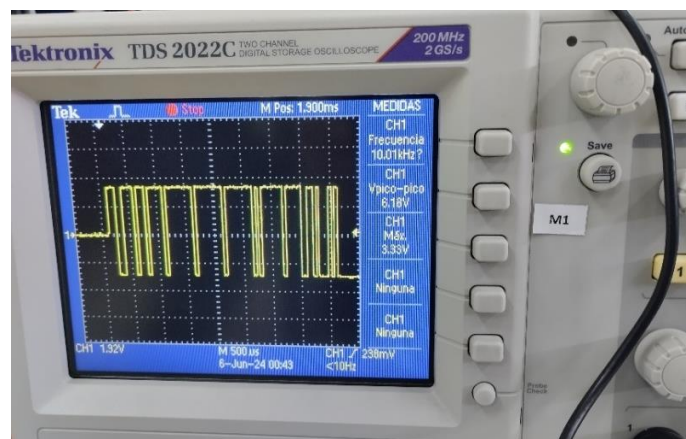


Figura 3.2. Visualización de un mensaje Modbus

Antes de probar el prototipo en una red con el software de emulación del maestro, se utilizaron PLCs como dispositivos esclavos para experimentar con la respuesta del prototipo en una red con varios esclavos. El maestro envió múltiples mensajes al bus de datos, y las respuestas de los diferentes dispositivos se pueden ver en la figura 3.3, que ilustra el ciclo de solicitud/respuesta en una red Modbus RTU.

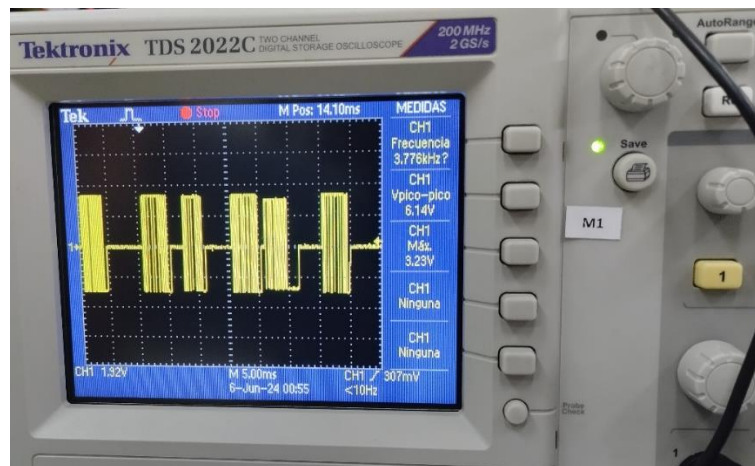


Figura 3.3. Visualización de varios mensajes Modbus en el bus de datos

3.1.3. Prueba de comunicación con un ordenador

Para esta prueba se utilizó un software que emula el comportamiento de un maestro Modbus, el software utilizado fue Modbus Poll, el cual se detalla a continuación.

Modbus Poll

Esta aplicación permite que el ordenador simule un dispositivo maestro de una red Modbus haciendo encuestas o polling a varios dispositivos esclavos, incluye varias funcionalidades propias de las funciones del protocolo como la escritura y lectura de uno o varios registros. La característica más útil para este proyecto es el monitoreo del bus de comunicación en tiempo real, dicho monitoreo muestra la trama de transmisión y la de recepción, con lo cual se puede verificar si los datos que se reciben están acorde a lo que se envió, otorgando así una gran facilidad en el entendimiento de la estructura de la trama para el protocolo Modbus RTU, no obstante, este software no se limita al protocolo antes mencionado, sino que además integra opciones para el protocolo Modbus TCP/IP.

Para el uso de este software es necesario un convertor RS-485 a usb como el que se muestra en la figura 3.4, este convertor fue proporcionado por el laboratorio en el cual se estuvo trabajando para realizar esta prueba.



Figura 3.4. Conversor RS-485-USB

El manual de uso de este software se presenta en la sección de anexos, como anexo 4, en el cual se detallan los pasos para establecer la conexión con cualquier dispositivo que maneje el protocolo Modbus RTU, y tenga como salida las líneas diferenciales A y B. Por otro lado, la figura 3.5 muestra la funcionalidad principalmente usada para esta prueba, la cual es el monitoreo del bus de datos, pudiendo evidenciar así la trama que se transmite por parte del maestro (software Modbus Poll) y la que se recibe por parte del esclavo (prototipo desarrollado).

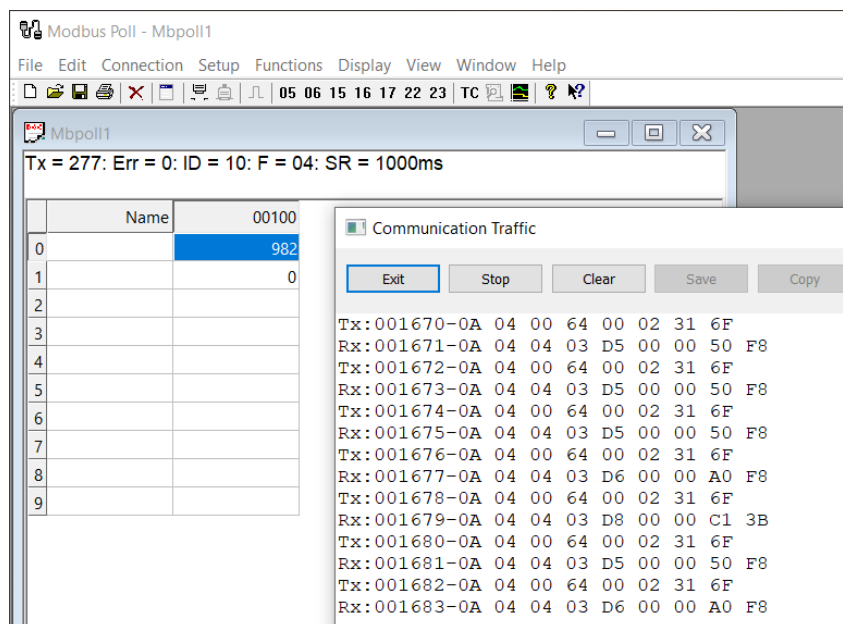


Figura 3.5. Monitoreo de la trama Modbus RTU

Por último, es importante la evidencia conjunta del prototipo con las lecturas con lo cual se presenta la figura 3.6 que evidencia la comunicación entre el ordenador y el prototipo, cabe enfatizar que el prototipo cuenta con indicadores leds, como se detalló en el capítulo dos, de esta forma, el led azul indica la transmisión de datos por parte del prototipo.

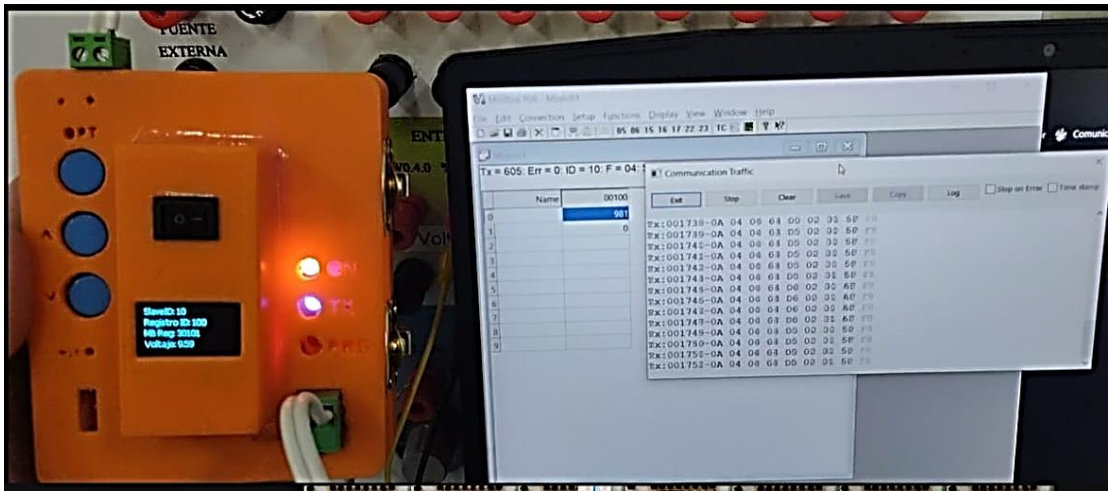


Figura 3.6. Comunicación entre prototipo y ordenador

3.1.4. Prueba de comunicación con PLC uno a uno

Para las pruebas de comunicación con PLC's en general, se utilizaron los controladores del fabricante Allen Bradley de la familia Micrologix 1100, sin embargo, para poder evidenciar la correcta comunicación entre estos dispositivos y el prototipo diseñado, se necesitan conversores avanzados de interfaces, dado que los PLC's destinados para la red Modbus RTU no manejan directamente el interfaz RS-485, de esta manera el laboratorio cuenta con los conversores propios de Allen Bradley modelo 1761-Net-AIC, como los que se muestran en la figura 3.7.

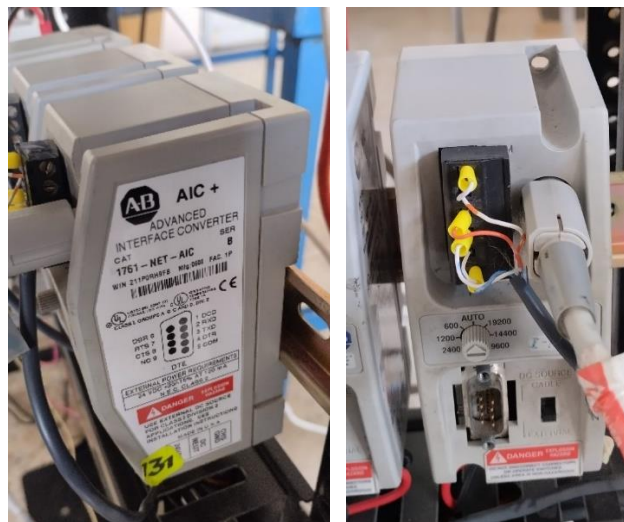


Figura 3.7. Conversor avanzado de interfaces

El conversor mostrado en la figura anterior posee tres puertos, 2 son para el interfaz RS232 mismos que son utilizados por los PLC's y según el cable utilizado pueden conectarse a estos dispositivos con diferentes conectores pudiendo ser estos mini-Din o DB-9.

Por otro lado, el puerto restante es para el interfaz RS-485 tal como se muestra en la figura 3.8 obtenida directamente desde el manual del fabricante.

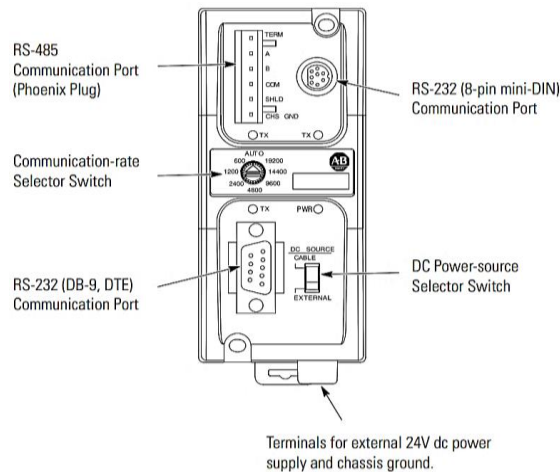


Figura 3.8. Disposición de puertos del convertidor [33]

Ahora bien, lo único que resta es la programación del PLC que figurará como maestro, para ello se utilizó el software RSLogix500 ya que este es el entorno propio de las plataformas de controladores SLC y Micrologix. Una de las principales ventajas de este software es que incluye las funcionalidades requeridas para el manejo de datos mediante el protocolo de interés, es así como la principal configuración a realizar es la de configurar el PLC de modo que su canal de comunicación sea de tipo maestro Modbus, tal como se muestra en la figura 3.9.

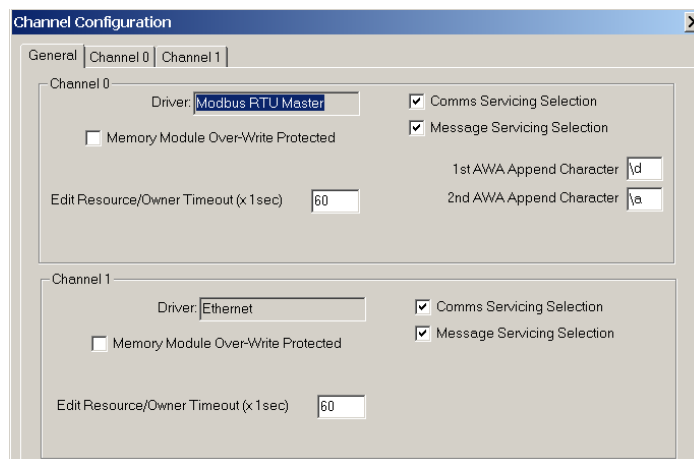


Figura 3.9. Configuración del canal en el PLC como maestro Modbus

Adicionalmente, el software permite configurar otras características asociadas a una red Modbus RTU dependiendo el rol del PLC, de forma particular cuando el PLC cumple la función de maestro el programa permite la configuración de parámetros como la velocidad, la paridad y dependiendo la versión permite la configuración de la estructura serial como la cantidad de bits de datos, la cantidad de bits de inicio y de parada. De forma ilustrativa se presenta la figura 3.10.

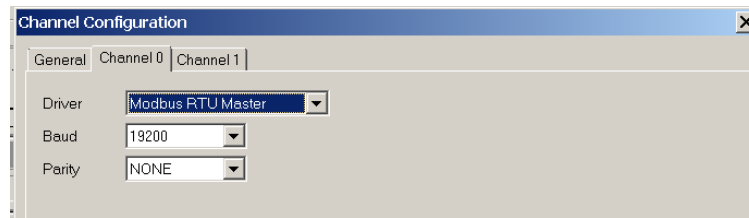


Figura 3.10. Parámetros en la configuración del PLC como maestro Modbus

Por último, se deben añadir los registros de mensaje en el programa y a su vez configurar dichos registros con los parámetros Modbus, es en esta parte donde la comunicación tiene un mayor grado de importancia, de modo que, si estos parámetros se configuran de forma equívoca, la comunicación será fallida, dentro de los parámetros a destacar se tienen los que se listan a continuación y se visualizan en la figura 3.11.

- **Modbus Command:** Función Modbus
- **Data Table Address:** Registro en el PLC que contiene la información a ser enviada o el lugar donde se almacenará la información recibida por el esclavo
- **Size in elementos:** Cantidad de registros a leer/escribir
- **MB Data Address:** Dirección del registro en el esclavo que contiene los datos de interés
- **Slave Node Address:** ID (identificador único) del esclavo al que se realiza la solicitud.
- **MB Address:** Dirección Modbus que puede ser de 5 o 6 dígitos constituida por la función y la dirección del registro en el esclavo que contiene los datos de interés.

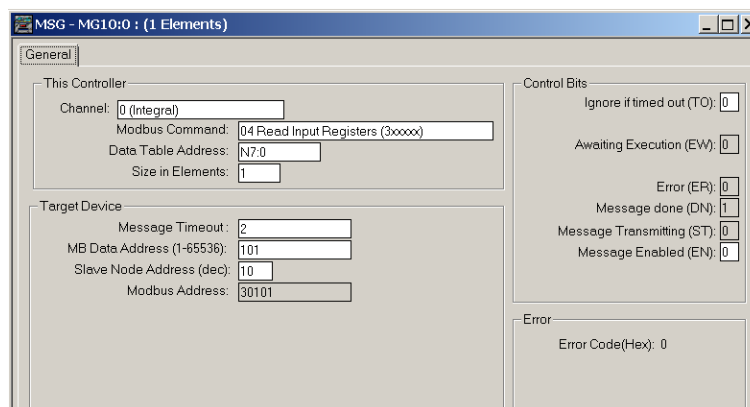


Figura 3.11. Parámetros Modbus en el bloque MSG

Adicionalmente y con fines únicamente ilustrativos, se presenta la figura 3.12 como ejemplo del programa utilizado ya que el reporte completo, obtenido del software RSLogix 500, se presenta en la sección de anexos como anexo 5. Dicha figura consta de dos contactos, uno NO y NC, el bloque de mensaje y un bloque escalador con parámetros para obtener el voltaje a partir del valor digital, producto de la conversión AD.

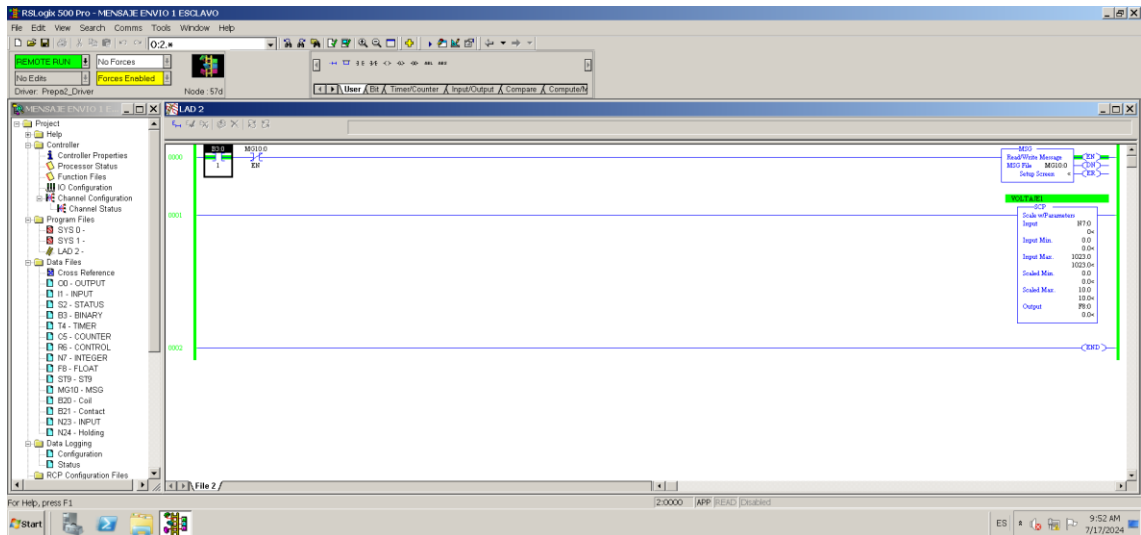


Figura 3.12. Programa para comunicación con el prototipo

Así, la figura 3.13, presenta la evidencia del prototipo funcionando con la red Modbus Serial, por la presencia de los cables no es tan visible dicha relación, sin embargo, para poder ser más riguroso en la constatación de la comunicación, se tienen los leds de los conversores. Para este caso en el que se utiliza un PLC, el prototipo está conectado al primer conversor prendiendo el indicador led de transmisión del puerto RS-485 que se detalló en la figura 3.8, mientras que los otros indicadores están apagados.

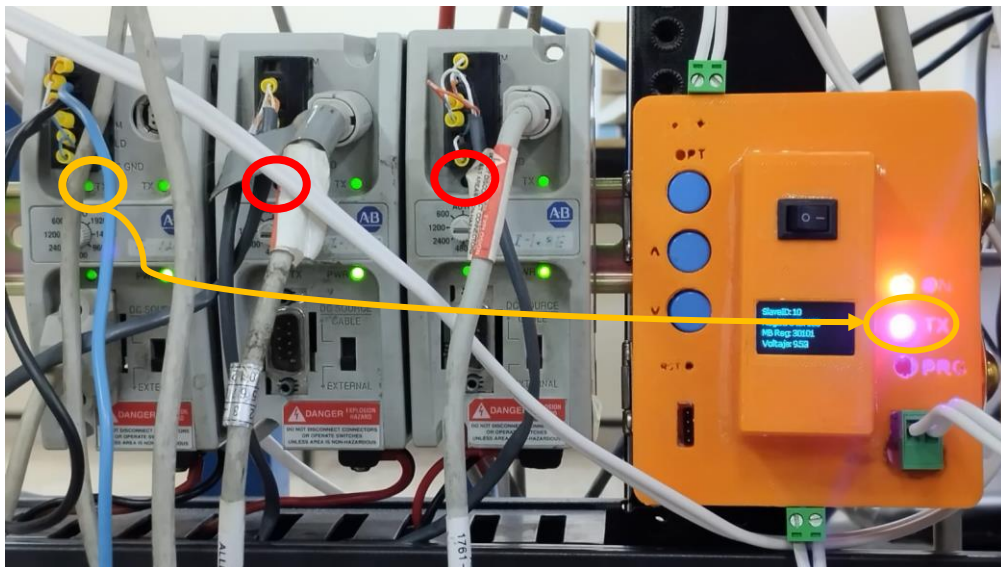


Figura 3.13. Comunicación prototipo y PLC a través del conversor

De igual manera la figura 3.14. respalda la información antes proporcionada con la disposición de indicadores leds proporcionada por el fabricante propiamente.

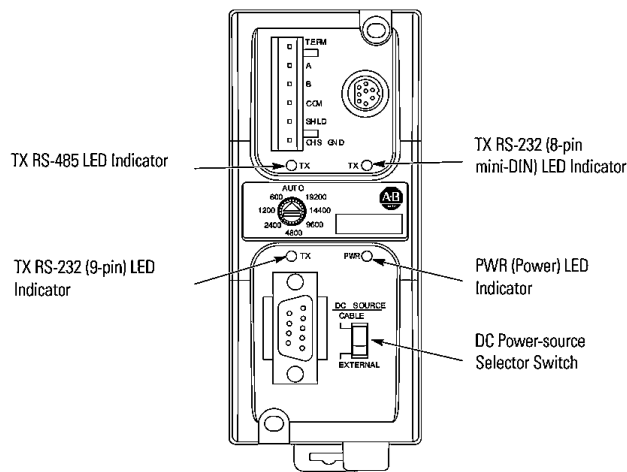


Figura 3.14. Disposición de indicadores de transmisión según el puerto [33]

3.1.5. Prueba de comunicación en una red Modbus

La última prueba realizada con el prototipo es la de su desempeño en una red Modbus con múltiples esclavos, para ello se utilizaron dos PLC's adicionales teniendo así una red de tres PLC's uno como maestro, dos como esclavos y un tercer esclavo que es el prototipo. Por otro lado, para poder realizar las solicitudes a múltiples esclavos se necesita la red de conversores, es por ello que en la figura 3.15 se ilustra las recomendaciones del fabricante para una red tipo Daisy-Chain, en donde se especifican puentes para indicar el inicio y fin del bus de comunicación, así como la forma de conectar los dispositivos derivados, esta forma de conexión es de vital importancia puesto que contiene las especificaciones del interfaz, principalmente la resistencia en las terminaciones de línea para evitar la reflexión o repetición de los datos.

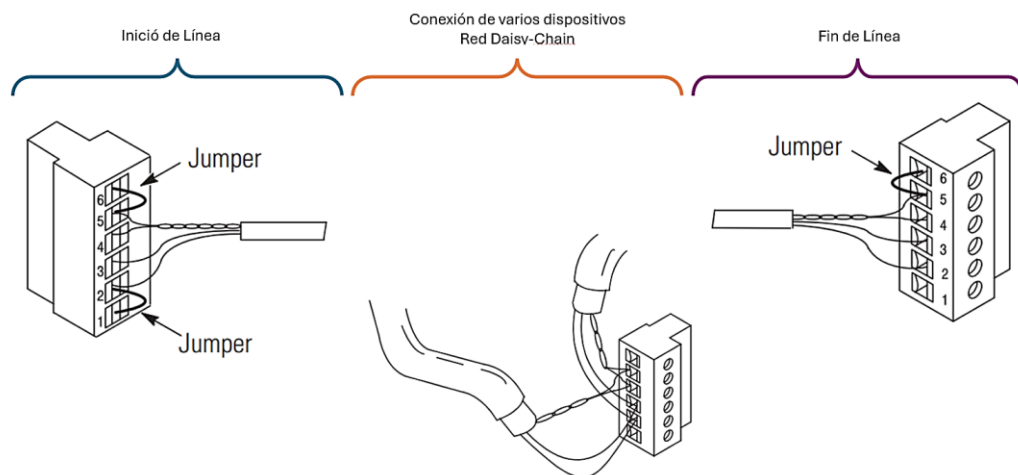


Figura 3.15. Forma de conexión de la red de conversores para Modbus Serial [33]

Cabe mencionar que el cableado según las recomendaciones del fabricante ya se encuentra realizado, puesto que los convertidores son utilizados en algunas prácticas de laboratorio referentes al protocolo Modbus, como evidencia de lo antes mencionado se presenta la figura 3.16 que ilustra la conexión de la red de convertidores, de izquierda a derecha, el primer dispositivo marca el inicio de la línea, el convertidor central es un dispositivo derivado mientras que el tercer convertidor marca el fin de la línea.



Figura 3.16: Red serial de convertidores avanzados de interfaz

Ahora bien, una vez derivando el prototipo en la red serial este tuvo una buena aproximación del mismo comportamiento que el caso donde se comunicaba con un PLC de forma uno a uno, es importante enfatizar que cuando ya se integran más dispositivos a la red, el tiempo de espera o Time Out de los mensajes juega un papel importante, ya que el microcontrolador usa un oscilador externo que se relaciona con la velocidad, mientras que los PLC's basan su arquitectura en arreglos de múltiples microcontroladores o FPGA's por lo que su velocidad no se relaciona con la frecuencia, sino en ciclos de escaneo. No obstante, por las pruebas realizadas el motivo de esta fluctuación no puede ser determinado ya que tiene periodos de variación aleatorios.

La principal causa a la que se le atribuye dicha fluctuación radica en las características propias del microcontrolador utilizado, según la hoja de datos del fabricante el microcontrolador maneja un nivel de errores dependiendo la velocidad serial de transmisión que se utilice de acuerdo con la figura 3.17, ya que, para velocidades menores a 19200 bps, y a tiempos de espera adecuados en el programa del PLC, el programa prototipo no presenta dichas fluctuaciones.

Baud Rate (bps)	$f_{osc} = 16.0000\text{MHz}$			
	U2Xn = 0		U2Xn = 1	
	UBRRn	Error	UBRRn	Error
2400	416	-0.1%	832	0.0%
4800	207	0.2%	416	-0.1%
9600	103	0.2%	207	0.2%
14.4k	68	0.6%	138	-0.1%
19.2k	51	0.2%	103	0.2%
28.8k	34	-0.8%	68	0.6%
38.4k	25	0.2%	51	0.2%
57.6k	16	2.1%	34	-0.8%
76.8k	12	0.2%	25	0.2%
115.2k	8	-3.5%	16	2.1%
230.4k	3	8.5%	8	-3.5%
250k	3	0.0%	7	0.0%
0.5M	1	0.0%	3	0.0%
1M	0	0.0%	1	0.0%
Max. ⁽¹⁾	1Mbps		2Mbps	

Figura 3.17. Ejemplos de ajustes para frecuencias de oscilador de uso común [25]

Donde

f_{osc} = Frecuencia de reloj del oscilador del sistema

$U2Xn = 0 \rightarrow$ Operación normal

$U2Xn = 1 \rightarrow$ Operación normal

$UBRRn$ = El registro de velocidad de transmisión USART

Dicha tabla indica el error que por defecto tendrá el microcontrolador en la transmisión de datos, por ejemplo, para una velocidad de 14400 baudios, se espera que de mil mensajes que se transmitan 6 de estos tengan problemas o que lleguen incompletos, ocasionando fallas o retrasos en la red. Es importante mencionar que dicha información únicamente se presenta como justificación a los percances que se tienen cuando se integran PLC's como esclavos en la red, debido a que en la programación la configuración de estos registros es transparente al programador, debido a que el IDE de Arduino se encarga de realizar las modificaciones a nivel de registros según se desee.

Al igual que en la prueba anterior, se presenta la figura 3.18 como evidencia de la comunicación entre el prototipo y la red Modbus antes descrita, dicho resultado destaca visiblemente, puesto que ahora los indicadores correspondientes a la transmisión en la red RS-485 en los tres convertidores se encuentran iluminados. Es importante aclarar que, debido a la velocidad en la transmisión, las imágenes presentadas son capturas de los videos que se grabaron para poder presentar la evidencia que se menciona, y de igual manera la programación referente a la red completa se detalla en la sección de anexos como anexo 6.



Figura 3.18. Comunicación del prototipo en una red Modbus

Cabe mencionar que dentro del bloque de mensaje MSG existe un apartado que indica los errores por los cuales la comunicación falló, por lo cual se considera oportuno mencionar que la forma para la identificación de errores en la red fue mediante el código de errores propios del software RSLogix500, la representación de estos códigos era en formato hexadecimal, por ejemplo, el error 0x37, uno de los errores más comunes cuando el maestro agota el tiempo de espera en recibir la respuesta del esclavo, de esta forma en la sección de anexos se detallan el código de errores como anexo 7.

Adicionalmente, se debe considerar el archivo de dato de tipo mensaje y su estructura, la figura 3.19 ilustra los campos que contiene dicho elemento, cada uno de los campos que se muestran tiene particularidades que permiten una correcta comunicación, y de acuerdo con la programación, algunos de estos campos sirven para dar paso al siguiente o como un mecanismo alternativo en la detección de errores.

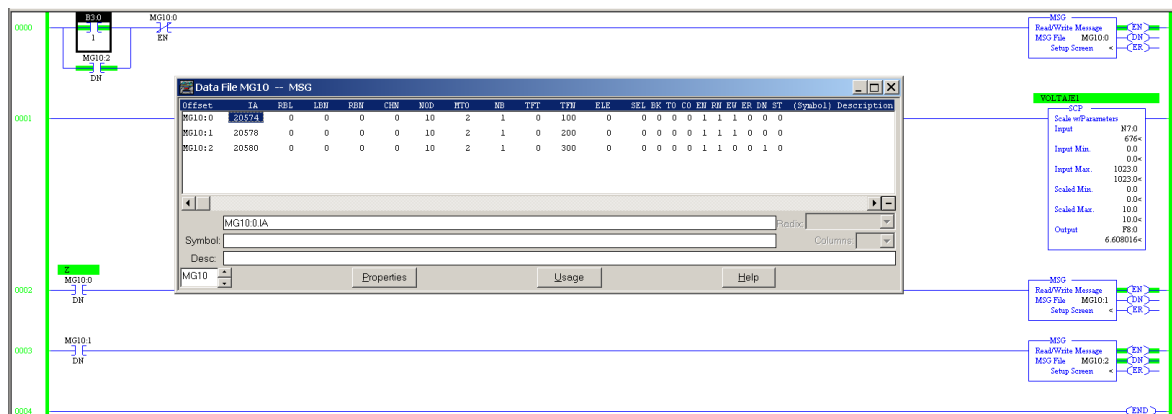


Figura 3.19. Archivo de datos tipo mensaje

El significado de cada campo se lo presenta en la tabla 3.1, la información que se detalla proviene del fabricante propiamente, de esta manera:

Tabla 3.1. Campos en el archivo de datos tipo mensaje

Campo	Nemotecnia	Descripción
IA	Internal Address	Dirección interna del dato en el controlador maestro que se está enviando o recibiendo
RBL	Remote Bit Length	Longitud en bits del dato remoto (del esclavo) que se está leyendo o escribiendo.
LBN	Local Bit Number	Número de bit local en el cual comenzará la transferencia de datos
RBN	Remote Bit Number	Número de bit remoto en el cual comenzará la transferencia de datos.
CHN	Channel	Canal del controlador por el cual se realiza la comunicación
NOD	Slave Node Address	Número del nodo del dispositivo de destino si los dispositivos están en una red Modbus, es decir, la dirección del esclavo
MTO	Message Time Out	Tiempo de espera, en segundos, antes de que el mensaje se considere fallido si no se recibe una respuesta
NB	Number of Bytes	Número de bytes de datos que se están transfiriendo en el mensaje.
TFT	Transmit Format Type	Tipo de formato de transmisión que se está utilizando (por ejemplo, RTU o ASCII)
TFN	Transmit File Number	Número de archivo de transmisión en el controlador maestro donde se almacenan los datos a enviar.
ELE	Element	Elemento específico del archivo de datos que se está utilizando para la transmisión
SEL	Selection	Selección del tipo de operación a realizar (por ejemplo, lectura o escritura)
BK	Block Transfer	Indica si la transferencia de datos se realizará en bloques.
TO	Ignore if timed out	Tiempo de espera específico antes de que la operación de comunicación se considere fallida.
CO	Command Code	Código de comando específico para la operación Modbus (por ejemplo, lectura de registros, escritura de bobinas, etc.)
EN	Message enabled	Habilita o deshabilita la operación del bloque MSG.
RN	Retry Node	Nodo de reintento. Se utiliza para manejar el nodo al cual se intentará reintentar la operación si la comunicación inicial falla.
EW	Execution Wait	Indicador que muestra que el mensaje está esperando ser ejecutado. Significa que el mensaje está en cola y no se ha iniciado aún su procesamiento.
ER	Error	Indicador de error que muestra si ha ocurrido algún problema durante la operación de comunicación.
DN	Message done	Indicador que muestra si la operación de comunicación se ha completado con éxito
ST	Status	Estado actual de la operación de comunicación, que puede incluir información detallada sobre el progreso o problemas encontrados.

Por otro lado, una de las pruebas que se pudo realizar y poder evidenciar en esta sección es la de exactitud en la lectura del voltaje proveniente de un sensor que tenga la salida normalizada de 0 a 10V, obteniendo así los valores que se muestran en la tabla 3.2.

Tabla 3.2. Cálculo de Errores

Tipo de Prueba	Voltaje Leído [V]	Voltaje Teórico [V]	Error [%]
Prototipo – Computadora	5,61	5,47	2,50
	9,94	10	0,60
	1,03	0,98	4,85
Prototipo – PLC (uno a uno)	9,76	9,47	3,02
	6,59	6,80	3,13
	2,43	2,54	4,17
Prototipo – Red Modbus	10,00	9,66	3,57
	4,25	4,52	5,92
	1,18	1,25	5,38

Como se muestra en la tabla, la mayoría de las mediciones tienen un error inferior al 5%, aunque las mediciones medias y bajas de voltaje presentan un error elevado. Esto se debe al tipo de sensor utilizado, ya que se emuló una fuente DC variable comandada por un potenciómetro. La impedancia total del prototipo contribuye a las diferencias entre las lecturas. Otro factor es el diodo Zener de 5.1V asociado a la protección del pin del microcontrolador. Además, la mayor variación de voltaje se atribuye al convertor AD del controlador, cuya resolución de 10 bits puede ser insuficiente para aplicaciones que requieren alta precisión. En el diseño no se consideró el uso de una referencia externa de voltaje, ya que implicaría una entrada adicional en el diseño y requeriría técnicas de reducción de ruido para maximizar la precisión del ADC.

Los resultados cualitativos obtenidos cumplen con los requisitos descritos al inicio del capítulo dos. Como no se dispone de un modelo para comparar la eficiencia del prototipo en función de la cantidad de mensajes transmitidos y los errores, solo se puede verificar su funcionamiento a través de las pruebas realizadas. Estas pruebas se llevaron a cabo junto con la práctica de Profibus en el laboratorio de redes industriales, que utiliza un motor CA. A pesar de las vibraciones y las EMI generadas por el motor, el uso de un par diferencial en el interfaz de comunicación minimiza significativamente estas interferencias, mostrando una afectación mínima por parte del motor.

3.1.6. Análisis económico


Uno de los intereses en el desarrollo de este prototipo fue el de presentar una alternativa escalable y accesible, que, además, incorpora características que el resto de las tarjetas de comunicación, cuyo funcionamiento es parecido al del prototipo desarrollado, no integra. Por esta razón, en la tabla 3.3 se muestra a detalle el listado de componentes utilizadas para poder contrastar dicha información con tarjetas de comunicación comerciales.

Tabla 3.3. Detalle económico del prototipo

COMPONENTE	CANTIDAD	P. UNITARIO	P. TOTAL
LM7812	1	\$0,60	\$0,60
LM7805	1	\$0,60	\$0,60
Capacitor electrolítico 0,33uF	2	\$0,15	\$0,30
Capacitor electrolítico 0,1uF	2	\$0,15	\$0,30
Diodo 1N4004	5	\$0,10	\$0,50
Cristal Oscilador 16MHz	1	\$0,35	\$0,35
Capacitor Cerámico 22nF	2	\$0,10	\$0,20
Capacitor Cerámico 10nF	1	\$0,10	\$0,10
Conector Phoenix 90°	2	\$0,35	\$0,70
Conector Phoenix recto	1	\$0,35	\$0,35
Leds colores	3	\$0,10	\$0,30
Potenciómetro prec. 100K	1	\$0,35	\$0,35
Diodo Zener 5,1V	1	\$0,25	\$0,25
Espadines macho recto	1	\$1,20	\$1,20
Espadines macho 90°	1	\$1,50	\$1,50
Resistencia 1/4W	4	\$0,07	\$0,28
ATMega 328P	1	\$5,50	\$5,50
Módulo MAX485	1	\$2,20	\$2,20
Botón 4P 5mm	1	\$0,15	\$0,15
Botón 4P 12mm	3	\$0,25	\$0,75
Socket	1	\$0,85	\$0,85
Switch 2 posiciones	1	\$0,75	\$0,75
Oled 0,96" I2C	1	\$6,40	\$6,40
Placa PCB	1	\$15,00	\$15,00
Impresión del Case	1	\$30,00	\$30,00
Cable gemelo AWG 18	3	\$0.50	\$1.50
TOTAL			\$70,98

De esta forma, sin considerar el tiempo de diseño de la PCB, diseño de la carcasa, y el tiempo de programación, el costo de esta tarjeta bordea los \$70 aproximadamente, incluyendo elementos de visualización y manipulación como los que se presentan en la tabla 3.4.

Tabla 3.4. Comparación económica entre prototipo y tarjetas comerciales

Descripción	Características	Imagen	Precio	
			Precio Neto	Precio envío
Convertidor de señal analógica a digital modbus Rs485, 4-20mA, adquisición de datos [34]	Módulo de adquisición de datos convertidor de señal 8 canales señal analógica 0-20mA/ 4-20mA a protocolo RS-485 Modbus RTU.		Precio Neto	\$52.95
			Precio envío	\$82.95

Convertidor de señales analógicas a RS-485 [35]	El módulo SM2 de 4 canales de entradas analógicas de Lumel está destinado a convertir señales estándar, de resistencia o de temperatura en datos numéricos, accesibles a través del puerto RS-485 o RS-232 mediante el protocolo MODBUS.		Precio Neto	\$227,99
			Precio envío	\$255,92

Como se muestra en la tabla anterior, los dispositivos comerciales similares al prototipo desarrollado son más costosos y requieren un valor adicional por envío, ya que no se venden en Ecuador. Además, tienen menos elementos de visualización e interacción, dificultando la detección de errores en una red Modbus, al no mostrar valores como voltaje, velocidad y registros.

De esta forma, el prototipo presentado supone una gran herramienta, compartiendo funciones con tarjetas de comunicación comerciales, e incluyendo elementos relativamente flexibles que permiten una mejor interacción del usuario con el equipo, además dichos componentes se los puede encontrar a nivel nacional lo que supone un ahorro significativo en la producción de este prototipo.

3.2 Conclusiones

El presente trabajo de integración curricular cumple los objetivos propuestos, así como el alcance propuesto, dicho cumplimiento de las actividades fue posible gracias a la fase teórica, misma que permitió el análisis y comprensión de la estructura de la trama y la forma en que esta se relaciona con el envío de datos, y posterior a ello permitió la selección de componentes a nivel de hardware y software para la implementación del prototipo de tarjeta de comunicación para la adquisición de una señal analógica de voltaje y su decodificación digital y posterior envío en una trama Modbus RTU.

Se analizó la estructura de la trama Modbus RTU y la diferencia entre los campos PDU y ADU, así como las diferentes funciones que presenta dicho protocolo, para su aplicación en la escritura y lectura de valores de retención o valores de entrada respectivamente, pudiendo ser estos a nivel de bit o a nivel de registros, y su correlación con el modo de transmisión half dúplex.

A través de la implementación del prototipo, se investigó sobre la importancia del cálculo referente a la disipación de calor de elementos reductores de voltaje, de forma particular, en los reguladores lineales y la ventaja que conlleva la división de la potencia total disipada, esto con la finalidad de evitar mecanismos de ventilación forzada o enfriamiento que supondrían un mayor costo en su implementación.

Se verificó la importancia de archivos Gerber en el diseño de placas de circuito impreso PCB, de esta forma se adquirieron habilidades y conocimientos en el manejo del software Eagle de diseño de PCB's, y la forma de estandarizar y comprobar reglas de diseño (DRC) en la PCB, y de errores (ERC) en el esquema electrónico.

El uso de cable UTP rígido de categoría no industrial afecta considerablemente la comunicación, si bien el par trenzado tiene una resistencia adicional debido a la cantidad de vueltas en su trenzado propiamente, la característica de su rigidez genera que las terminales que conectan los dispositivos en ambos lados de la línea puedan fallar, debido a que son susceptibles a fracturarse por lo que se pierde la continuidad. De esta forma no se recomienda su uso para comunicaciones y en su lugar se sugiere el uso de cable flexible.

Es importante conocer las limitaciones que poseen los microcontroladores según su datasheet, y comprender la tasa de error a diferentes velocidades, de forma particular para el proyecto a una velocidad común de 19200 baudios y considerando un total de 1000 mensajes, según la guía del microcontrolador Atmega328P, fallarán 2 mensajes por lo que el prototipo tendrá menos resistencia al ruido cuando los valores de error sean altos en función de una mayor velocidad.

Debido a las características del microcontrolador y la programación estructurada que este admite, el uso de dos buses de comunicación afecta significativamente a la comunicación del protocolo cuando esta no es uno a uno, ya que atiende la solicitud de comunicación y posterior a ello debe atender la solicitud de actualización en la pantalla.

Dentro del diseño de componentes electrónicas, el diseño modular representa una ventaja económica en la sustitución de elementos que por situaciones como vida útil o fallas en su operación dejen de funcionar, descartando así la obsolescencia total del prototipo y el desperfecto se solucione cambiando el componente.

3.3 Recomendaciones

Debido a la limitación del microcontrolador en el manejo de dos buses de datos, se recomienda el uso de otro microcontrolador como la ESP32 ya que esta placa de desarrollo posee un doble núcleo, es decir, dos procesadores de forma que un procesador puede destinarse únicamente al manejo del bus I2C, mientras que el segundo procesador puede dedicarse al intercambio de información y armado de la trama según el protocolo.

Para la codificación del microcontrolador se propone el uso de la metodología de programación secuencial mediante el cambio de estados para asegurar el cumplimiento de los diferentes eventos asociados a dichos estados, ya que dicha metodología fue probada y tuvo excelentes resultados en la comunicación con otros dispositivos esclavos, sin embargo, al incluir la pantalla oled se generaban retardos que afectaban el cambio de estado de los eventos que retornaban a la comunicación de datos, con lo cual se producía la caída de la red serial.

Como alternativa y mejora en el diseño del prototipo, se recomienda el cambio de pantalla oled por el LCD 16*2 ya que al manejarse a nivel de bits representa una mejora significativa evitando problemas de colapso en el microcontrolador por el uso de dos buses de comunicación, y en consecuencia, evitando el colapso de la red serial, sin embargo, la ventaja que supone a nivel de comunicación se contrasta con la programación de los campos de visualización para el usuario y los parámetros del menú configurable, añadiendo un número considerable de líneas de código lo cual repercute directamente en el uso de la memoria del microcontrolador.

En lo referente al uso de módulos ya fabricados, se recomienda el análisis y entendimiento de su funcionamiento mediante su esquema electrónico, esta práctica no solo permite la identificación y corrección de fallas, sino que además permite un mejor manejo de errores, de forma particular, las líneas del módulo Max485 en algunos casos están invertidas en función de la serigrafía de la placa, se hicieron pruebas con tres módulos y dos de estos tenían la señal negativa en A y la señal positiva en B según su serigrafía.

Una mejora significativa evolucione al prototipo desarrollad, sería su integración con otras versiones del protocolo, el prototipo en su salida entrega una trama de bits, mismos que conforman el PDU y el ADU para esta versión en particular del protocolo. Si se discrimina el PDU, y se añaden los campos que requieran otras versiones como TCP/IP al ADU, se tendría una tarjeta más completa, integrable a cualquier red de dispositivo y que fácilmente cambie entre las versiones del protocolo,

4 REFERENCIAS BIBLIOGRÁFICAS

- [1] V. G. Jimenez, R. Y. Yuste, y L. Martínez, *Comunicaciones Industriales Siemens*. Marcombo, 2012.
- [2] «Transmisión de datos», *Wikipedia, la enciclopedia libre*. 20 de mayo de 2024. Accedido: 16 de junio de 2024. [En línea]. Disponible en: https://es.wikipedia.org/w/index.php?title=Transmisi%C3%B3n_de_datos&oldid=160233963
- [3] V. M. CARLOS, *Comunicaciones industriales*. Ediciones Paraninfo, S.A., 2019.
- [4] AULA21, «Qué son las redes de comunicación industrial y cómo funcionan». Accedido: 16 de junio de 2024. [En línea]. Disponible en: <https://www.cursosaula21.com/que-son-las-redes-de-comunicacion-industrial/>
- [5] W. Tomasi, *Sistemas de comunicaciones electrónicas*. Pearson Educación, 2003.
- [6] N. O. ALONSO y Vvaa, *Redes de comunicaciones industriales*. Editorial UNED, 2013.
- [7] R. N. Lacoba, *La guía exprés de redes*. Rocío Navarro Lacoba, 2014.
- [8] «Estándares y Tecnologías - Estándares y Tecnologías», MINTIC Colombia. Accedido: 16 de junio de 2024. [En línea]. Disponible en: <https://www.mintic.gov.co/portal/715/w3-article-5236.html>
- [9] «Estandares comunicacion», SlideShare. Accedido: 16 de junio de 2024. [En línea]. Disponible en: <https://es.slideshare.net/slideshow/estandares-comunicacion/208788458>
- [10] Modbus Organization, «Modbus_Application_Protocol_V1_1b3.pdf». 24 de octubre de 2006. Accedido: 16 de junio de 2024. [En línea]. Disponible en: https://modbus.org/docs/Modbus_Application_Protocol_V1_1b3.pdf
- [11] «Modbus: Qué es y cómo funciona | Comunicaciones Industriales». Accedido: 16 de junio de 2024. [En línea]. Disponible en: <https://www.cursosaula21.com/modbus-que-es-y-como-funciona/>
- [12] «Determinism in industrial ethernet:A technology overview — Part 2Determinism in industrial ethernet:
A technology overview — Part 2 :: Process Online». Accedido: 16 de junio de 2024. [En línea]. Disponible en: <https://www.processonline.com.au/content/industrial-networks-buses/article/determinism-in-industrial-ethernet-br-a-technology-overview-part-2-966929628>
- [13] B. S. Manuel, *Ingeniería de instrumentación de plantas de proceso*. Ediciones Díaz de Santos, 2019.
- [14] D. D. Chowdhury, *NextGen Network Synchronization*. Springer Nature, 2021.
- [15] NOVUS Automation, «Conceptos Fundamentales de RS485 y RS422». 17 de diciembre de 2019. [En línea]. Disponible en: <https://www.novusautomation.com/es/download/conceptos-fundamentales-de-rs485-y-rs422>
- [16] plc247.com, «What is RS-485 Communication?», plc247.com. Accedido: 31 de julio de 2024. [En línea]. Disponible en: <https://plc247.com/what-is-rs-485-communication/>
- [17] Modbus.org, «Modbus_over_serial_line_V1_02.pdf». 20 de diciembre de 2006. Accedido: 9 de julio de 2024. [En línea]. Disponible en: https://www.modbus.org/docs/Modbus_over_serial_line_V1_02.pdf
- [18] A. Castro y R. Fusario, *Comunicaciones y Redes: Para profesionales en sistemas de información*. Alpha Editorial, 2015.
- [19] andhurta, «▷ Protocolo Modbus | Para Principiantes», EYEMUC. Accedido: 10 de julio de 2024. [En línea]. Disponible en: <https://eeymuc.co/31-protocolo-modbus/>
- [20] S. Gamboa, «PROTOCOLO DE COMUNICACIÓN INDUSTRIAL: MODBUS».
- [21] «Simplex vs Duplex Fiber: What are the Differences?», CBT Nuggets. Accedido: 9 de julio de 2024. [En línea]. Disponible en: <https://www.cbtnuggets.com/blog/technology/networking/simplex-vs-duplex-fiber>
- [22] N. C. Braga, *Electrónica Analógica*. Editora Newton C. Braga, 2019.

- [23] V. M. M. Sáiz y A. P. López, *Sistemas electrónicos de potencia en el buque*. Ed. Universidad de Cantabria, 2006.
- [24] J. V. del Real, *Circuitos lógicos digitales 4ed*. Marcombo, 2024.
- [25] Microchip Technology Inc, «ATMEGA328P». Accedido: 24 de junio de 2024. [En línea]. Disponible en: <https://www.microchip.com/en-us/product/atmega328p>
- [26] Naylamp Mechatronics, «NodeMCU-32 30-pin ESP32 WiFi», Naylamp Mechatronics - Perú. Accedido: 24 de junio de 2024. [En línea]. Disponible en: <https://naylampmechatronics.com/espressif-esp/384-nodemcu-32-30-pin-esp32-wifi.html>
- [27] Naylamp Mechatronics, «Módulo MAX485 Conversor RS485 a Serial TTL», Naylamp Mechatronics - Perú. Accedido: 24 de junio de 2024. [En línea]. Disponible en: <https://naylampmechatronics.com/conversores-ttl/62-conversor-rs485-a-serial-ttl.html>
- [28] «sn75176a.pdf». Accedido: 24 de junio de 2024. [En línea]. Disponible en: https://www.ti.com/lit/ds/symlink/sn75176a.pdf?ts=1719283296555&ref_url=https%253A%252F%252Fwww.google.com%252F
- [29] Naylamp Mechatronics, «Display Oled 0.96" I2C 128*64 SSD1306», Naylamp Mechatronics - Perú. Accedido: 24 de junio de 2024. [En línea]. Disponible en: <https://naylampmechatronics.com/oled/850-display-oled-096-i2c-12864-ssd1306.html>
- [30] «Pantalla LCD 16x2 es una pantalla alfanumérica de 16 columnas y 2 líneas», Electromanía Perú. Accedido: 24 de junio de 2024. [En línea]. Disponible en: <https://www.electromania.pe/producto/pantalla-lcd-1602/>
- [31] alldatasheet.es, «LM7805 Datasheet(PDF)». Accedido: 30 de junio de 2024. [En línea]. Disponible en: <http://www.alldatasheet.es/datasheet-pdf/pdf/82833/FAIRCHILD/LM7805.html>
- [32] W. O. Hachen, «Implementación de filtros pasa bajos», Medium. Accedido: 1 de julio de 2024. [En línea]. Disponible en: <https://willyexe97.medium.com/implementaci%C3%B3n-de-filtros-pasa-bajos-d50626c972be>
- [33] Bradley Allen, «Convertidor de interface avanzado (AIC+)», julio de 2003.
- [34] «Convertidor de señal analógica a digital modbus rs485, 4 20mA, adquisición de datos - AliExpress 1420», aliexpress. Accedido: 21 de julio de 2024. [En línea]. Disponible en: https://es.aliexpress.com/item/4000381538848.html?src=ibdm_d03p0558e02r02&sk=&aff_platform=&aff_trace_key=&af=&cv=&cn=&dp=
- [35] «SM2, RS-485 MODBUS Module with 4 Analogue Inputs - PSE - Priggen Spec, 209,30 €», SM2, RS-485 MODBUS Module with 4 Analogue Inputs - PSE - Priggen Spec, 209,30 €. Accedido: 21 de julio de 2024. [En línea]. Disponible en: <https://www.priggen.com/SM2-RS-485-MODBUS-Module-with-4-Analogue-Inputs>
- [36] «Modbus Poll user manual». Accedido: 22 de julio de 2024. [En línea]. Disponible en: <https://www.modbustools.com/mbpoll-user-manual.html>

5 ANEXOS

ANEXO I. Esquema electrónico del prototipo

ANEXO II. Identificación de las etapas en la placa

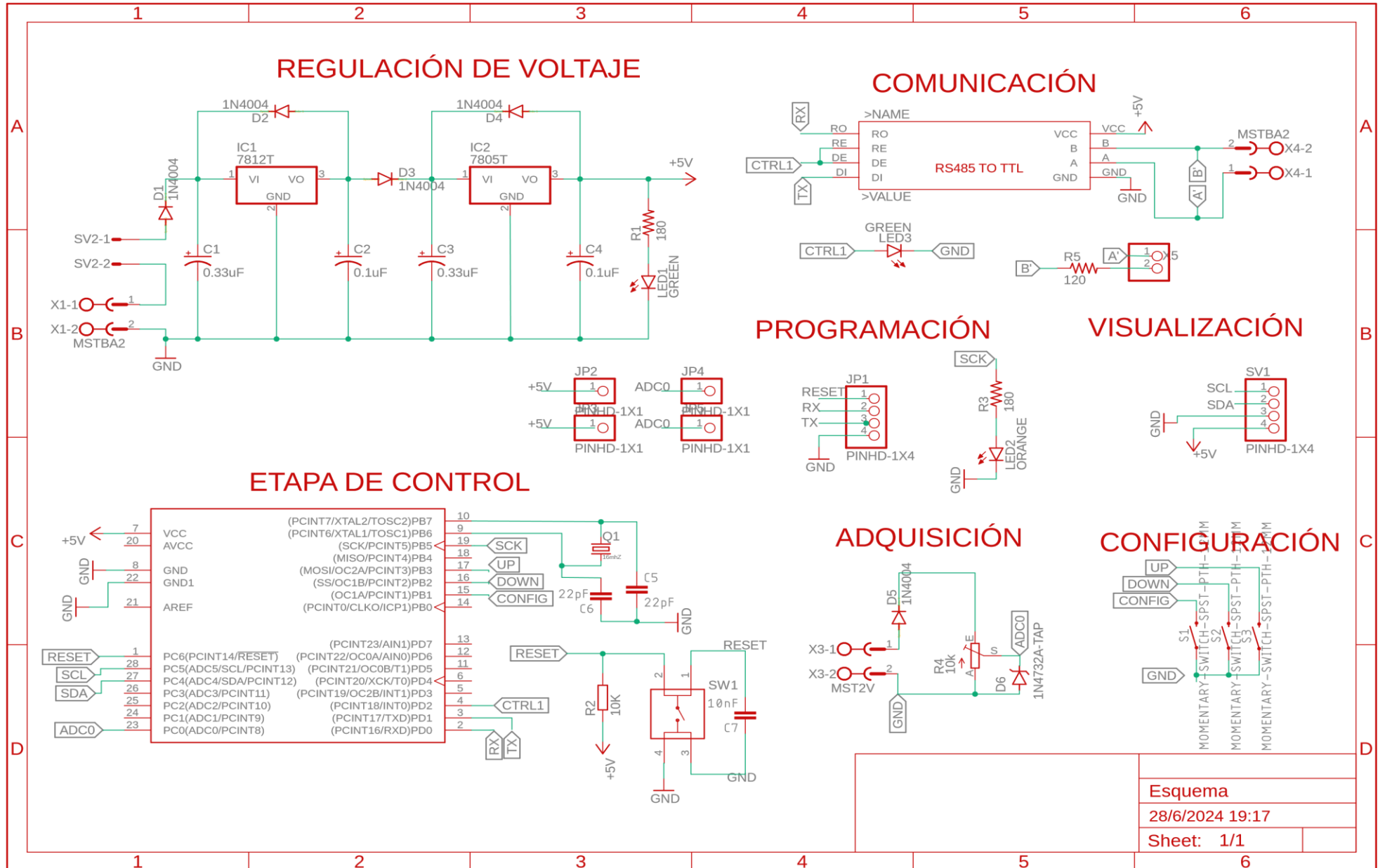
ANEXO III. Vistas del prototipo

ANEXO IV: Manual de configuración software Modus Poll

ANEXO V: Códigos de errores en el bloque MSG

ANEXO VI: Manual de usuario

5.1 ANEXO I: Esquema del Prototipo



5.2 ANEXO II: Identificación de las etapas en la placa

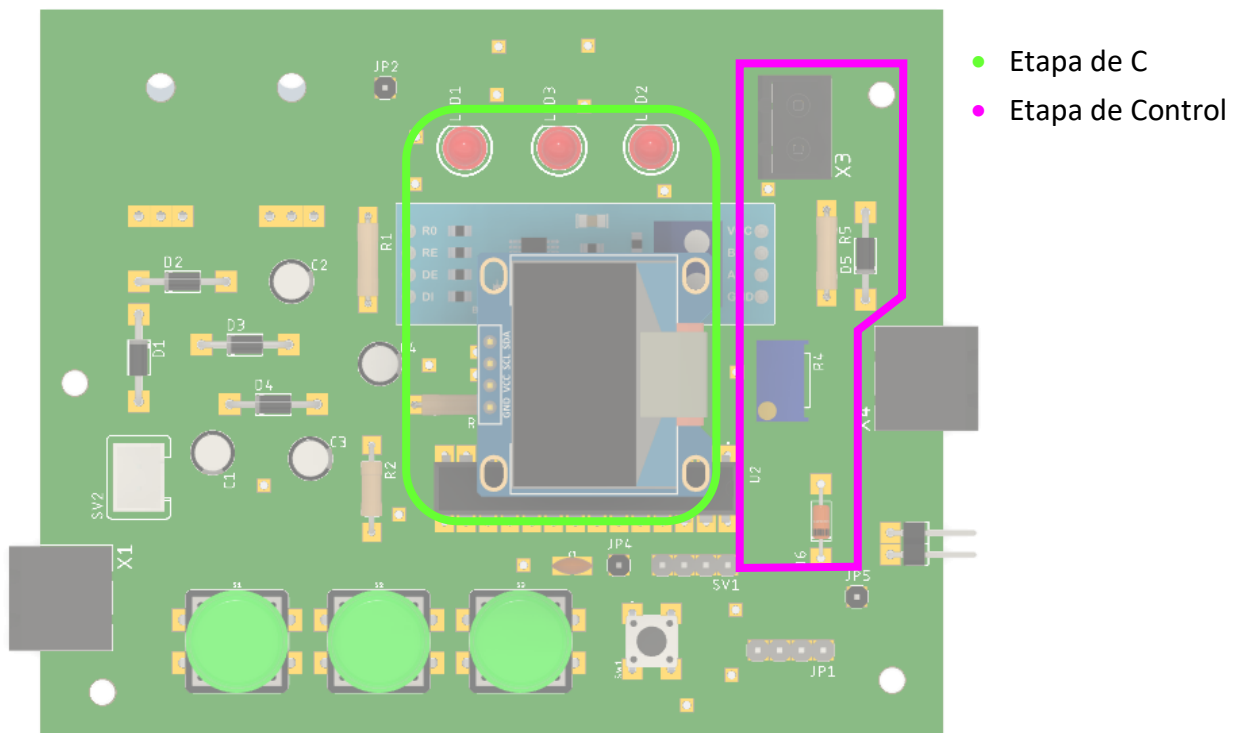
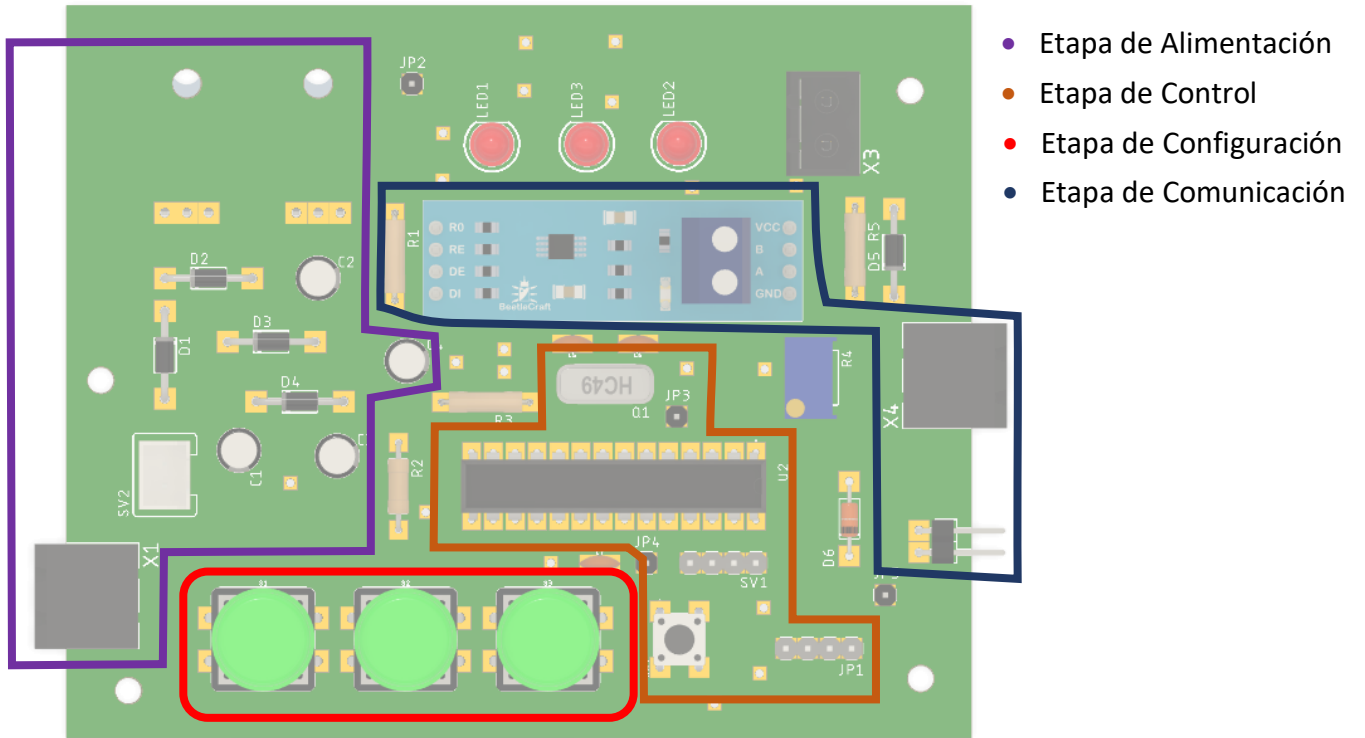


Figura AII.1. Etapas diseñadas en PCB

5.3 ANEXO III: Vistas del prototipo

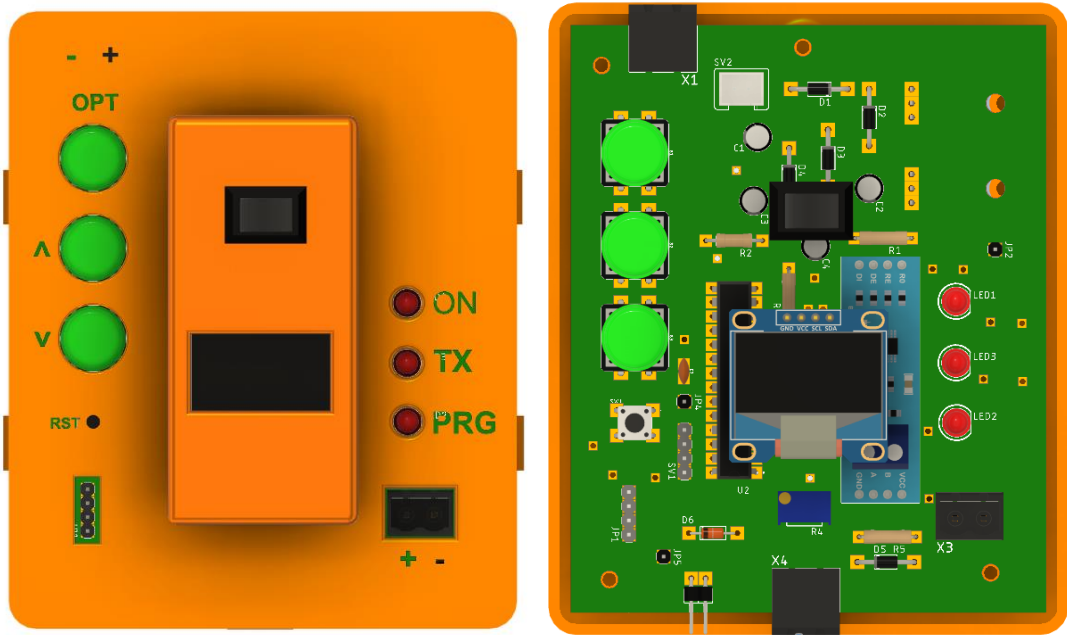


Figura AIII.1. Vista superior con y sin cubierta

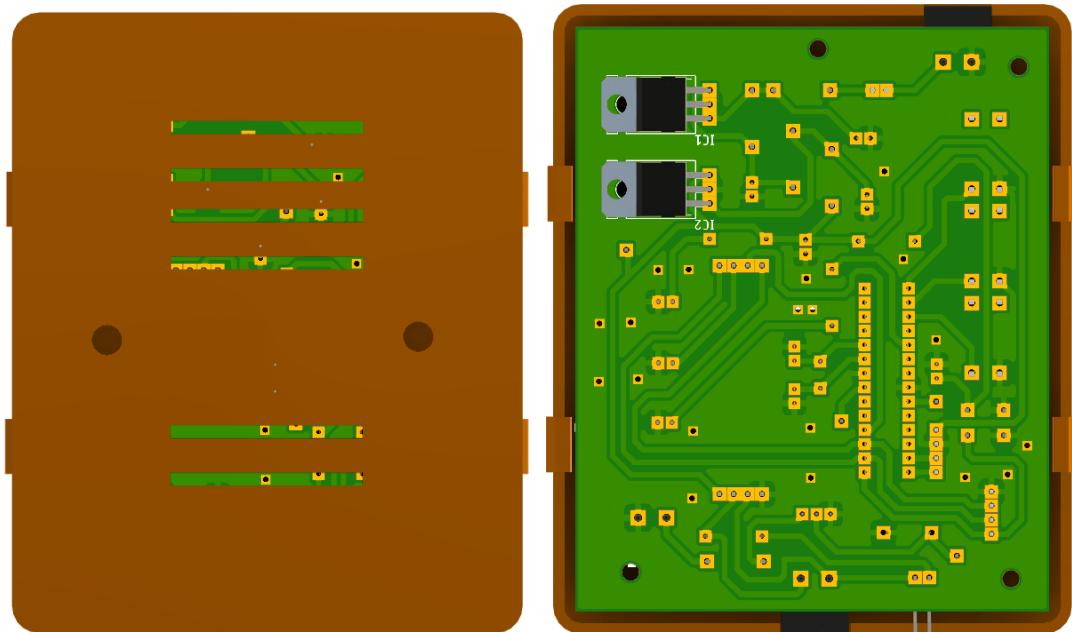


Figura AIII.2. Vista inferior con y sin cubierta



Figura AIII.3. Vista frontal y posterior del prototipo

5.4 ANEXO IV: Manual de configuración software Modbus Poll

Modbus Poll es un software de simulación de maestro Modbus diseñado para probar y depurar dispositivos Modbus esclavos, esta guía tiene como objetivo servir de ayuda para configurar y utilizar Modbus Poll para las diferentes necesidades de comunicación Modbus.

Modbus Poll es una herramienta versátil para la lectura y escritura de datos en dispositivos Modbus que utilizan Modbus RTU o ASCII en redes RS232/RS-485, Modbus TCP/IP y Modbus UDP/IP, incluyendo la encapsulación de Modbus en paquetes TCP/UDP. Soporta diversas funciones Modbus como leer bobinas, registros y entradas, escribir bobinas y registros, y funciones de diagnóstico. Permite registrar datos en archivos de texto o Excel y ofrece múltiples formatos de visualización para celdas, incluyendo enteros y flotantes en diferentes ordenes de registros/bytes. Adicionalmente, cuenta con características avanzadas como automatización OLE, monitoreo del tráfico de datos, impresión, selección de fuentes, colores condicionales, gráficos en tiempo real y la capacidad de guardar y abrir espacios de trabajo [36].

Descarga e Instalación

Como paso previo a los pasos para la configuración del software es importante descargar la versión más reciente del programa en el sitio web oficial de Modbus Poll y ejecutarlo con la ayuda del asistente de instalación, una vez instalado se debe añadir un ícono de acceso rápido en la pantalla de inicio como el que se muestra en la figura AIV.1, dicho ícono corresponde a la versión 10.5.3 del software.

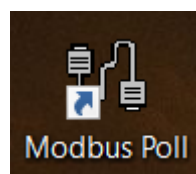


Figura AIV.1. Acceso rápido Modbus Poll

Ventana Principal

Una vez instalado el software se procede a abrirlo, teniendo como inicio la ventana que se muestra en la figura AIV.2, Modbus Poll utiliza una interfaz de documentos múltiples lo que significa que se pueden abrir varias ventanas. Cada uno con diferentes contenidos de datos de diferentes dispositivos esclavos al mismo tiempo.

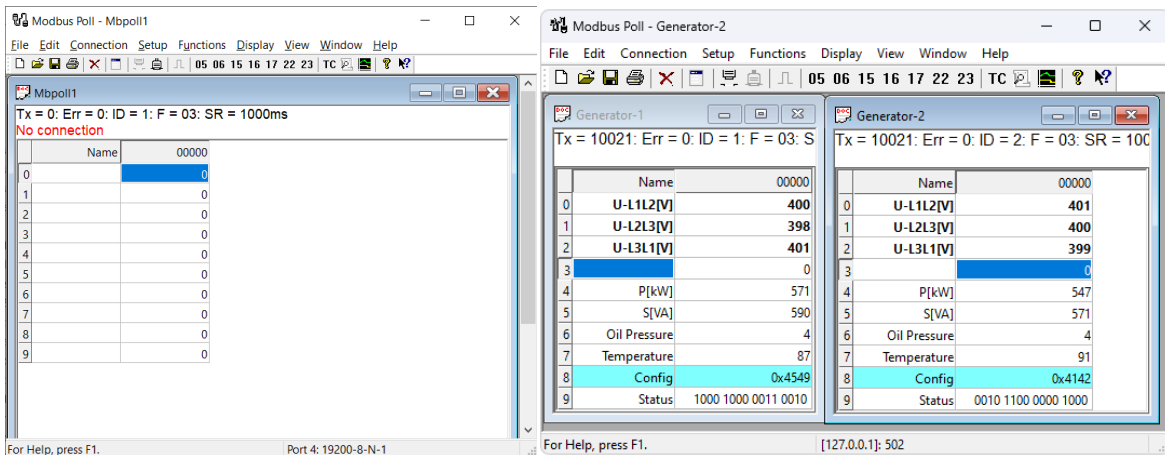


Figura AIV.2. Ventana de inicio en Modbus Poll (derecha), uso de múltiples documentos (izquierda)

Dentro de la ventana principal se tienen bandas y barras de herramientas muy intuitivas para el usuario como las que se detallan en la figura AIV.3, cada una de estas herramientas varía de acuerdo con las necesidades que deba satisfacer el maestro.

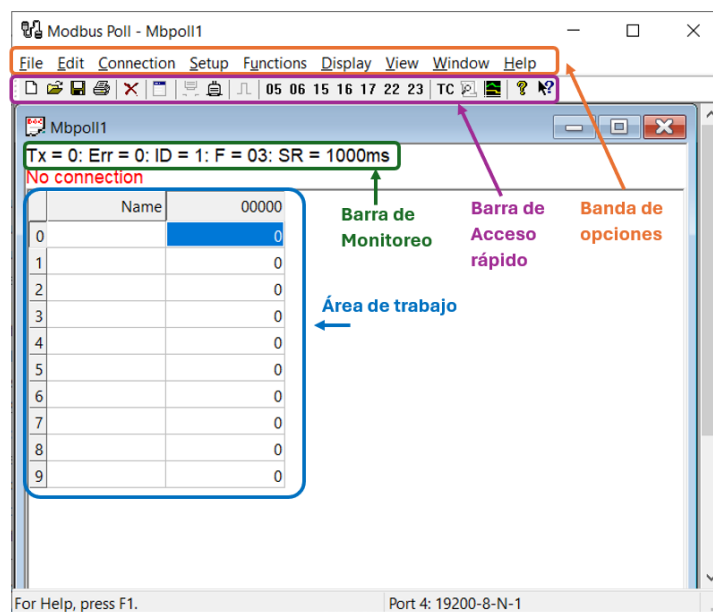


Figura AIV.3. Elementos en la ventana de Modbus Poll

Conexión

Para la conexión con el dispositivo se tienen dos alternativas, se puede acceder con el acceso rápido del teclado, mediante la tecla F3, o en su defecto, en la banda de opciones seleccionar la alternativa "Connection", y la opción Connect del submenú desplegable, una vez realizada dicha configuración se muestra la ventana de la figura AIV.4.

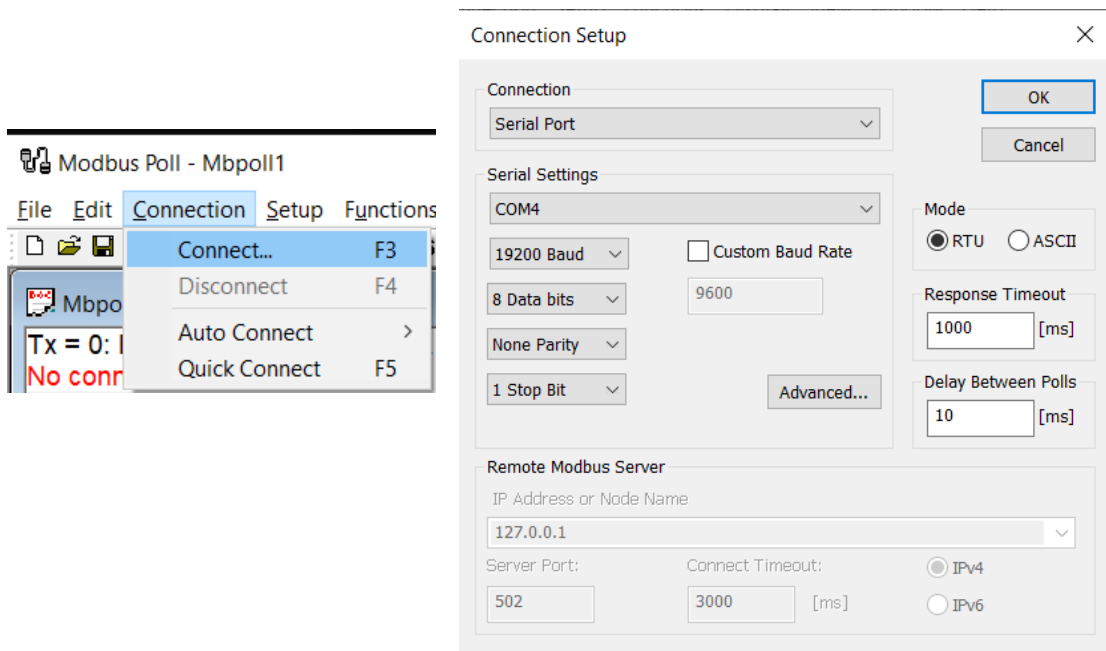


Figura AIV.4. Cuadro de diálogo de conexión


En la sección connection del cuadro de diálogo, se tiene 5 opciones:

- **Serie:** Modbus sobre línea serie. RS232 o RS-485. Se puede utilizar un convertidor serie USB.
- **Modbus TCP/IP:** Seleccione TCP/IP si desea comunicarse con una red MODBUS TCP/IP. En este caso, el ID de esclavo es el igual que el ID de unidad utilizado en MODBUS TCP/IP. El número de puerto es el 502 predeterminado. Si la conexión falla, intente hacer ping a su dispositivo en el símbolo del sistema. Si el comando ping falla, el Modbus Poll también falla.
- **Modbus UDP/IP:** Seleccione UDP/IP si desea comunicarse con una red MODBUS UDP/IP. Esto es lo mismo que Modbus TCP/IP, pero en su lugar se utiliza el protocolo UDP sin conexión.
- **Modbus RTU/ASCII sobre TCP/IP:** Se trata de un mensaje RTU o ASCII enviado a través de una red TCP/IP en lugar de líneas seriales.
- **Modbus RTU/ASCII a través de UDP/IP:** Se trata de un mensaje RTU o ASCII enviado a través de una red UDP/IP en lugar de líneas seriales.

Para interés de este manual únicamente se presentará la configuración de los parámetros referentes a la comunicación serial, de esta forma.

- **Puerto:** Indica el puerto USB donde se conecta el conversor RS-485 a USB

- **Modo:** Utilice esta opción para seleccionar el modo RTU o ASCII. RTU predeterminada.
- **Tiempo de espera de respuesta:** Especifica el tiempo que Modbus Poll debe esperar una respuesta de un dispositivo esclavo antes de darse por vencido. El valor predeterminado es 1000 ms.
- **Retraso mínimo entre solicitudes:** Esta configuración garantiza un retraso mínimo hasta que se transmita la siguiente solicitud, independientemente de la velocidad de escaneo. La resolución de esta configuración es de aproximadamente 15 ms. Es posible en algunos ordenadores obtener una mejor resolución, pero no en todos.
- **Velocidad en baudios del usuario:** Especifique una velocidad en baudios personalizada si ninguna de las velocidades en baudios predeterminadas es útil.
- **Formato de la trama Serial:** Para los bits de datos se puede configurar una trama de 7 u 8 bits de datos, para la paridad se puede configurar ninguna, par o impar, mientras que para los bits de parada se pueden configurar entre uno y dos bits de parada.

	<p>Si establece este valor por debajo de 20 ms, no se puede garantizar el intervalo de tiempo de 3,5 caracteres entre la respuesta y una nueva solicitud. Esto se debe a que el programador de Windows cambia de tarea cada 10 a 20 ms. Si sondea varios esclavos en una red serie RS-485, NO debe establecer el valor inferior a 20 ms. Esto es para garantizar el intervalo de tiempo de 3,5 caracteres. En una red TCP/IP, menos de 20 ms está bien. La conexión en serie a un solo dispositivo esclavo de menos de 20 ms está bien.</p>
---	---

Definición de lectura/escritura

Modbus poll permite añadir configuraciones para definir los datos que se van a supervisar en la ventana activa de la pantalla de trabajo, para acceder a dicha pantalla se tiene el acceso rápido por teclado mediante la tecla F8, o mediante el acceso que se describe en la figura AIV.5.

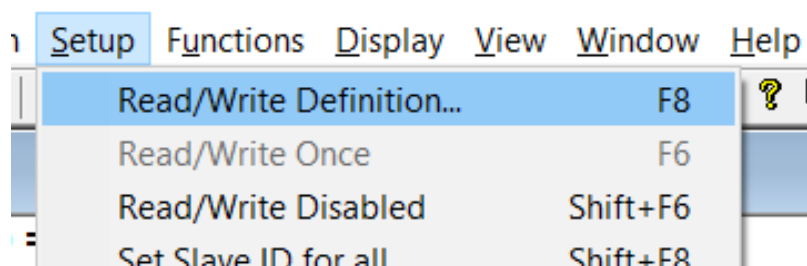


Figura AIV.5. Acceso a la configuración de lectura/escritura

Una vez presionada la tecla o accedido mediante la banda de opciones se despliega el cuadro de configuración como el que se muestra en la figura AIV.6, en dicha ventana se configurarán los parámetros sobre los dispositivos esclavos, así como la función modbus utilizada y el tamaño de datos a leer.

Figura AIV.6. Cuadro de configuración lectura/escritura

En este cuadro se tienen los siguientes parámetros:

- **ID del esclavo:** Rango de 1 a 255. (Las especificaciones del protocolo MODBUS dicen 247). También se acepta el valor 0 para comunicarse directamente con un dispositivo MODBUS/TCP o MODBUS/UDP.
- **Código de función:** Se puede seleccionar 1 de los 8 códigos de función.
 - Funciones de lectura: Los datos devueltos por las funciones de lectura se muestran en la ventana de cuadrícula.
 - 01: Bobinas de lectura (0x)
 - 02: Leer entradas discretas (1x)
 - 03: Lectura de registros de retención (4x)
 - 04: Lectura de registros de entrada (3x)
 - Funciones de escritura: Las funciones de escritura escriben los datos que se muestran en la ventana de cuadrícula.

- 05: Escribir bobina simple (escribe en estado de bobina)
- 06: Escribir registro único (Escribe en registros de retención)
- 15: Escribir varias bobinas (Escribe en bobinas)
- 16: Escribir varios registros (Escribe en registros de retención)

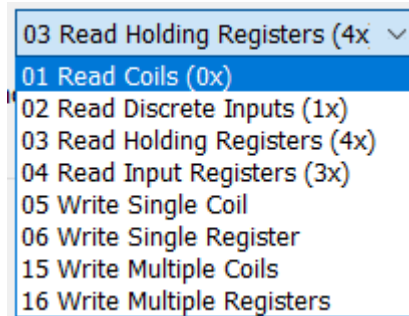


Figura AIV.7. Funciones Modbus en el software Modbus Poll

- **Dirección:** Algunas especificaciones de protocolo utilizan la dirección o los registros del dispositivo. El primer dígito describe la función que se va a utilizar. Eso significa que la dirección del dispositivo 40101 se identifica con la dirección 100. El "4" significa que los registros de tenencia y los registros 4x, al ser de base uno este cuenta los registros desde dicho número.
- **Scanrate:** La velocidad de escaneo se puede establecer de 0 a 3600000 ms, se debe tener en cuenta que establecer una velocidad de escaneo menor a la del tiempo de espera no tiene sentido.

Una de las principales ventajas de este software, es que previo al establecimiento de comunicación, arma la trama tanto RTU como ASCII, para poder hacer un mejor monitoreo, como se detalla en la figura AIV.8, por ejemplo, para hacer la solicitud de lectura al esclavo 10, cuyo valor de lectura se encuentra en el registro 100 y se utiliza la función 4 de lectura de inputs registers, se tiene:

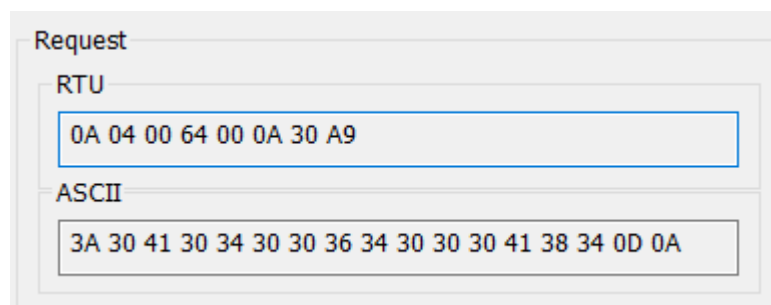


Figura AIV.8. Armado de trama previa comunicación.

Monitoreo del tráfico de datos

Por último en este manual, se presenta la forma de acceder al monitoreo del bus de comunicación, esta herramienta permite observar la trama que se envía y la que se recibe en función de la trama previamente mostrada en la figura AIV.8, para ello, en la barra de acceso rápido se selecciona la opción de lupa, y una vez iniciada la comunicación se muestra el resultado de una conexión exitosa como en la figura AIV.9.

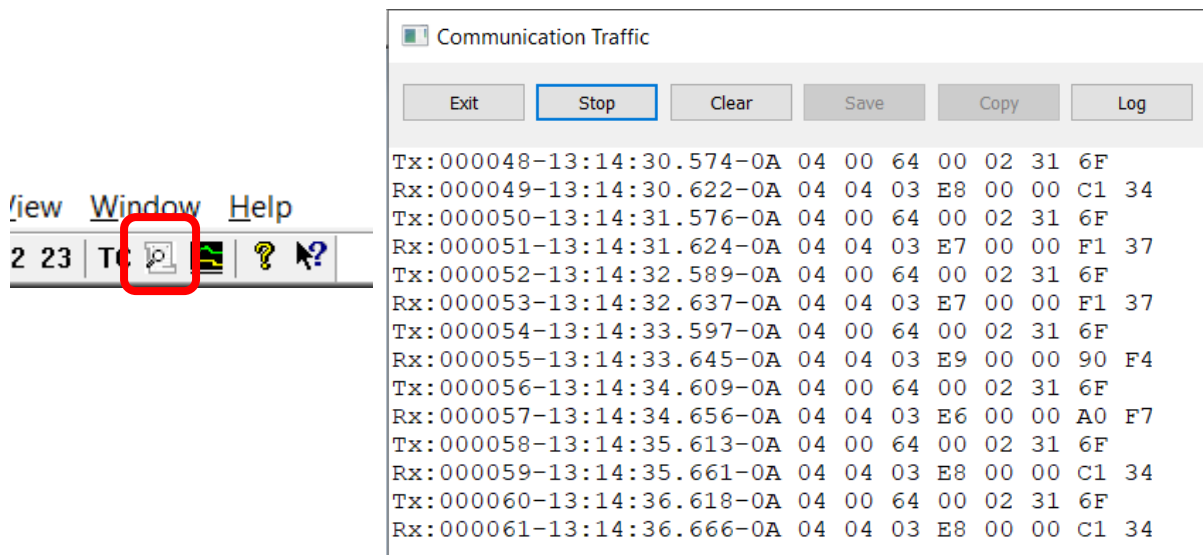


Figura AIV.9. Herramienta de monitoreo del tráfico de datos en la red

Para evidenciar una comunicación defectuosa se tienen dos indicadores principales

En la barra de monitoreo se muestran las siguientes siglas:

- Tx: Transmisiones realizadas por parte del maestro.
- Err: Errores en la transmisión
- ID: Esclavo al que se apunta
- F: Función modbus
- ScanRate: Velocidad de escanéo.

La figura AIV.10 muestra un ejemplo de lo que sería una conexión fallida, teniendo así un total de 133 transmisiones por parte del maestro y 133 errores en dichas transmisiones, adicionalmente en la barra de monitoreo se resalta en rojo el evento por el cual se registró el error al momento de la comunicación.

```
Tx = 133: Err = 133: ID = 10: F = 04: SR = 1000ms
Timeout error
```

Figura AIV.10. Ejemplo de conexión fallida por falta de respuesta del esclavo

La segunda forma de evidenciar errores en la red es mediante la herramienta de monitoreo, ya que, para el ejemplo antes presentado cuando el esclavo no responde se tiene el monitoreo que se muestra en la figura AIV.11, donde no se tiene una recepción de datos Rx.

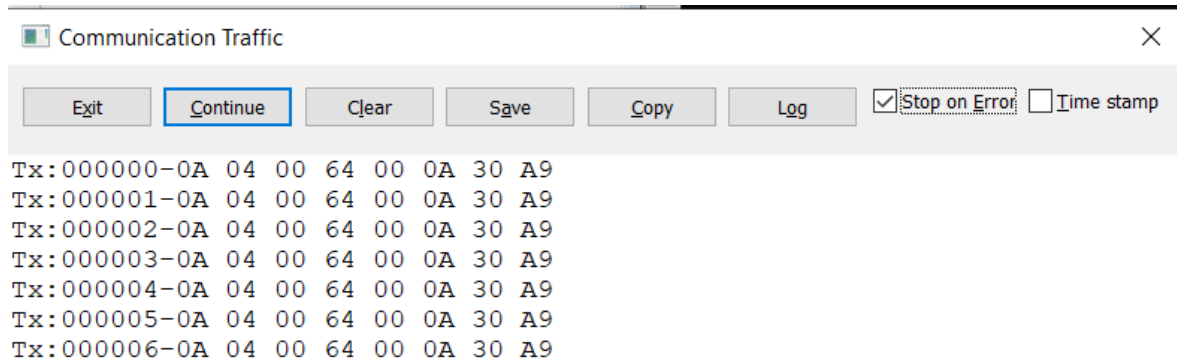


Figura AIV.11. Conexión fallida en herramienta de monitoreo de red.

Cabe aclarar que el manual aquí presentado se detalla en función de la utilidad que se le dio durante el desarrollo del prototipo, no obstante, Modbus Poll hace integra muchas más opciones y facilidades para el manejo de las diferentes tramas modbus, para ello cualquier información adicional puede ser consultada en el manual de usuario propio del software, en la dirección: [Manual de usuario de Modbus Poll \(modbustools.com\)](http://modbustools.com) [36].

5.5 ANEXO V: Códigos de errores en el bloque MSG

MSG Instruction Error Codes

When the processor detects an error during the transfer of message data, the processor sets the ER bit and enters an error code that you can monitor from your programming software.

Error Code	Description of Error Condition
02H	Target node is busy. NAK No Memory retries by link layer exhausted.
03H	Target node cannot respond because message is too large.
04H	Target node cannot respond because it does not understand the command parameters OR the control block may have been inadvertently modified.
05H	Local processor is off-line (possible duplicate node situation).
06H	Target node cannot respond because requested function is not available.
07H	Target node does not respond.
08H	Target node cannot respond.
09H	Local modem connection has been lost.
0BH	Target node does not accept this type of MSG instruction.
0CH	Received a master link reset (one possible source is from the DF1 master).
0FH	DCOMM button was activated while an ASCII instruction was waiting to execute.
10H	Target node cannot respond because of incorrect command parameters or unsupported command.
12H	Local channel configuration protocol error exists.
13H	Local MSG configuration error in the Remote MSG parameters.
15H	Local channel configuration parameter error exists.
16H	Target or Local Bridge address is higher than the maximum node address.
17H	Local service is not supported.
18H	Broadcast is not supported.
20H	PCCC Description: Host has a problem and will not communicate.
21H	Bad MSG file parameter for building message.
30H	PCCC Description: Remote station host is not there, disconnected, or shutdown.
37H	Message timed out in local processor.
39H	Local communication channel reconfigured while MSG active.
3AH	STS in the reply from target is invalid.
40H	PCCC Description: Host could not complete function due to hardware fault.
45H	MSG reply cannot be processed. Either Insufficient data in MSG read reply or bad network address parameter.
50H	Target node is out of memory.
60H	Target node cannot respond because file is protected.
70H	PCCC Description: Processor is in Program Mode.
80H	PCCC Description: Compatibility mode file missing or communication zone problem.
81H	Modbus Error 1: Illegal Function
82H	Modbus Error 2: Illegal Data Address
83H	Modbus Error 3: Illegal Data Value
84H	Modbus Error 4: Slave Device Failure
85H	Modbus Error 5: Acknowledge
86H	Modbus Error 6: Slave Device Busy
87H	Modbus Error 7: Negative Acknowledge
88H	Modbus Error 8: Memory Parity Error
89H	Modbus Error: Non-standard reply. Actual code returned can be found in the upper byte of sub-element 22.

Error Code	Description of Error Condition
90H	PCCC Description: Remote station cannot buffer command.
B0H	PCCC Description: Remote station problem due to download.
C0H	PCCC Description: Cannot execute command due to active IPBs.
D0H	No IP address configured for the network, or Bad command - unsolicited message error, or Bad address - unsolicited message error, or No privilege - unsolicited message error, or Multihop messaging cannot route request
D1H	Maximum connections used - no connections available.
D2H	Invalid internet address or host name.
D3H	No such host exists.
D4H	Cannot communicate with the name server.
D5H	Connection not completed before user-specified timeout.
D6H	Connection timed out by the network.
D7H	Connection refused by destination host.
D8H	Connection was broken.
D9H	Reply not received before user-specified timeout.
DAH	No network buffer space available.
DBH	Multi-hop messaging CIP message format error.
DCH	Class 3 CIP connections are duplicated for same IP address.
DFH	Multi-hop messaging has no IP address configured for network.
E0H	Expansion I/O Communication Module Error. The error code returned can be found in the upper byte of sub-element 22.
E1H	PCCC Description: Illegal Address Format, a field has an illegal value.
E2H	PCCC Description: Illegal Address format, not enough fields specified.
E3H	PCCC Description: Illegal Address format, too many fields specified.
E4H	PCCC Description: Illegal Address, symbol not found.
E5H	PCCC Description: Illegal Address Format, symbol is 0 or greater than the maximum number of characters support by this device.
E6H	PCCC Description: Illegal Address, address does not exist, or does not point to something usable by this command.
E7H	Target node cannot respond because length requested is too large.
E8H	PCCC Description: Cannot complete request, situation changed (file size, for example) during multi-packet operation.
E9H	PCCC Description: Data or file is too large. Memory unavailable.
EAH	PCCC Description: Request is too large; transaction size plus word address is too large.
EBH	Target node cannot respond because target node denies access.
ECH	Target node cannot respond because requested function is currently unavailable.
EDH	PCCC Description: Resource is already available; condition already exists.
EEH	PCCC Description: Command cannot be executed.
EFH	PCCC Description: Overflow; histogram overflow.
F0H	PCCC Description: No access.
F1H	Local processor detects illegal target file type.
F2H	PCCC Description: Invalid parameter; invalid data in search or command block.
F3H	PCCC Description: Address reference exists to deleted area.
F4H	PCCC Description: Command execution failure for unknown reason; PLC-3 histogram overflow.
F5H	PCCC Description: Data conversion error.
F6H	PCCC Description: The scanner is not able to communicate with a 1771 rack adapter. This could be due to the scanner not scanning, the selected adapter not being scanned, the adapter not responding, or an invalid request of a "DCM BT (block transfer)."
F7H	PCCC Description: The adapter is not able to communicate with a module.
F8H	PCCC Description: The 1771 module response was not valid size, checksum, etc.

Error Code	Description of Error Condition
F9H	PCCC Description: Duplicated Label.
FAH	Target node cannot respond because another node is file owner (has sole file access).
FBH	Target node cannot respond because another node is program owner (has sole access to all files).
FCH	PCCC Description: Disk file is written protected or otherwise inaccessible (off-line only).
FDH	PCCC Description: Disk file is being used by another application; update not performed (off-line only).
FFH	Local communication channel is shut down.

5.6 ANEXO VIII: Manual de Usuario

El siguiente anexo tiene por finalidad presentar al operador una guía completa sobre el uso del componente desarrollado para su utilización, cabe mencionar, que al ser un prototipo puede ser mejorado para añadirle funcionalidades más avanzadas, no obstante, como cualquier otro dispositivo electrónico puede requerir cambio de piezas a futuro, razón por la cual, los mantenimientos en mención se deben hacer una vez que se haya comprendido tanto el diseño como el funcionamiento del componente, para así poder garantizar la continuidad de su funcionamiento y que las partes que se reemplacen sean aptas para el dispositivo.

VIII.1 DESCRIPCIÓN DEL PRODUCTO

El componente que se detalla es un prototipo de tarjeta de comunicación, que a partir de una conversión analógica digital proveniente de un sensor, cuya salida sea un valor de voltaje en un rango estandarizado de 0 – 10V, genere una trama Modbus RTU para que el campo Data contenga la lectura del sensor y de esta forma poder tener un dispositivo de campo adicional a la red Modbus.

VIII.2. ESPECIFICACIONES DEL PROTOTIPO

Tabla VIII.1. Especificaciones eléctricas del prototipo

Característica	Min	Typ	Max	Unidad
Voltaje de Alimentación	12	24	35	V _{DC}
Temperatura de almacenamiento	0	-	75	°C
Temperatura de operación	0	-	30	°C
Voltaje de entrada sensor analógica	0	-	15	V _{DC}
Corriente de Operación	-	200	-	mA
Valor 1L para comunicación RS-485 $V_B - V_A$	-	> 0.2	-	V _{DC}
Valor 0L para comunicación RS-485 $V_A - V_B$	-	> 0.2	-	V _{DC}
Valor indefinido para comunicación RS-485	-	< 0.2	-	V _{DC}
Resistencia terminadora	114	120	126	Ω
Máximo voltaje pico reverso	-	50	-	V _{DC}
Máxima corriente de alimentación	-	1	-	A
Máxima corriente a la entrada de lectura analógica			300	mA

VIII.3. DISTRIBUCIÓN DE PUERTOS

El prototipo se presenta en la figura VIII.1, correspondiente a las diferentes vistas del componente, así como la distribución de sus puertos considerando todas las terminales de entrada para un mejor entendimiento del operador.

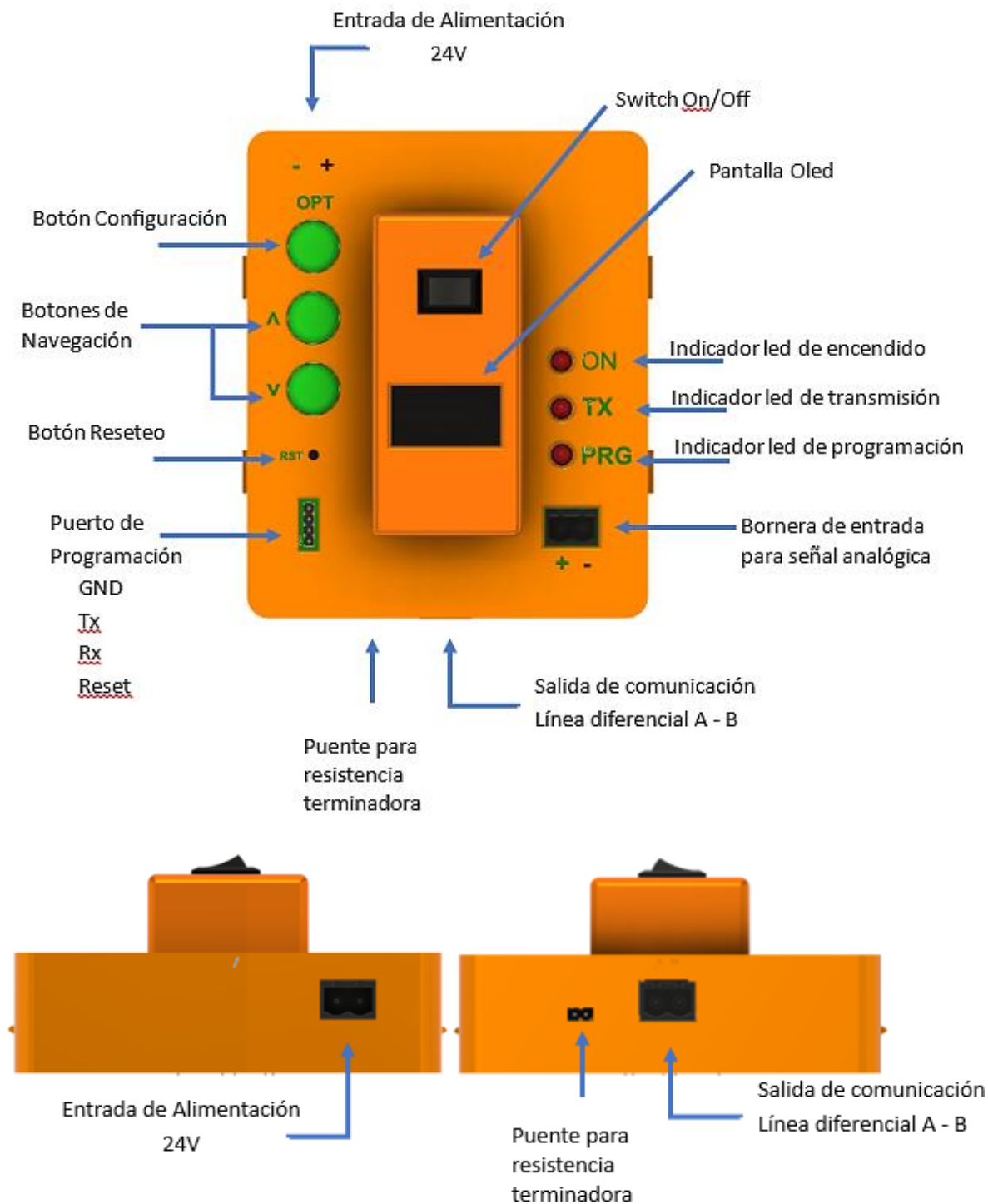


Figura VIII.1. Distribución de puertos y partes del prototipo

VIII.4. FUNCIONAMIENTO DEL PROTOTIPO

Como paso previo al funcionamiento del equipo, este debe estar energizado en sus terminales con 24V, una vez alimentado se debe cambiar el estado del switch On/Off para que circule energía por todo el prototipo, la correcta polarización del circuito y alimentación de este se evidencia con el encendido del indicador led amarillo, como lo describe la siguiente figura.



Figura VIII.2. Encendido del prototipo



El prototipo cuenta con las protecciones necesarias para evitar una falla en la polarización al momento de energizarla, no obstante, es importante considerar siempre la parte positiva y negativa de la fuente para evitar daños en el componente, y en caso de polarización errónea desacoplar el dispositivo de la fuente lo más rápido posible. Una buena señal de mala polarización es que el indicador led no se encenderá.

Una vez encendido el prototipo se mostrará en la pantalla los logotipos de la universidad y del departamento, es decir, se mostrará una pantalla inicial con el texto “Escuela Politécnica Nacional”, después de un par de segundos se proyectará el texto “Departamento de Automatización y Control Industrial”, de igual manera esta pantalla se proyectará un par de segundos, las pantallas se muestran en la figura VIII.3.



Figura VIII.3. Pantallas de Inicialización

Una vez transcurrido el tiempo de inicialización programado, la pantalla por defecto que siempre se mostrará, se presenta en la figura VIII.4, esta pantalla tiene los campos:

- **Slave ID:** Indica la dirección del prototipo con la cual se debe configurar el dispositivo maestro para hacerle solicitudes.
- **Registro ID:** Indica el registro del prototipo en el que se almacena el dato analógico.
- **MB Reg:** Indica el registro Modbus de 5 dígitos con base 1.
- **Voltaje:** Indica el voltaje leído en las terminales correspondientes.



Figura VIII.4. Pantalla por default


Existe una segunda pantalla a la cual se accede mediante la pulsación del botón de navegación con dirección hacia abajo, una vez que es presionado dicho pulsante, se tiene la pantalla secundaria que proporciona información adicional sobre el funcionamiento del prototipo.


- **Valor mínimo:** Voltaje mínimo enviado desde el maestro (aplicación para control por nivel alto y nivel bajo)

- **Valor máximo:** Voltaje máximo enviado desde el maestro (aplicación para control por nivel alto y nivel bajo)
- **Baud Rate:** Velocidad de transmisión de información



Figura VIII.4. Pantalla secundaria

	<p>Para que exista una correcta comunicación, todos los dispositivos en la red Modbus, o el maestro cuando la comunicación es uno a uno, deben tener la misma velocidad, caso contrario la comunicación será fallida, según las velocidades admitidas por el PLC son: 4800, 9600, 19200, 38400.</p>
---	---

	<p>Existen valores agregados por defecto a cada campo, cada que el dispositivo se reinicie o se apague NO ALMACENA la última configuración, sino que todos los parámetros se restablecen con los valores programados, en caso de requerir un cambio de valores por default, se debe acceder a la programación del microcontrolador.</p>
---	---


VIII.5. CONFIGURACIÓN DEL PROTOTIPO POR EL USUARIO

Para la configuración de parámetros, se debe presionar el botón “OPT” el cual hace referencia a las opciones a las que se puede tener acceso en un menú configurable, el acceso a dicho menú se lo realiza desde cualquiera de las pantallas antes presentadas, sea la pantalla por defecto o la pantalla secundaria.

Dentro de los parámetros de configuración se consideraron los de mayor incidencia en la comunicación, de esta forma fueron contemplados:

- Dirección del prototipo: Dirección del esclavo
- Registros
 - ADC: Registro que contiene el valor de la conversión AD.
 - Vmin: Registro que almacena el valor mínimo enviado por el maestro.
 - Vmax: Registro que almacena el valor máximo enviado por el maestro.
- Baud Rate: Velocidad de transmisión

Para el acceso a estos parámetros, como ya se mencionó, se deberá presionar el botón de configuración, y mediante los botones de navegación explorar cada una de las opciones antes detalladas, la forma de navegación se pensó para ser bastante intuitiva al operador, de forma que su configuración sea rápida y sencilla.

	<p>Al ser un botón de configuración de parámetros, la comunicación quedará interrumpida hasta que dichas configuraciones se realicen, y el usuario salga del menú configurable retornando a la pantalla por defecto, caso contrario si el usuario no confirma las configuraciones y se mantiene en alguna pantalla asociada a los parámetros de la comunicación, esta no se retomará.</p>
--	---

➤ **Identificador del esclavo (Esclavo ID)**

Por defecto este parámetro es el primero en visualizarse cuando se presiona por primera vez el botón de opciones, mostrando en pantalla todas las configuraciones, pero destacando este parámetro como se muestra en la figura VIII.5.



Figura VIII.5. Menú configurable parámetro Esclavo ID

Para acceder a la configuración de este parámetro se deberá presionar nuevamente el botón de opciones, con lo cual en la pantalla únicamente se visualizará el parámetro de interés, y el valor que actualmente posee como se visualiza en la figura VIII.6.



Figura VIII.6. Configuración de esclavo ID

Para asignarle un valor diferente al identificador único que tendrá el prototipo, se debe manipular los botones de navegación para subir y bajar el valor respectivamente, adicionalmente, en la programación se consideran redundancias de acuerdo con el protocolo ya que no permite asignar valores menores a 1, siendo este el valor más bajo que se le puede asignar al prototipo y 247 como el valor más alto.

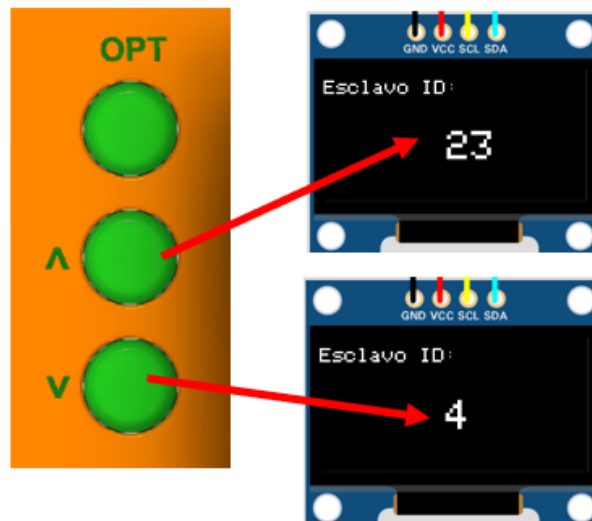



Figura VIII.7. Configuración de esclavo ID

Una vez configurada la dirección deseada que tendrá el prototipo, únicamente se deberá volver a presionar el botón de configuración que en este caso funciona como botón de confirmación, posterior a la ello, el prototipo volverá a mostrar la pantalla por defecto con la nueva dirección del esclavo.

	<p>Al cambiar la dirección del esclavo, se cambia únicamente en la programación interna del prototipo, por lo que, si dicha dirección no es modificada en el programa del maestro, se tendrá error de <i>Time Out</i> ya que el maestro está solicitando información a un esclavo que ya no se encuentra en la red.</p>
---	---

➤ Registros

El procedimiento para acceder a este parámetro es idéntico al parámetro anterior, se accede mediante el botón de opciones, con la diferencia que a partir de la figura VIII.5, se debe utilizar los botones de navegación para llegar a este parámetro, y de igual forma se enfatiza el parámetro con el símbolo ">>" para la completa seguridad del usuario al acceder al parámetro.



Figura VIII.8. Menú configurable parámetro Registros

Al presionar el botón opciones, se accede al parámetro con lo cual se despliega la imagen que se detalla en la figura VIII.9, misma que contiene un submenú con los tres registros que se utilizan, la navegación en este submenú es similar al del menú configurable, y de igual manera, se tiene el símbolo ">>" para resaltar el registro de interés a modificar. Por defecto, se presenta el registro ADC, y si se desea acceder al resto de registros se deberán utilizar los botones de navegación.




Figura VIII.9. Submenú configurable para registros

Cabe mencionar que en esta sección se modifican los registros contenedores de los valores tanto de entrada como de retención, para realizar dicho cambio, se utiliza la secuencia de configuración utilizada para el parámetro del Esclavo ID, se debe presionar el botón de opción en el registro deseado, con los botones de navegación se aumentará o disminuirá el valor preestablecido.



Figura VIII.10. Submenú configurable para registros

Para confirmar el valor de registro únicamente se deberá presionar, por última vez el botón de opciones/confirmación, y al hacerlo el programa retomará la pantalla por defecto, si el registro modificado es el del convertor ADC, este se visualizará en la pantalla en mención, mientras que, si los registros modificados corresponden a los valores mínimo y máximo, estos se visualizarán en la pantalla secundaria.

	<p>Al cambiar la dirección de cualquiera de los registros, estos se cambian únicamente en la programación interna del prototipo, por lo que, si las direcciones no son modificadas en el programa del maestro, se tendrán lecturas equívocas o fallará la comunicación.</p>
---	---

➤ **Baud Rate**

El procedimiento, para la configuración del Baud Rate es similar a los parámetros antes configurados, con la diferencia que ahora el usuario no puede aumentar ni disminuir la velocidad con los botones de navegación, una vez seleccionada la opción de baud rate en el menú configurable como se muestra en la figura VIII.11.



Figura VIII.11. Menú configurable parámetro Baud Rate

Presionando nuevamente el botón de opciones y que permite ingresar a cada uno de los parámetros, se ingresa a la configuración del Baud Rate, en este parámetro a diferencia de los anteriores, el usuario NO podrá aumentar o disminuir la velocidad de transmisión con los botones de navegación, en su lugar, el usuario podrá seleccionar una de cuatro opciones de velocidad, estas opciones fueron tomadas en función de las velocidades que acepta el PLC Micrologix 1100. De esta forma, el usuario podrá interactuar entre dichas opciones mediante los botones de navegación y al seleccionar la velocidad requerida se deberá pulsar nuevamente el botón de opciones para guardar la configuración, tal como se muestra en la figura VIII.12

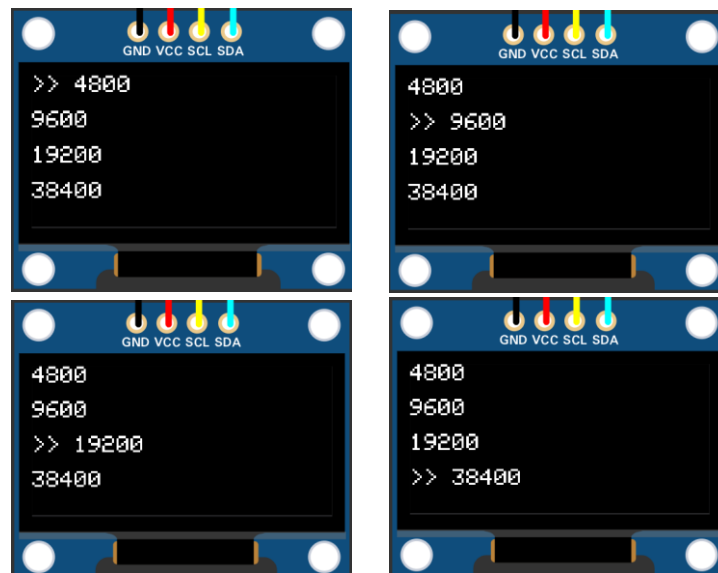


Figura VIII.12. Velocidades de transmisión disponibles

Comprendiendo la importancia que tiene la velocidad de transmisión en este prototipo, este parámetro es el único que integra un mensaje final al presionar el botón de confirmación (botón de configuración), que le indica al usuario la velocidad configurada para que este tenga la seguridad de la velocidad a la que se está manejando el tráfico de datos en el dispositivo, y con ello sea más rápida la detección de errores por diferencias entre las velocidades de otros dispositivos.



Figura VIII.13. Mensajes de confirmación para la velocidad

➤ Salir

Por último, se plantea una última opción que lejos de ser un parámetro configurable, es una opción para minimizar la acción de presionar el botón de acceso al menú de forma accidental, de igual manera que los otros parámetros, para confirmar esta opción únicamente basta con desplazarse hasta ella con los botones de navegación y presionar el botón de confirmación, una vez realizado este procedimiento, el prototipo mostrará nuevamente la pantalla por defecto y conservará los últimos cambios que se realizaron, reanudando la comunicación.



Figura VIII.14. Mensajes de confirmación para la velocidad

VIII.5. TRANSMISIÓN DE DATOS

En esta sección únicamente se presenta la figura VIII.15, la cual muestra el led indicador de color azul, dicho led tiene por finalidad parpadear conforme se transmitan datos, su función es parecida a la de los jacks ethernet, mismos que poseen dos leds, una para indicar que se encuentra energizado, lo cual ya está detallado en la sección de funcionamiento, y otro led que indica el intercambio de información, de esta forma se presenta la figura antes mencionada.



Figura VIII.15. Indicador de transmisión de datos

VIII.6. PROGRAMACIÓN DEL PROTOTIPO

Para la programación del microcontrolador que comanda el prototipo, se utilizará el software sobre el cual fue codificado dicho elemento electrónico, de esta forma, en caso de requerir cambiar la programación del microcontrolador, se debe utilizar la placa de desarrollo Arduino uno sin el microcontrolador, es decir únicamente la placa, y con ello conectar mediante cable dupont o cable jumper de tipo hembra, a la ranura de programación habilitada en el prototipo, esta ranura cuenta con 4 terminales importantes en la programación del microcontrolador tal como se detalla en la figura VIII.16.

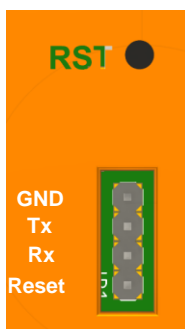


Figura VIII.16. Indicador de transmisión de datos

Por último, se añade un indicador led color rojo que parpadeará cuando el microcontrolador esté siendo codificado para descartar errores en la programación de este como se muestra en la figura VIII.17.

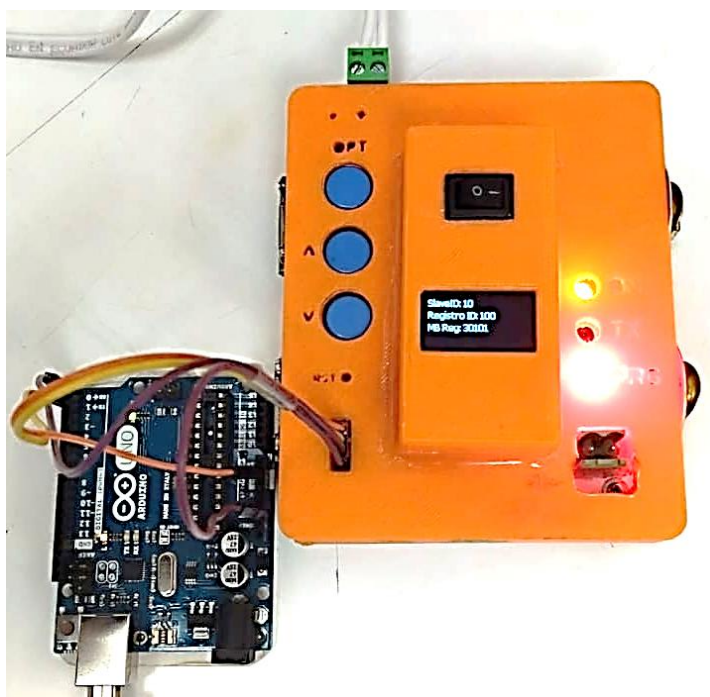


Figura VIII.17. Indicador de programación del microcontrolador

VIII.7. MANETENIMIENTO Y CALIBRACIÓN DEL PROTOTIPO

El prototipo diseñado, fue pensado para que sea fácilmente calibrado y en caso de fallas, el cerebro del microcontrolador sea fácilmente reemplazado, para ello a nivel de calibración, se considera únicamente el valor analógico que se adquiere por parte del sensor y que posteriormente es digitalizado. Para la calibración de este valor, se tiene un potenciómetro de precisión, el cual varía la resistencia en función de un tornillo giratorio que aumenta o disminuye la resistencia entre las terminales, para calibrar dicho elemento basta con tener la medida real y la media a la salida entre tierra y el pin central del potenciómetro con lo que se puede ajustar el valor que se tiene al valor real del sensor, como se muestra en la figura VIII.18.

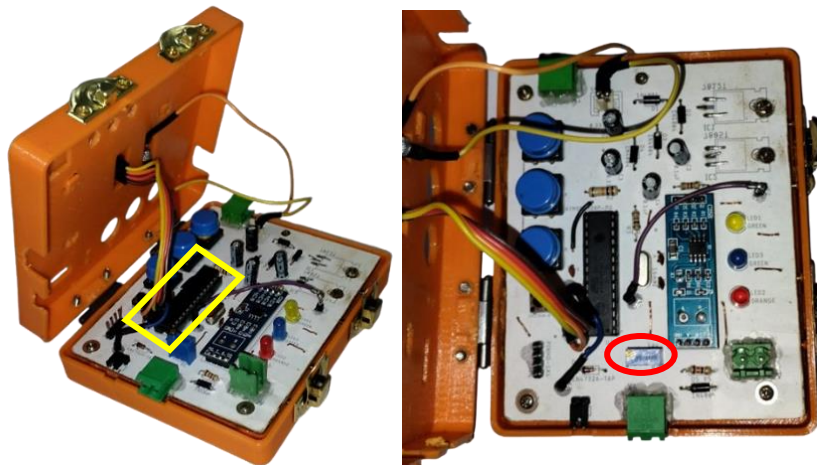


Figura VIII.18. Elementos de calibración y mantenimiento del prototipo.

Otra consideración en el mantenimiento del componente es la posibilidad de reemplazar el microcontrolador que figura como cerebro del prototipo, dicho componente se monta sobre un sócalo permitiendo así su fácil remoción e inclusive la programación externa sin hacer uso de los pines en el prototipo, sino que el microcontrolador puede ser retirado, programado en una placa Arduino Uno y posterior a ello reposicionarlo en su lugar y tener el mismo funcionamiento en el componente.



Para la programación del microcontrolador mediante el IDE de Arduino, es importante que dicho componente cuente con el bootloader de Arduino ya que, de no ser así, el software no reconocerá al microcontrolador.