

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

**DESARROLLO DE UN DISPOSITIVO DEVICENET PARA
ADQUISICIÓN DE DATOS Y DE UNA APLICACIÓN DE
MONITOREO DE PROCESOS INDUSTRIALES BASADA EN EL
SERVIDOR WEB DEL PLC ML-1400.**

**DISEÑO E IMPLEMENTACIÓN DE UN PROTOTIPO DE
DISPOSITIVO ANALÓGICO INDUSTRIAL PARA ADQUISICIÓN DE
DATOS Y MONITOREO DESDE UNA RED DEVICENET.**

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

MARCO VINICIO TIGSE RIVERA

Marco.tigse@epn.edu.ec

DIRECTOR: Dra. Ing. SILVANA DEL PILAR GAMBOA BENÍTEZ

silvana.gamboa@epn.edu.ec

DMQ, abril 2024

CERTIFICACIONES

Yo, Marco Vinicio Tigse Rivera 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.

MARCO VINICIO TIGSE RIVERA

Certifico que el presente trabajo de integración curricular fue desarrollado por MARCO VINICIO TIGSE RIVERA, bajo mi supervisión.

DRA.-ING. SILVANA DEL PILAR GAMBOA BENÍTEZ
DIRECTOR

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.

MARCO VINICIO TIGSE RIVERA

DRA.-ING. SILVANA DEL PILAR GAMBOA BENÍTEZ

DEDICATORIA

El presente trabajo va dedicado con mucho amor y cariño a mis padres, quienes me brindaron todo su apoyo incondicional durante todo el transcurso de mis estudios, quienes me enseñaron el valor de la humildad, dedicación, esfuerzo, honestidad, paciencia, el trabajo duro, que a pesar de no contar con todos los recursos sacrificaron mucho tiempo en mi desarrollo profesional y sobre todo me enseñaron a nunca darme por vencido en cumplir mis metas y sueños. Por tal motivo mi trabajo lo ofrezco como homenaje en representación a todos sus arduos esfuerzos y el tiempo brindado hacia mí.

AGRADECIMIENTO

Agradezco a Dios por permitirme llegar hasta estas instancias, por permitirme gozar de buena salud junto a toda mi familia, brindo mi más sincera gratitud a mis padres que fueron un pilar muy importante durante mi formación como profesional.

A mi madre María Rivera quien me motivo a seguir adelante, a nunca rendirme, a luchar por cumplir mis metas, sueños y sobre todo del valor de la honestidad, a mi padre José Tigse quien a pesar de todo a seguido aquí, quien me enseñó el valor la responsabilidad de cumplir un deber y siempre hacer las cosas con el mayor cuidado y dedicación.

A mi hermano José Luis, que me enseñó lo que es el trabajo duro, el sacrificio, el poder soportar largas horas de desvelo, malos días y quien fue una inspiración para no rendirme. A mi hermano Juan Carlos y a mi hermana Jessica Paulina, que a pesar de su carácter me brindaron su apoyo incondicional.

Expreso mi agradecimiento hacia todos mis amigos más confiables, Bryan, Carlos, Cristian, Daniel, David, Jonathan y Elvis quienes me acompañaron y supieron brindar su apoyo, consejos, guía, paciencia y sobre todo su más sincera y única amistad.

Agradezco a mi tutora Dra. Silvana Gamboa por toda su gran paciencia, comprensión y guía para poder realizar y cumplir con todos mis trabajos con el fin de poder lograr alcanzar mi más grande meta.

Brindo mi más sincero agradecimiento hacia varios maestros que estuvieron en todo mi proceso de formación profesional, pero de manera más cordial hacia Johnny Chuqui y Mateo Vásquez quienes me supieron orientar y apoyar en el desarrollo de mi proyecto.

Para finalizar quiero expresar mi gratitud a Tami que a pesar de haberte conocido en tan poco tiempo me ha sabido brindar su apoyo, paciencia, dedicación, compromiso, cariño y me recordó que la sinceridad ante todo es primordial para ser una mejor persona.

ÍNDICE DE CONTENIDO

CERTIFICACIONES	I
DECLARACIÓN DE AUTORÍA	II
DEDICATORIA	III
AGRADECIMIENTO	IV
ÍNDICE DE CONTENIDO	V
RESUMEN.....	VII
ABSTRACT.....	VIII
CAPÍTULO I.....	1
INTRODUCCIÓN	1
1.1 Objetivo general	1
1.2 Objetivos específicos.....	1
1.3 Alcance	1
1.4 Marco teórico.....	3
1.4.1 PROTOCOLO DEVICENET	3
1.4.1.1 PROTOCOLO CAN EN DEVICENET	4
1.4.1.2 PROTOCOLO CIP EN DEVICENET	4
1.4.2 ENVÍO DE DATOS EN LA RED DEVICENT	5
1.4.2.1 TRAMA DE DATOS.....	6
1.4.2.2 TRAMA CAN	6
1.4.2.3 IDENTIFICADOR CAN	8
1.4.2.4 MENSAJES E/S	9
1.4.3 HOJA DE DATOS ELECTRONICA (EDS)	9
1.4.4 IMPLEMENTACIÓN DE LA RED DEVICENET	10
CAPÍTULO II.....	12
2. METODOLOGÍA	12
2.1 REQUERIMIENTOS DE PROTOTIPO DE DISPOSITIVO ANALÓGICO	12
2.2 HERRAMIENTAS.....	13
2.2.1 SELECCIÓN DE DISPOSITIVOS Y ELEMENTOS.....	15
2.2.2 SELECCIÓN DE SOFTWARE.....	18
2.3 DISEÑO DE TARJETA DE PROTOTIPO DE DISPOSITIVO	18
2.4 IMPLEMENTACIÓN DE DISPOSITIVO EN LA RED	22
2.4.1 CONFIGURACIÓN DE PROTOTIPO DE DISPOSITIVO	23
2.4.1.1 LÓGICA DE PROGRAMACIÓN.....	23
2.4.1.2 ESTABLECIMIENTO DE COMUNICACIÓN	26
2.4.2 CONFIGURACIÓN DE RED EN RSNETWORX	28

2.4.2.1 CONFIGURACION DEL ARCHIVO EDS	33
2.4.3 CONFIGURACION DE PLC.....	38
2.5. VERIFICACIÓN DE CONEXIÓN A LA RED DE PROTOTIPO.....	41
CAPÍTULO III.....	43
RESULTADOS, CONCLUSIONES Y RECOMENDACIONES.....	43
3.1. RESULTADOS	43
3.1.1.PRUEBAS DE NIVEL DE VOLTAJE.....	44
3.1.2.PRUEBA DE LECTURA DE NÚMERO DE NODO	46
3.1.3.PRUEBA DE ESTABLECIMIENTO DE CONEXIÓN	48
3.1.4.PRUEBA DE RECONEXIÓN DE DISPOSITIVO EXPERIMENTAL	52
3.1.5.PRUEBA DE ENVIÓ DE DATOS DE SEÑAL ANALÓGICA POR MEDIO DE LA RED 53	
3.2. CONCLUSIONES.....	56
3.3. RECOMENDACIONES.....	57
REFERENCIAS BIBLIOGRÁFICAS	59
ANEXOS.....	61

RESUMEN

El presente documento se orienta en el diseño e implementación de un prototipo de dispositivo para la adquisición de datos analógicos y su correspondiente envío a través de una red basada en DeviceNet. Se parte por una breve explicación y definiciones acerca del protocolo DeviceNet, de los requerimientos necesarios para el correcto envío de datos tanto a nivel de hardware como software y de un análisis de la estructura de la trama de datos durante la de comunicación entre los dispositivos que conforman una red de DeviceNet.

Los puntos desarrollados parten del análisis de las tramas de datos de la red, para luego proceder al desarrollo del software y hardware del prototipo de sensor capaz de comunicarse con el dispositivo maestro de la red. Además, se desarrolla una hoja de datos electrónica (EDS) para el reconocimiento del dispositivo en la red DeviceNet. Así también se procede al diseño de una placa PCB orientada a la toma y envío de datos a altas velocidades en un ambiente industrial sujeta a perturbaciones como ruido.

El estudio finaliza al realizar diferentes tipos de pruebas de funcionamiento, tales como el desempeño del dispositivo implementado desde la inicialización de la red hasta el envío de datos hacia el PLC se efectúe de manera correcta. Adicional se prueba la operación de la red cuando el número de nodo es distinto al número de dispositivo configurado en la red, con la finalidad de poder comprobar que el prototipo puede volver a realizar el proceso de reconexión con la red.

PALABRAS CLAVE: Sensor, Prototipo, Configuración, Red DeviceNet, Trama de datos, Velocidad de transmisión.

ABSTRACT

This document is oriented to the design and implementation of a prototype device for the acquisition of analog data and its corresponding sending through a DeviceNet based network. It starts with a brief explanation and definitions about the DeviceNet protocol, the necessary requirements for the correct sending of data both at hardware and software level and an analysis of the structure of the data frame during the communication between the devices that make up a DeviceNet network.

The points developed start from the analysis of the data frames of the network, and then proceed to the development of the software and hardware of the prototype sensor capable of communicating with the master device of the network. In addition, an electronic data sheet (EDS) is developed for device recognition in the DeviceNet network. The design of a PCB board oriented to take and send data at high speeds in an industrial environment subject to disturbances such as noise is also carried out.

The study ends by performing different types of operational tests, such as the performance of the implemented device from the initialization of the network to the sending of data to the PLC is done correctly. Additionally, the operation of the network is tested when the node number is different from the device number configured in the network, in order to verify that the prototype can perform the reconnection process again.

KEYWORDS: Sensor, Prototype, Configuration, DeviceNet Network, Data Frame, Transmission Speed.

CAPÍTULO I

INTRODUCCIÓN

En el trabajo a realizar se propone el desarrollo e implementación de un dispositivo analógico para su uso en el laboratorio de Redes Industriales, como un prototipo para la adquisición y envío de datos, además su posible integración a una red que se basa en un protocolo industrial de DeviceNet.

1.1 Objetivo general

Diseñar e implementar un prototipo de dispositivo analógico industrial para la adquisición de datos y monitoreo desde una red DeviceNet

1.2 Objetivos específicos

Realizar la búsqueda bibliográfica con referencia al funcionamiento de dispositivos DeviceNet en el monitoreo y control de procesos industriales.

Establecer los requerimientos para el diseño e implementación de un prototipo de dispositivo de adquisición de datos analógico DeviceNet.

Diseñar un prototipo de dispositivo que cumpla con los requerimientos necesarios para la adquisición de datos analógico DeviceNet.

Seleccionar los elementos de hardware y el software necesarios para el desarrollo del dispositivo DeviceNet.

Implementar un prototipo de dispositivo que cumpla con los requerimientos para la adquisición de datos analógico DeviceNet.

1.3 Alcance

A. Fase Teórica

Se plantea realizar una recopilación bibliográfica sobre las características de los distintos dispositivos de hardware que son utilizados para la adquisición de datos en aplicaciones industriales.

Se realizará una síntesis bibliográfica acerca de la adquisición de datos orientados a una red DeviceNet, y de las características de la estructura de comunicación industrial desde el PLC CompactLogix con los dispositivos que se encuentran dentro de la red DeviceNet.

Se establecerá los requerimientos necesarios que debe cumplir el prototipo de adquisición de datos analógicos a través de una red DeviceNet.

Se realizará la selección del protocolo de comunicación que utilizará el prototipo de dispositivo de adquisición de datos a través de una red DeviceNet, para establecer la comunicación con el PLC CompactLogix

Se seleccionará la tarjeta embebida y de los dispositivos adicionales que permitan utilizar el protocolo de comunicación entre el prototipo de dispositivo de adquisición de datos analógicos con el PLC CompactLogix

Se seleccionará las distintas herramientas de software de acceso libre que posibiliten: la programación de la tarjeta embebida, comunicación entre el diseño de una PCB, la simulación de un proceso industrial para realizar el análisis del comportamiento del prototipo de dispositivo de adquisición de datos analógicos.

B. Fase de diseño

Se realizará el diseño del hardware del prototipo, incluyendo las diferentes etapas de acondicionamiento de señales de entrada analógicas y de alimentación del dispositivo en conjunto con la selección de todos los elementos electrónicos necesarios para implementar la comunicación mediante el protocolo DeviceNet.

Se realizará el diseño del software del prototipo de dispositivo de adquisición de datos analógicos, donde se incluye toda la programación con los requisitos necesarios para la adquisición de datos, procesar la información, establecer comunicación con la red con la posibilidad de un intercambio de datos bidireccional y una vez integrado en la red la comunicación es de manera unidireccional el esclavo únicamente puede enviar los datos de la señal analógica.

C. Fase de implementación

Se realizará la implementación del prototipo de dispositivo de adquisición de datos analógicos que se ajusten a los criterios de diseño de la sección anterior.

Se realizará la programación y la configuración necesaria en base a los criterios de programación de la tarjeta embebida para establecer la comunicación del dispositivo con la red DeviceNet.

D. Fase de validación/análisis de resultados/pruebas de funcionamiento

Se realizarán las correspondientes pruebas mediante la implementación de un proceso industrial que requiera del uso de un sensor tipo analógico para la medición ya sea de nivel, velocidad o temperatura donde se verificará el adecuado funcionamiento del prototipo mediante la adquisición de los datos analógicos de voltaje y su correspondiente visualización en la interfaz del PLC por medio de la comunicación con la red DeviceNet.

1.4 Marco teórico

En la sección actual se brinda información teórica necesaria para entender el problema planteado y su correspondiente solución de poder integrar un nuevo dispositivo en la red DeviceNet. Por tal motivo es necesario realizar la descripción e investigación del protocolo DeviceNet enfocando su uso en ambientes industriales.

Además, se revisan los protocolos que conforman la capa física, así como la capa de aplicación, del proceso de envío y recepción de datos en la red y sobre todo de la configuración de la trama de datos que debe manejar un dispositivo DeviceNet para lograr la comunicación en la red.

1.4.1 PROTOCOLO DEVICENET

El protocolo de comunicación DeviceNet tiene sus inicios en el año de 1993 por el fabricante Allen Bradley y pertenece a una red de comunicación industrial que integra la estructura de red Netlinx. Al tratarse de una red abierta permite la integración de nuevos dispositivos, además que facilita la comunicación entre redes que manejan el protocolo CIP en el nivel de aplicación, tales como ControlNet y EtherNet/IP [1][2].

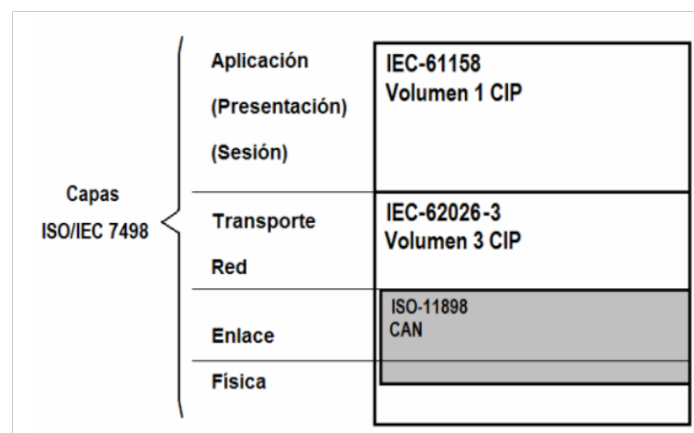


Figura 1.1. Estandarización de DeviceNet [7]

Las redes DeviceNet se basa en el protocolo CIP (Common Industrial Protocol) con la finalidad de poder proporcionar una conectividad entre dispositivos industriales sencillos como sensores y actuadores con dispositivos de mayor complejidad (PLCs) de manera

simple, al brindar comunicación directa, sin la necesidad de contar con un módulo E/S para cada dispositivo [2][3].

DeviceNet usa un protocolo de capa de aplicación donde todos los datos son transmitidos mediante mensajes que contienen la información específica en bytes. Es una comunicación se concreta únicamente entre dispositivos que cumplen con un grupo de requisitos. Los mensajes en este tipo de red pueden contener un byte para determinar un número de clase, tipo de dato, direcciones de nodos y la cantidad de mensajes a recibir [3].

1.4.1.1 PROTOCOLO CAN EN DEVICENET

DeviceNet utiliza el protocolo CAN (Controller Area Networking) como capa física y de enlace de datos, siendo una comunicación serial que utiliza el estándar ISO 11898, limitando la velocidad de comunicación a un máximo de 1 mega y un mínimo de 125 Kbps, pero con la ventaja de contar con una baja sobrecarga de información dentro de la red [3] [7].

En el protocolo CAN los datos se transmiten mediante una trama o frame de datos que contiene en su estructura un campo denominado identificador con un número máximo de 11 bits y de un campo de datos de hasta 8 bytes, cada campo de datos puede contener hasta 8 bits de información [4].

El protocolo CAN permite la selección de mensaje de datos de mayor importancia o prioridad sin sufrir una deficiencia en el ancho de banda de la red, esto posibilita que los datos nunca entren en conflicto y que la secuencia en la red de mantenga, además que al utilizar una red tipo CAN cualquier dispositivo conectado a la red logra establecer comunicación e intentar transmitir datos siguiendo un determinado orden en el tiempo [3].

1.4.1.2 PROTOCOLO CIP EN DEVICENET

El protocolo CIP (Common Industrial Protocol) se utiliza a nivel de capa de aplicación del modelo OSI y se caracteriza porque no requiere de usar una estructura compleja para poder interactuar con dispositivos sencillos de los niveles inferiores de la pirámide de automatización, esto permite una estructura estándar para la comunicación entre diferentes niveles de redes con la finalidad de eliminar los obstáculos de transmisión de datos que parten desde los niveles inferiores de un sistema de control industrial [5] [6].

DeviceNet al utilizar el protocolo CIP garantiza que se tenga una caracterización común a las demás redes que pertenecen a la denominada familia CIP, ordenada de acuerdo al número de volumen como se puede visualizar en la Figura 1.2 [7].

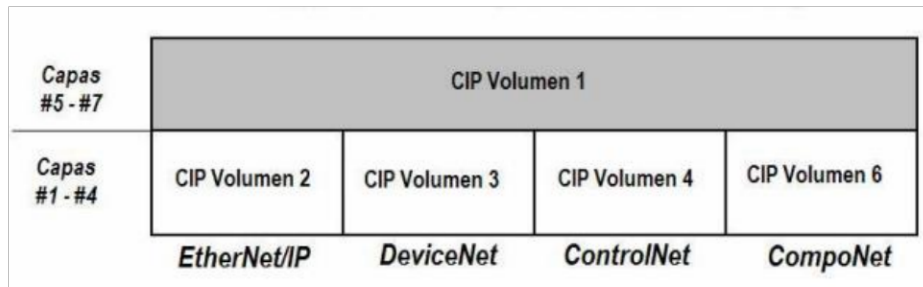


Figura 1.2. Diagrama de familias pertenecientes a redes CIP [7]

El protocolo CIP que comprende el volumen 3 donde se encuentra DeviceNet, especifica la adaptación de protocolo a la capa de enlace que maneja CAN con la finalidad de poder unir ambos protocolos cada uno actuando de manera conjunta [6].

1.4.2 ENVÍO DE DATOS EN LA RED DEVICENT

La trama de datos que maneja DeviceNet es el resultado de la unión de los protocolos CAN y de CIP este último permite definir un formato para lograr la identificación de los objetos que representan la información de los dispositivos que se encuentran en la red, de esta manera pueden ser clasificados acorde a su clase, instancia y atributos [3].

CAN establece dos niveles lógicos para la capa física y de enlace de datos, un estado de 1 lógico (Recesivo) y un estado de 0 lógico (Dominante) para determinar el acceso a la red esto se puede visualizar en la Figura 1.3. Los datos son enviados y recibido mediante los canales CAN H y CAN L a manera de niveles de voltaje diferencial, esto permite tanto asegurar que los datos no se pierdan en la red y la reducción del ruido [5][8].

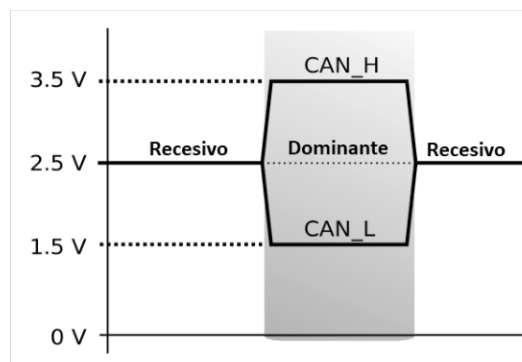


Figura 1.3. Representación de estados de uno y cero lógicos [9]

Un estado dominante prevalece frente a un estado recesivo esto como medida de protección para evitar colisiones de información en la red, dado el caso que si varios nodos intentan transmitir información. El estado determina si el número de identificación es menor o mayor, si es el segundo caso, el nodo deja de transmitir debido a que detecta la colisión

con el identificador menor siendo el de mayor importancia e intenta realizar la transmisión de datos una vez finalizada la comunicación con el nodo de identificación inferior [9].

1.4.2.1 TRAMA DE DATOS

Como se mencionó en el apartado 1.4.2 al utilizar el formato del bus CAN, la trama de datos debe acoplarse a DeviceNet, para ello CAN establece cuatro tipos de tramas con la finalidad de poder integrar una correcta comunicación entre dispositivo, entre ellos se tiene los siguientes:

- Trama de datos: Es el tipo de configuración que utiliza DeviceNet que cuenta con toda la información de los datos que se envían a través de la red [5]. Los datos se encuentran ordenados en 7 distintos campos como son el caso para inicio de comunicación, para control, dato, redundancia de errores, arbitraje de trama y para fin de comunicación
- Trama remota: Esta configuración se da cuando es necesario que otro dispositivo transmita la información de trama de datos donde se incluye el mismo identificador [5] [10].
- Trama de error: Este tipo de trama permite la detección de algún error producido al momento de realizar la transmisión de datos. Donde los errores son transmitidos en la red con la finalidad de poder establecer estados de alerta. Los errores más comunes son: warning, error passive y bus off este último coloca al dispositivo en un estado en el cual no podrá enviar ni recibir información de la red [5] [10].
- Trama de sobrecarga: Permite establecer retardos con el objetivo de proporcionar mayor tiempo entre cada ciclo de transmisión de cada trama de datos [5].

1.4.2.2 TRAMA CAN

La trama CAN cuenta con dos tipos de formatos para la transmisión de datos, el formato de trama estándar y extendido, la diferencia radica en la longitud de datos para el identificador, para el caso de DeviceNet se usa una trama estándar tal y como se puede visualizar en el Figura [9].

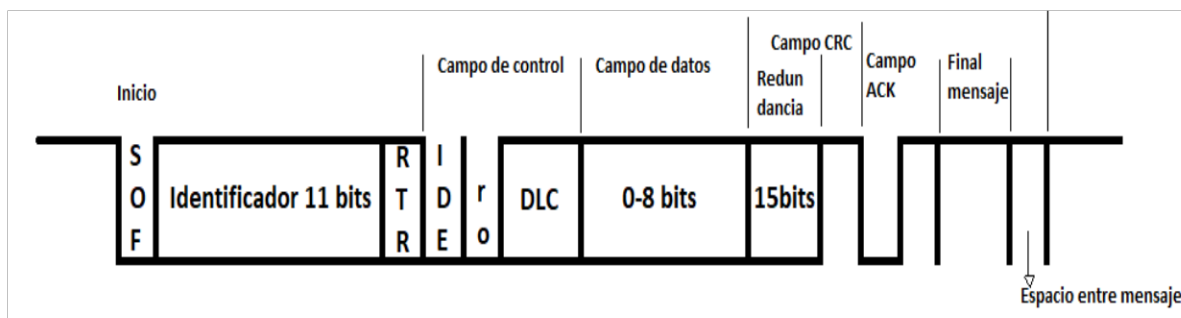


Figura 1.4. Trama CAN estándar para DeviceNet [10]

En la siguiente tabla 1.1 se muestra el formato de trama estándar para la red de DeviceNet:

Tabla 1.1. Formato de trama de datos para DeviceNet

Campo	Número de bits	Descripción
Trama de Inicio	1	Puede estar en cero o uno lógico, inicia la transmisión de la trama CAN [5].
ID (Identificador CAN)	11	Corresponde a un número de código único para diferenciar cada uno de los dispositivos que se cuentan en la red, además que proporciona la prioridad de la trama [5] [12].
RTR (Transmisión remota)	1	Los dos estados del bit representan si esta se encuentra en estado dominante, que es usado para trama de datos y en el estado recesivo para solicitud de transmisión remota [12]
IDE (extensión para el identificador ID)	1	Para el caso de formato estándar el estado del bit IDE es de carácter Dominante, pertenece al campo de control [12]
R0 (bit reservado)	1	Es un bit reservado que se encuentra en estado Dominante, pertenece al campo de control [12].
DLC (longitud de datos)	4	Representa el número de datos en mensajes a enviar, puede ir desde 0 hasta un valor de 8 mensajes en byte, pertenece al campo de control [12]
Datos	0-63	Se encuentra dividido en secciones de 8 bytes [5]
Redundancia cíclica	15	Bits asignados para la revisión de redundancia de errores [5]

Delimitador CRC	1	Le bit se debe encontrar en estado recesivo [5]
Chequeo de errores	2	El primer bit corresponde al bit ACK mientras que el segundo representa el delimitador ACK que se encuentra en estado recesivo [5] [12].
Fin de comunicación	7	Determina si la comunicación de la información de los datos a terminado [5].

1.4.2.3 IDENTIFICADOR CAN

DeviceNet utiliza el identificador CAN que se encuentra dividido en dos partes, con la finalidad de poder contar con un formato de comunicación segura y que los datos sean dirigidos directamente hacia el dispositivo deseado, como se detalló en la Tabla 1.1 los bits a usar son 11 [5].

En la Tabla 1.2 se presenta la división en secciones de los 11 bits del identificador CAN para DeviceNet.

Tabla 1.2. Clasificación de grupo de mensajes según el identificador [5]

Identificador ID											Tipo grupo acorde al mensaje
10	9	8	7	6	5	4	3	2	1	0	
0	Message ID				F. MAC ID						Grupo 1
1	0	F. MAC ID					Message ID				Grupo 2
1	1	Message ID			F. MAC ID						Grupo 3
1	1	1	1	1	Message ID						Grupo 4
1	1	1	1	1	1	1	x	x	x	x	Identificador inválido

Para el caso de DeviceNet la sección de Message ID pertenece a la identificación del tipo de mensaje, esta sección puede variar dependiendo del grupo al que pertenece el ID del dispositivo, mientras que la sección de F.MAC ID representa la dirección del número de nodo en el que se encuentra el dispositivo para la comunicación. El número de nodo se cuenta limitado en un rango de 0-63 siendo el 0 la dirección de nodo del dispositivo maestro [5][3].

Los mensajes del grupo 1 definen el intercambio de datos en carácter prioritario, se evita la colisión de datos al comparar primero el Message ID y luego el MAC ID, el dato con menor Message ID es el primero en ser transmitido en la red [5].

Los mensajes del grupo 2 establece la comunicación predefinida como Maestro/Esclavo, permite la conexión entre un dispositivo asignado como maestro dentro de la red con varios esclavos, por lo general para dispositivos simples como sensores y actuadores se usa el grupo 2 para establecer la comunicación, la prioridad del mensaje viene dada primero por el MAC ID y luego por el Message ID [5].

1.4.2.4 MENSAJES E/S

Los mensajes E/S en DeviceNet permiten establecer el tipo de datos a recibir o enviar, desde el dispositivo Maestro (Scanner) y los dispositivos esclavos que se encuentran en la red, estos pueden ser POLLED, CICLICO, COS y Bit-Estroboscopico (Strobed) [5][3].

Los mensajes tipo POLLED son de carácter solicitud/ respuesta, dicho mensaje se envía entre dispositivo maestro y esclavo para tener una conexión deseada, los mensajes son enviados a velocidades a elección del maestro [3].

Los mensajes tipo CICLICO son mensajes que se envían desde el dispositivo esclavo hacia el maestro cada cierto tiempo con una velocidad ya establecida, este tipo de mensajes permiten verificar el estado de los datos de manera continua [3].

Los mensajes tipo COS permiten la comunicación entre el maestro y el esclavo únicamente cuando se produce un cambio de estado de los datos que son transmitidos en el mensaje [1][3].

Finalmente, los mensajes tipo Strobed son únicamente para datos de entrada hacia el maestro, no se puede realizar un envío de datos hacia el esclavo, la configuración tipo Strobed no permite realizar un control sobre el esclavo únicamente verificar sus datos [1].

1.4.3 HOJA DE DATOS ELECTRONICA (EDS)

Un EDS (Electronic Data Sheet) es un archivo que contiene toda la información acerca del dispositivo que se encuentra integrado en la red, como el nombre del fabricante, número de serie, tipo de dispositivo y el tipo de datos que maneja en Mensajes E/S tanto para entrada o salida [1] [5].

Los archivos EDS se encuentran configurados en formato ASCII, cada archivo se diseña por cada fabricante de los dispositivos que sean compatibles con la red de DeviceNet, además permite realizar la configuración más sencilla del dispositivo durante el diseño de la red al permitir la selección de los tipos de datos [1][5].

En la siguiente Tabla 1.3 se especifican las secciones más importantes en el que se encuentra estructurado un EDS:

Tabla 1.3. Descripción de secciones relevantes de una hoja electrónica EDS

Sección	Descripción
File	Proporciona información tales como fecha y hora de creación del documento, nombre del dispositivo
Device	Contiene el nombre del fabricante del dispositivo, nombre de asignación para la red DeviceNet, tipo de dispositivo y el número de serie. Además, permite vincular el EDS con su correspondiente dispositivo físico [5].
Device Classification	Dicha sección no es de carácter obligatorio cuando el dispositivo se conecta a una red DeviceNet, contiene la información con el tipo de red que es compatible el dispositivo [5].
IO_info	Se especifica el tipo de dato en formato E/S , del tamaño de datos a enviar o recibir y de la descripción de cada uno de las funciones que cumple un bit [5]
Varian_IO_Info	Permite establecer configuraciones distintas a las establecidas en la sección de IO_info [5].
Parameters	Contiene la configuración necesaria que se realiza en el dispositivo al momento de agregarlo a la red [5].

1.4.4 IMPLEMENTACIÓN DE LA RED DEVICENET

La red DeviceNet se implementa con una topología tronco/derivación lo que permite que la señal de transmisión y recepción de datos se distribuya en toda la red, además que la línea

de alimentación para cada dispositivo pueda ir dentro del mismo cable de datos, esto facilita la instalación y reduce el cableado en una red industrial [4].

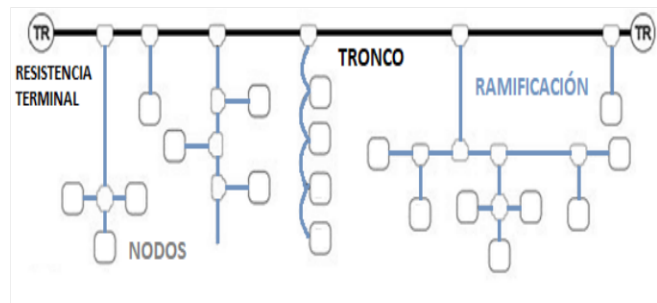


Figura 1.5. Ejemplo de topología de una red DeviceNet [5]

Una red DeviceNet se compone por la ramificación de los distintos dispositivos (nodos) conectados a una sola línea principal (línea troncal), las resistencias terminales únicamente son colocadas en los extremos de la línea principal de transmisión de datos, como se visualiza en la Figura 1.5. Así también en la Tabla 1.4 se describen las características relevantes de la implementación de la red.

Tabla 1.4. Las características de una red DeviceNet se presentan en la Tabla:

Característica	Descripción:
Alimentación	La red cuenta con una alimentación en la línea principal de 24V DC para todos los dispositivos
Cable	Redondo/Plano Dos hilos para alimentación Dos hilos para comunicación Malla para ruido
Velocidad	125-250-500 Kbps
Distancia	500-250-100 [m] para tamaño total de la red 6 [m] entre cada nodo 156-78-39 [m] total de la ramificación
Resistencias Terminales	120 Ohm cada una conectada a los extremos inicial y final de la red para reducir las reflexiones de señal
Nodos	64 número máximo permitidos para la red partiendo desde 0-63

CAPÍTULO II

2. METODOLOGÍA

El proceso de diseño e implementación del prototipo de dispositivo analógico para una red DeviceNet, se especifica en el presente capítulo, donde se considera los requerimientos tanto de los elementos electrónicos como del software a utilizar, además de los criterios de diseño de una placa, con la finalidad de cumplir con todos los requisitos necesario tanto a nivel de hardware como a nivel de programación para poder realizar una correcta integración de datos en una red DeviceNet.

2.1 REQUERIMIENTOS DE PROTOTIPO DE DISPOSITIVO ANALÓGICO

El prototipo de dispositivo tiene que ser capaz de realizar la lectura de un dato analógico enviado por medio de un sensor con salida de voltaje en un rango de 0 a 10 voltios. La señal debe ser adquirida por el dispositivo y enviados a través de la red DeviceNet hacia el PLC. El dispositivo únicamente envía datos mas no puede recibir algún comando de orden por parte del PLC.

Los requerimientos del dispositivo se establecen acorde a la operación que debe cumplir en la red DeviceNet siendo estos los siguientes:

- **Comunicación con protocolo CAN:** DeviceNet al utilizar el protocolo CAN como interfaz de física de comunicación, los dispositivos a seleccionar deben poder realizar la comunicación con el maestro(Scanner), de interpretar la trama de datos recibidas al momento de establecer comunicación, y de igual manera de enviar datos codificados en el protocolo CAN y CIP.
- **Velocidades de comunicación:** DeviceNet puede operar con tres velocidades diferentes de comunicación dependiendo de la longitud total de la red, para el caso de la red disponible en el laboratorio de redes Industriales la velocidad de operación es de 125Kbps establecido por el scanner, el dispositivo debe poder realizar la lectura y envío de datos a dicha velocidad sin saturar la red.
- **Características físicas de conexión:** Una red DeviceNet cuenta con su propia alimentación de 24V DC, dos cables para comunicación, transmisión de datos y que lo hace mediante el uso de voltajes diferenciales no mayores a 3.5 V DC, con un voltaje mínimo de 1.5 V DC y la correspondiente malla para eliminar ruidos

externos, además al ser un prototipo el dispositivo implementado en la red se puede ubicar en cualquier número de nodo de la red física.

2.2. HERRAMIENTAS

Los elementos tanto hardware como software deben cumplir con los distintos requerimientos para un correcto funcionamiento, tanto en la lectura de la entrada analógica del dispositivo, como del envío de los datos a través de la red DeviceNet, cada una de las herramientas a seleccionar deben ser compatibles con el protocolo de comunicación de la red.

Según las características y requerimientos especificados con anterioridad se cuenta con las siguientes tarjetas de desarrollo:

STM32F103C8T6: es una tarjeta de desarrollo de la familia STM32-basse, cuenta con dos pines dedicados a la comunicación de protocolo Can, se alimenta con 3.3 V, además que cuenta con un oscilador de 72MHz, por lo general es utilizado para proyectos donde no se requiere de unas grandes cantidades de procesamiento, los tipos de comunicación que puede soportar la tarjeta mencionada se las puede visualizar en la figura 2.1.

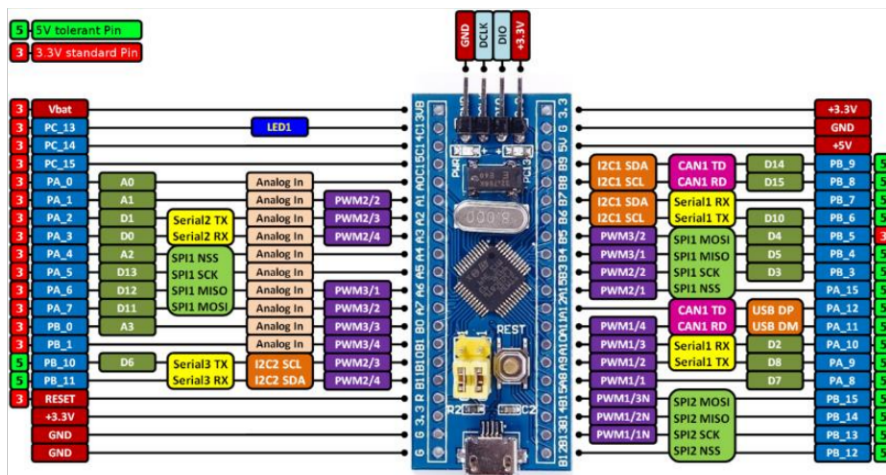


Figura 2.1. Descripción de funciones de cada uno de los pines de tarjeta de desarrollo STM32F103C8T6 [13]

La tarjeta STM32 para la comunicación Can requiere de un dispositivo adicional para poder operar con voltaje diferencial en los pines de comunicación Can TD y Can RD, por lo tanto, se debe adicionar un módulo MCP2551 mismo que se muestra en la figura 2.2.

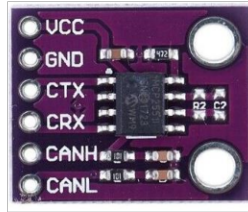


Figura 2.2. Circuito integrado para habilitar el uso de voltaje diferencial para comunicación CAN [14]

ESP32 Waveshare: es una tarjeta de desarrollo que utiliza las plataformas de Espressif IDF, Arduino IDE y PlatformIO. Cuenta con un procesador de doble núcleo lo que permite tener mayores velocidades de comunicación y transmisión de datos lo que permite su uso para el dispositivo en desarrollo. Se puede utilizar una tarjeta de la familia ESPE32, pero al no contar con pines dedicados para la comunicación CAN se puede optar por otros tipos de comunicación como es el caso de la comunicación UART y SPI.

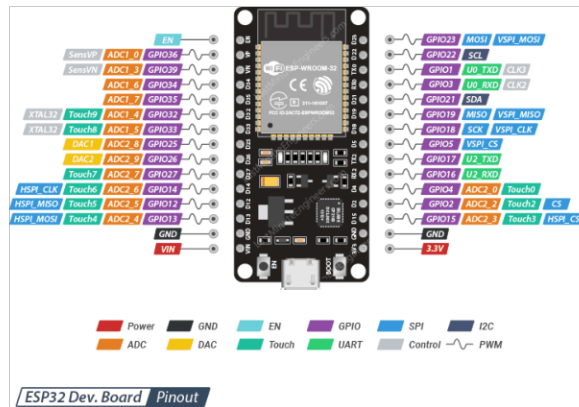


Figura 2.3. Especificaciones de pines de tarjeta de desarrollo ESP32 Waveshare [15]

Arduino NANO: Al igual que el caso anterior con la ESP32, la tarjeta de desarrollo de bajo costo de Arduino no cuenta con pines dedicados a la comunicación CAN, se opta por el uso de una comunicación SPI, además se requiere del uso de un módulo por separado para la conversión a comunicación CAN como se puede observar en la figura 2.5.

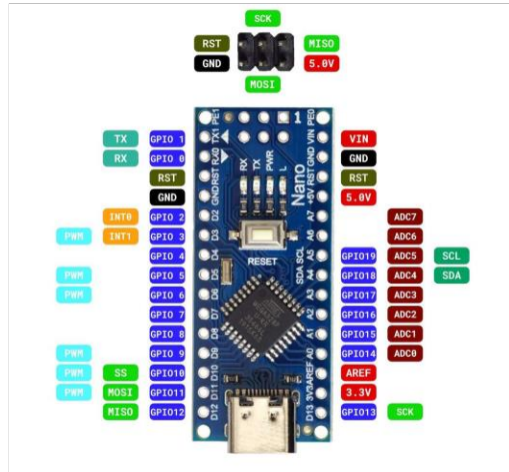


Figura 2.4. Descripción de pines de Arduino nano [16]

El módulo extra disponible es el MCP2515 TJA1050 que utiliza como entrada una interface de comunicación SPI y habilitando a la salida una interface de comunicación CAN, además que puede operar a distintos niveles de voltaje diferencial para la comunicación siendo un máximo de 5 V DC.



Figura 2.5. Módulo can bus MCP2515 TJA1050 compatible con Arduino [17]

2.2.1 SELECCIÓN DE DISPOSITIVOS Y ELEMENTOS

La selección de un microcontrolador que cuente con las especificaciones de poder operar con un protocolo de comunicación CAN, además de contar con las características necesarias para el envío y recepción de datos mediante los puertos CAN-H y CAN-L y que al manejar voltajes diferenciales requiere de componentes adicionales dado en el caso de ser tarjetas de desarrollo económicas.

Las tarjetas de desarrollo presentadas en la sección 2.2 cada una tiene sus ventajas y desventajas para cada una de las aplicaciones, para el caso de la comunicación CAN las

tres tarjetas pueden ser utilidad, pero se toma en cuenta, la disponibilidad de los módulos para la comunicación CAN. El módulo de la figura 2.2 que usa la STM32 no se encuentra disponible o solo una variante que no asegura una comunicación óptima al tener que retirar las resistencias terminales integradas en el módulo. Solamente se integran si el dispositivo se encuentra al inicio o final de la lineal principal de la red DeviceNet como se puede visualizar en la Figura 1.4.

Se opta por un Arduino Nano, debido a que asegura una comunicación estable al utilizar un módulo MCP2515 TJA1050 donde se puede integrar o retirar de manera manual la resistencia terminal mediante un jumper. Dado en el caso que el dispositivo se encuentre al final de toda la red en la rama principal o también denominado tronco como se especifica en la Figura 1.4.

El número de pines digitales para la modificación del número de nodo en el que se encuentra el dispositivo es de gran importancia siendo un máximo de 7 pines, 6 de ellos destinados a proporcionar e identificar el número de nodo y el restante para establecer el dispositivo en modo de lectura de pines o en modo de operación.

La tarjeta debe contar con al menos un pin para la lectura analógica, se descarta la STM32 y ESP32 al contar con una lectura de voltaje de un rango de 0-3.3V DC, el Arduino Nano cuenta con un rango de lectura de 0-5V DC lo que facilita el diseño de un convertidor reductor de voltaje.

Como adicionales para verificar el estado del dispositivo en la red se tienen los siguientes elementos:



Figura 2.6. LCD 16x2 con módulo I2C [18]

A fin de utilizar el menor número de pines posibles de la tarjeta se cuenta con un LCD con comunicación I2C con cuatro pines correspondientes, dos para comunicación y los dos restantes para alimentación 5V DC y GND.



Figura 2.7. Regulador de voltaje positivo L7812GV [19]

La red al contar con alimentación integrada de 24V DC y la tarjeta de desarrollo que soporta únicamente 12V DC, el uso de un regulador de voltaje como el que se observa en la Figura 2.7 cuenta con entrada máxima de 35 V DC y salida de 12 VDC, es necesario con el fin de evitar el uso de una fuente externa a la red.



Figura 2.8. USB-TTL para comunicación UART [20]

El módulo USB-TTL se utiliza de manera especial para mantener una comunicación UART mediante los pines TX y RX del Arduino nano como se puede visualizar en la Figura 2.8, para visualizar el estadio del dispositivo dentro de la red y de los datos transmitidos mediante el uso de una computadora.

Los capacitores se utilizan para mantener el valor de voltaje estable, reducir el ruido tanto a la entrada del pin de lectura analógica como de la fuente en la tarjeta.

CompactLogix: es un PLC del fabricante Allen Bradley, cuenta con módulos de entradas analógicas y digitales integradas, para operar con la red DeviceNet debe estar en conjunto a un Scanner compatible con el PLC y que a su vez es el que actúa como intermediario en el intercambio de información con los dispositivos esclavos de la red.



Figura 2.9. PLC CompactLogix [21]

2.2.2 SELECCIÓN DE SOFTWARE

Son varios los softwares a utilizar, tanto para realizar la programación del microcontrolador como para la configuración de la red DeviceNet, entre ellos se tiene los siguientes:

Arduino IDE: es un software de uso libre, que se encuentra en constante actualización, posee un gran soporte en cuanto a librerías para los distintos dispositivos compatibles con la plataforma de programación de microcontroladores, además de ser una aplicación intuitiva y con varios ejemplos de programas en C para diferentes tipos de utilidades y operaciones.

RSNetworx: Es un software netamente diseñado para la supervisión, monitoreo y diagnóstico para redes de carácter abierto ODVA y el poder detectar algún posible error de manera automática, esta herramienta integra las características necesarias para realizar una configuración y operación para un diseño de una red DeviceNet [18].

RSLogix5000: Es un software para la programación de PLCs compatibles, entre ellos se encuentra el CompactLogix, siendo un software específico preparado para la operación con la red DeviceNet [1].

2.3. DISEÑO DE TARJETA DE PROTOTIPO DE DISPOSITIVO

Se parte por el diseño de la fuente de voltaje que alimenta toda la tarjeta a partir de la fuente de 24 V DC disponible por la red DeviceNet, se considera que la placa debe contar con una fuente lo suficiente para alimentar a la tarjeta de desarrollo Arduino Nano, módulo MCP2515, LCD 16*2 con módulo I2C.

Además de contar con borneras para la conexión con la fuente de 24 V DC, para la comunicación CAN-H y CAN-L y de un circuito de acondicionamiento para la entrada de

voltaje de 0 a 10 V DC reducidos a un nivel de voltaje que soporta el puerto analógico del microcontrolador siendo este de 0 a 5 V DC todo sin afectar la integridad de la tarjeta.

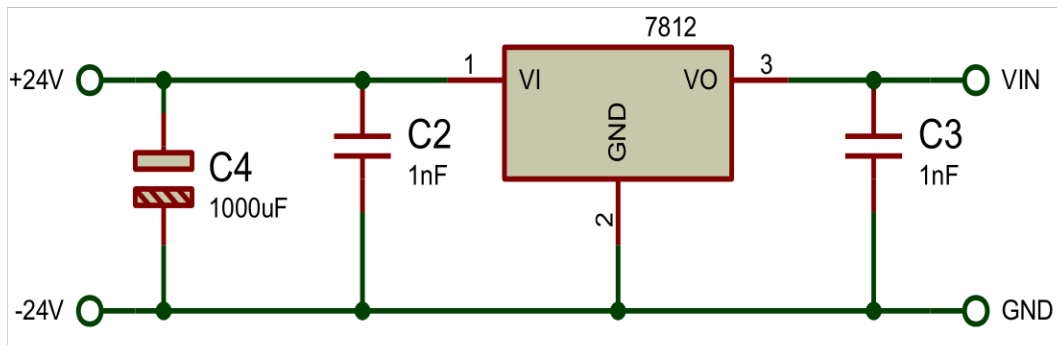


Figura 2.10. Diseño de fuente para alimentación del microcontrolador

La salida de la fuente diseñada es de un máximo de 12V DC necesario para alimentar la tarjeta de Arduino Nano siendo el voltaje que puede soportar.

Los datos de entrada hacia la tarjeta de desarrollo son de un nivel de voltaje parametrizado tal como se muestra en la siguiente Tabla 2.1

Tabla 2.1. Valores de voltaje de entrada del sensor y voltaje acondicionado

Voltaje de entrada VIN [V]	Voltaje de entrada a la tarjeta [V]
0	0
10	5

Se obtiene la ecuación que representa la conversión del voltaje de entrada al voltaje que puede soportar el microcontrolador a partir de gráficas de la Figura 2.11.

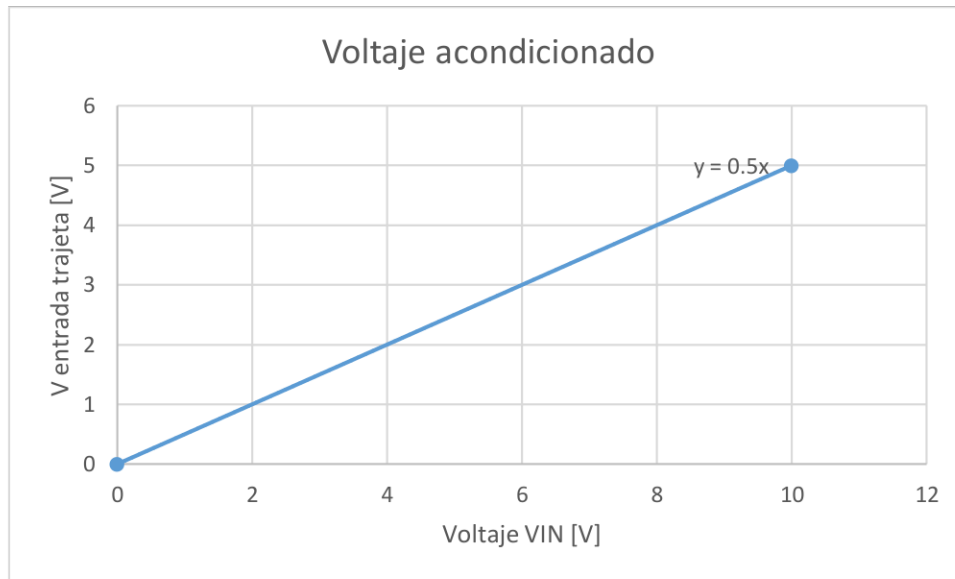


Figura 2.11. Gráfica de voltaje de entrada y acondicionado

Con la finalidad de evitar el uso de una fuente externa de voltaje al utilizar un amplificador operacional de ganancia 0.5 se opta por un divisor de voltaje.

$$V_{out} = \left(\frac{R1}{R1 + R2} \right) * V_{in}$$

Ecuación 2.1. Divisor de voltaje

Donde $\frac{R1}{R1+R2}$ son representados por un potenciómetro de precisión de 100K Ohm, el valor del voltaje de salida se calibra de manera manual hasta obtener una ganancia de 0.5.

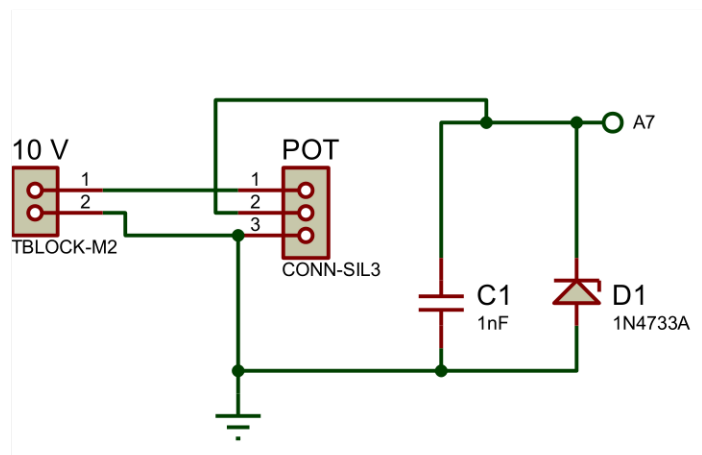


Figura 2.12. Circuito de acondicionamiento de voltaje para entrada de 0-10 a 0-5 V

Se asegura que el voltaje a la entrada del pin analógico A7 del Arduino Nano no supere los 5 V, mediante el uso de un diodo zener que soporta máximo 5.1V.

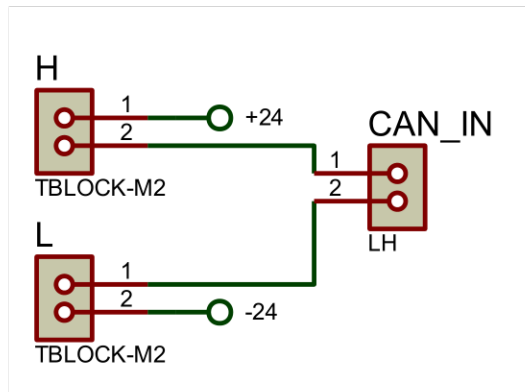


Figura 2.13. Sección para ingreso de voltaje de 24 V y de pines de comunicación CAN

Al contar con el módulo MCP2515 los pines que corresponden a la comunicación SPI deben ser conectado correctamente a la tarjeta de desarrollo tomando en cuenta que el primer pin no es usado para la comunicación.

De igual manera se cuenta con los pines de para la comunicación I2C para mostrar la información mediante el uso de una pantalla LCD sobre el estado del dispositivo en la red, para la comunicación serial UART y los pines para los interruptores que permiten el cambio del número del nodo físico del dispositivo.

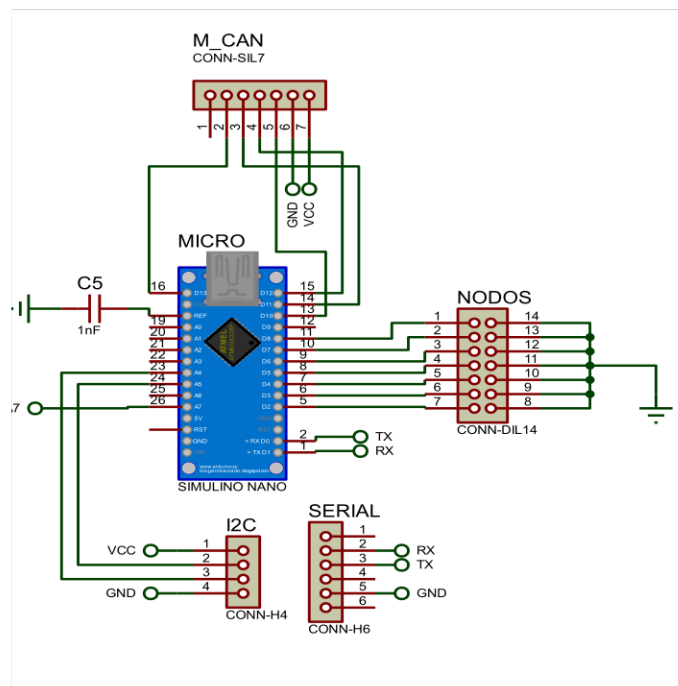


Figura 2.14. Diagrama de conexión de entradas y salidas para el Arduino Nano

Una vez establecida todas las conexiones necesarias hacia el microcontrolador se realiza el diseño de la PCB mostrada en la Figura 2.15

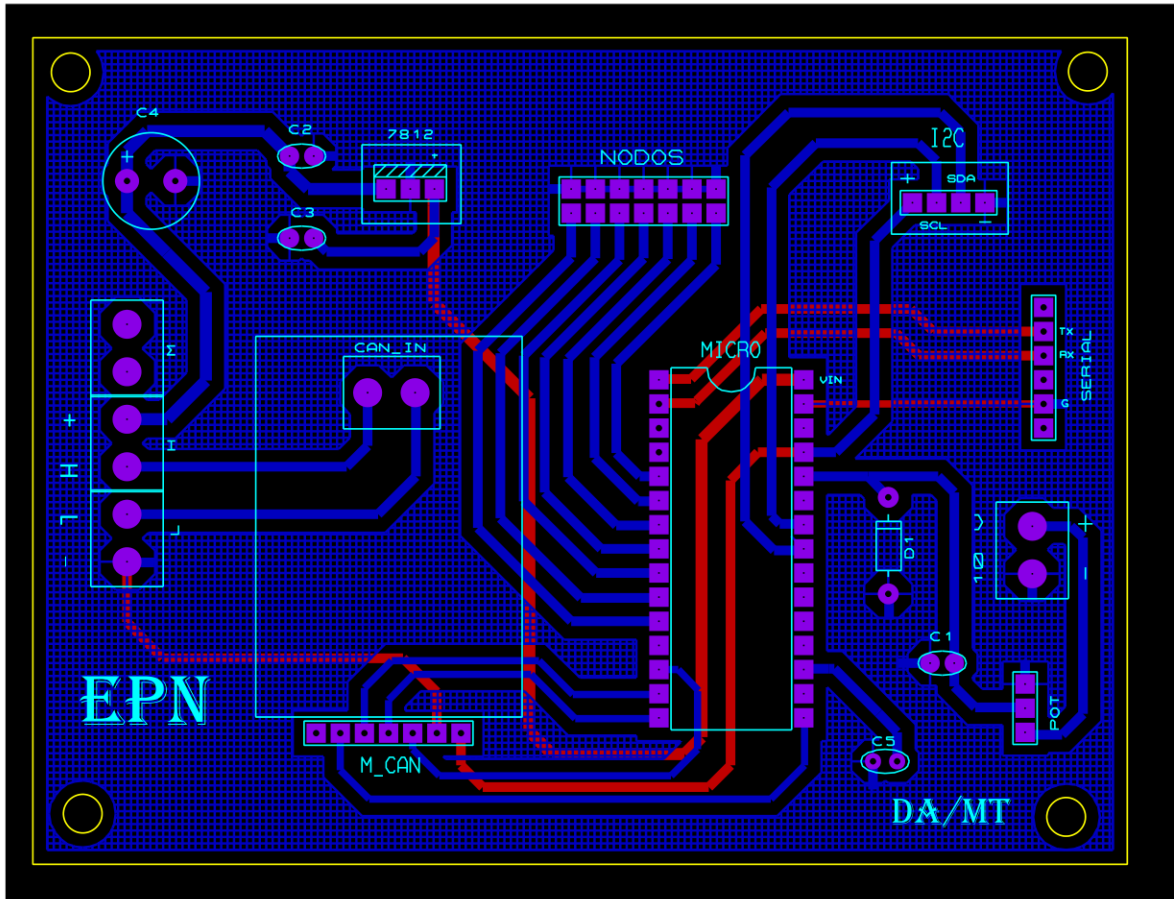


Figura 2.15. PCB de prototipo de dispositivo analógico

Las consideraciones para un buen diseño de PCB dependen del tipo de operación que realizara la placa, para el caso del prototipo no utiliza valores de voltaje considerablemente altos, únicamente el caso de la fuente regulada a 12V requiere de un mayor grosor de las pistas, de igual manera un caso crítico son los puertos de comunicación tanto para CAN como para SPI e I2C, las pistas no son muy largas.

Contar con una malla para reducir los ruidos externos es de gran importancia, todos los pines que no son utilizados son directamente conectados a la malla.

Se requiere de un diseño a doble lado de las pistas con la finalidad de evitar contar con un gran número de puentes siendo en muchos casos perjudiciales por el ingreso de señales de interferencia.

2.4. IMPLEMENTACIÓN DE DISPOSITIVO EN LA RED

Para integrar el prototipo de dispositivo analógico en la red DeviceNet se requiere una secuencia para que la red pueda reconocer los datos que son enviados desde el dispositivo hacia el PLC y puedan realizar el proceso de análisis y uso en un proceso industrial.

2.4.1 CONFIGURACIÓN DE PROTOTIPO DE DISPOSITIVO

El dispositivo debe cumplir con dos procesos, el primero es cuando la red se enciende por primera vez y el dispositivo debe ser capaz de integrarse a la red, mientras que el segundo es una vez que el dispositivo se encuentra enviando los datos de la señal analógica.

2.4.1.1 LÓGICA DE PROGRAMACIÓN

Con respecto a la configuración y programación del microcontrolador se tiene dividido en dos secciones con la finalidad de agilizar al dispositivo, los cuales son los siguientes:

En la primera sección es necesario realizar la lectura de los pines que determinan el número del nodo en el que se encuentra el dispositivo, esto mediante los dipswitch físicos. Además de contar con un pin adicional para establecer el dispositivo en modo de lectura o de operación.

Al encontrarse el dispositivo en modo de lectura, se realiza como su nombre lo indica mediante los interruptores físicos es posible realizar el cambio del número de nodo y de manera automática se crean las matrices que corresponden a los identificadores tanto del maestro como del esclavo para poder realizar el proceso de establecimiento de conexión que está constituido de 15 pasos que consiste en la recepción identificador ID y datos en formato de mensajes de 8 bits del maestro y la respuesta por parte del dispositivo esclavo.

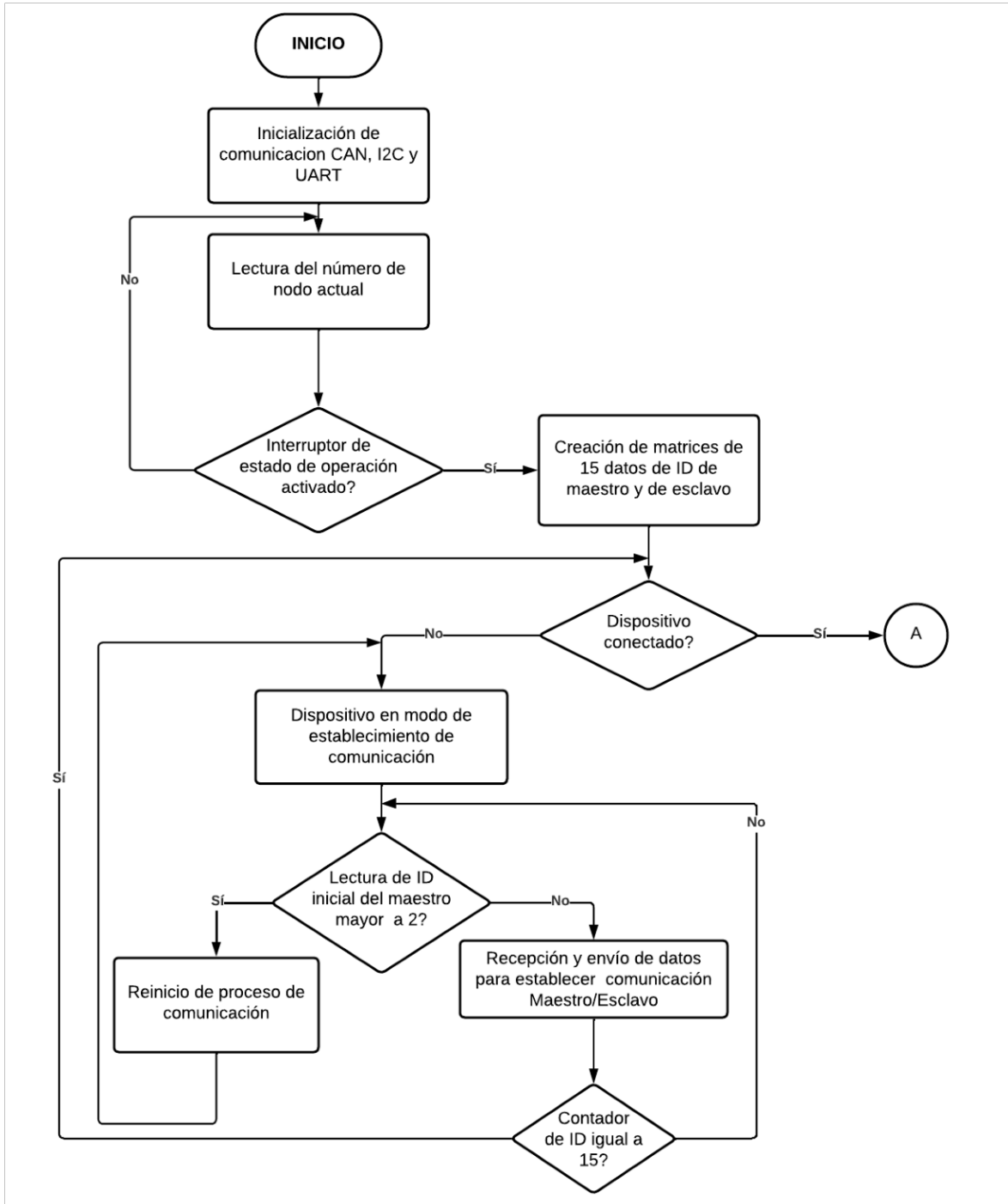


Figura 2.16. Lectura y Establecimiento de comunicación del Dispositivo al encender la red.

En ocasiones cuando se realiza el proceso de establecimiento de comunicación, puede suceder que el proceso de los 15 pasos no se complete de manera correcta, el scanner puede volver a enviar el ID inicial del proceso para volver a intentar realizar la comunicación.

Si se da el caso que el dispositivo debe volver nuevamente a la cuenta al estado inicial con la finalidad de reiniciar el proceso de establecimiento de comunicación este proceso se puede repetir de manera indefinida hasta lograr la comunicación entre el maestro y el esclavo en la red, las series de 15 pasos son tomados en base al sensor fotoeléctrico disponible en el laboratorio de redes industriales.

Para la segunda sección se considera que el dispositivo se encuentra conectado en la red y reconocido por el scanner, también que los datos son transmitidos y recibidos por el PLC.

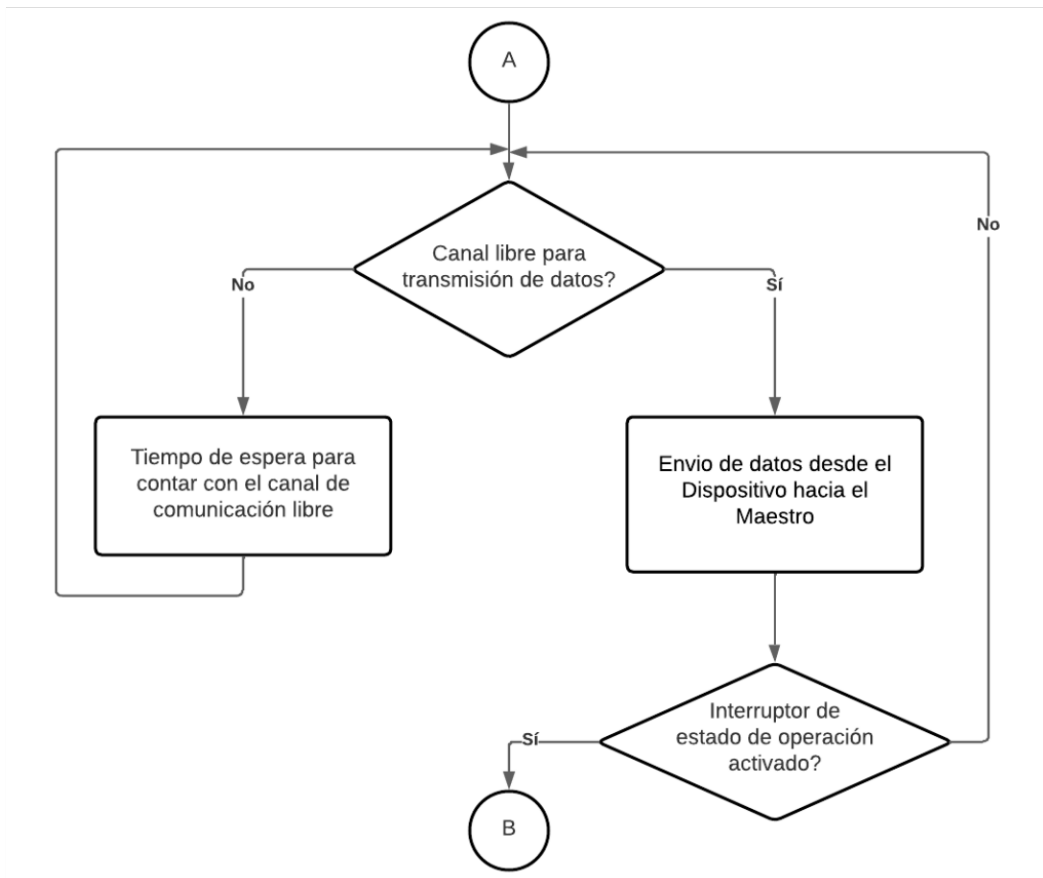


Figura 2.17. Envío de datos desde el esclavo hacia el maestro

Para evitar una colisión de datos por algún otro dispositivo que se encuentre transmitiendo en la red se cuenta con un tiempo de espera, al igual que la propia configuración del protocolo de comunicación.

Una vez se finaliza la transmisión de datos, se verifica que el pin de estado no se encuentre activado, si se da el caso entonces el dispositivo se desconecta de la red para iniciar nuevamente el proceso de restablecimiento de comunicación.

2.4.1.2 ESTABLECIMIENTO DE COMUNICACIÓN

La red DeviceNet disponible en el laboratorio de redes industriales, opera mediante una trama de datos de 11 bits para el identificador ID y que corresponde al modo normal de operación.

Al configurar la red DeviceNet se establece el número de nodo del dispositivo dentro del Frame ID en la sección del MAC ID, la sección de Message ID contienen el valor de la cantidad de datos a enviar.

Para establecer comunicación la dirección del Frame ID debe pertenecer al grupo de tipo 2 cuyo rango va desde [400-5FF] en hexadecimal, esto al tratarse de un sensor con características de operación simples como es el caso del sensor fotoeléctrico disponible en el laboratorio de redes industriales.

Para el caso de estudio el dispositivo (prototipo de sensor) se encuentra en el nodo 10 como ejemplo, dicho valor debe tener la posibilidad de ser modificado de manera física mediante interruptores con la finalidad de ubicar el sensor en cualquiera de los nodos disponibles del scanner de la red DeviceNet siendo el valor máximo de 63 en decimal.

En la siguiente Tabla 2.2 se presenta la distribución de Frame ID para el proceso de conexión del dispositivo en la red:

Tabla 2.2. Proceso de establecimiento de conexión del dispositivo esclavo

	Estado	Frame ID Hexadecimal	Frame ID Binario									Descripción		
			G2		MAC ID						MESSAGE ID			
Maestro	1	0X0456	1	0	0	0	1	0	1	0	1	1	0	ID Maestro para iniciar la comunicación con el esclavo
			2	10						6				
Esclavo		0X0457	1	0	0	0	1	0	1	0	1	1	1	ID de respuesta del Esclavo
			2	10						7				
Maestro	2	0X0456	1	0	0	0	1	0	1	0	1	1	0	ID Solicitud para información del dispositivo esclavo (Información introducida en el EDS)
			2	10						6				
Esclavo		0X0453	1	0	0	0	1	0	1	0	0	1	1	
			2	10						3				
Maestro	3	0X0454	1	0	0	0	1	0	1	0	1	0	0	
			2	10						4				
Esclavo		0X0453	1	0	0	0	1	0	1	0	0	1	1	
			2	10						3				

Maestro	4	OX0454	1	0	0	0	1	0	1	0	1	0	0	
			2	10							4			
Esclavo	4	OX0453	1	0	0	0	1	0	1	0	0	1	1	
			2	10							3			
Maestro	5	OX0454	1	0	0	0	1	0	1	0	1	0	0	
			2	10							4			
Esclavo	5	OX0453	1	0	0	0	1	0	1	0	0	1	1	
			2	10							3			
Maestro	6	OX0454	1	0	0	0	1	0	1	0	1	0	0	
			2	10							4			
Esclavo	6	OX0453	1	0	0	0	1	0	1	0	0	1	1	
			2	10							3			
Maestro	7	OX0454	1	0	0	0	1	0	1	0	1	0	0	
			2	10							4			
Esclavo	7	OX0453	1	0	0	0	1	0	1	0	1	1	1	
			2	10							7			
Maestro	8	OX0454	1	0	0	0	1	0	1	0	1	0	0	
			2	10							4			
Esclavo	8	OX0453	1	0	0	0	1	0	1	0	1	1	1	
			2	10							7			
Maestro	9	OX0454	1	0	0	0	1	0	1	0	1	0	0	
			2	10							4			
Esclavo	9	OX0453	1	0	0	0	1	0	1	0	0	1	1	
			2	10							3			
Maestro	10	OX0454	1	0	0	0	1	0	1	0	1	0	0	
			2	10							4			
Esclavo	10	OX0453	1	0	0	0	1	0	1	0	0	1	1	
			2	10							3			
Maestro	11	OX0454	1	0	0	0	1	0	1	0	1	0	0	Tipo de dato de entrada tipo encuesta
			2	10							4			
Esclavo	11	OX0453	1	0	0	0	1	0	1	0	0	1	1	
			2	10							3			
Maestro	12	OX0454	1	0	0	0	1	0	1	0	1	0	0	Tipo de dato de entrada tipo cambio de estado
			2	10							4			
Esclavo	12	OX0453	1	0	0	0	1	0	1	0	0	1	1	
			2	10							3			
Maestro	13	OX0454	1	0	0	0	1	0	1	0	1	0	0	Tipo de dato de salida tipo encuesta
			2	10							4			
Esclavo	13	OX0453	1	0	0	0	1	0	1	0	0	1	1	
			2	10							3			
Maestro	14	OX0454	1	0	0	0	1	0	1	0	1	0	0	Tipo de dato de salida tipo cambio de estado
			2	10							4			
Esclavo	14	OX0453	1	0	0	0	1	0	1	0	0	1	1	
			2	10							3			

Maestro	15	0X0454	1	0	0	0	1	0	1	0	1	0	0	Solicitud del maestro para información de tasa de paquetes estimada
			2	10						4				
0X0453		1	0	0	0	1	0	1	0	0	1	1		
		2	10						3					

Una vez que la conexión entre el dispositivo y el scanner sea exitosa, acorde a la configuración el dispositivo analógico enviará datos al maestro con un tiempo de espera predeterminado, se tiene en cuenta las indicaciones de la siguiente Tabla 2.3 del Frame ID para envío de datos en configuración maestro/esclavo.

Tabla 2.3. Configuración de Frame ID para envío de datos desde el esclavo hacia el maestro

	Estado	Frame ID Hexadecimal	Frame ID Binario										Descripción	
			G	MESSAGE ID					MAC ID					
Esclavo	1	0X034A	0	1	1	0	1	0	0	1	0	1	0	ID del esclavo conectado a la red
			1	13					10					
			Grupo	I/O Tipo de mensaje a enviar	Dirección de nodo del esclavo									

Se define la prioridad de envío de datos al pertenecer al grupo uno, donde el dispositivo con menor Message ID tiene la mayor prioridad, en el caso de contar con dos Message ID idénticos, se compara el MAC ID y le dispositivo ubicado en el nodo más bajo realiza el envío de datos hacia el maestro.

2.4.2 CONFIGURACIÓN DE RED EN RSNETWORX

RSNetWorx permite realizar la configuración de la red de manera virtual, una vez que todos los dispositivos se encuentran conectados de manera física en la red, cada dispositivo permite verificar el número de nodo al que pertenece, por esta razón antes de poner en marcha la red el prototipo de sensor analógico debe contar con un nodo diferente a los otros dispositivos.

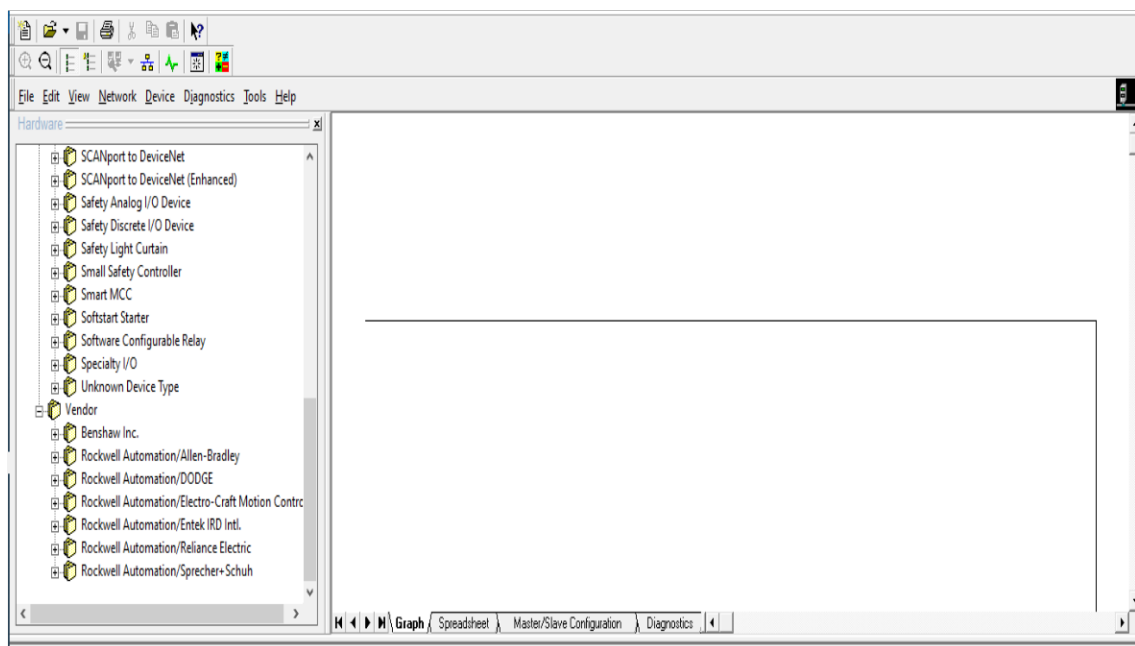


Figura 2.18. Interfaz para el diseño de la red DeviceNet

Una vez que se inicia el programa por defecto en muchos casos aparecen algunos de los dispositivos conectados en la red, al pertenecer otra red configurada y cargada en el scanner, para contar con la pantalla en blanco se elige la opción de nuevo proyecto.

En las carpetas ubicadas en la parte izquierda se tiene todos los dispositivos disponibles y que son compatibles por defecto con la red DeviceNet siendo únicamente los que pertenecen al fabricante Rockwell Automation. En las carpetas se elige el scanner 1769-ECR de la serie A con el que se cuenta en el laboratorio y por defecto la ubicación del nodo es la 00 siendo esta la dirección del dispositivo maestro.

Se realiza el mismo procedimiento para los demás dispositivos que se encuentran conectados a la red, para añadir un nuevo dispositivo es necesario la instalación del EDS del mismo en la aplicación “EDS Hardware Installation Tool”.

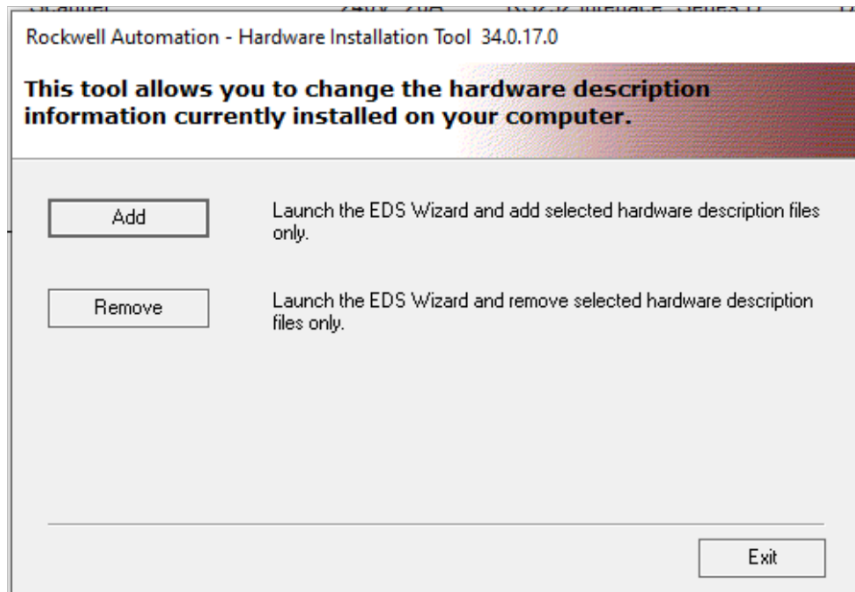


Figura 2.19. Ventana de la aplicación para la instalación de nuevos archivos EDS

La aplicación brinda la posibilidad de remover o instalar un nuevo EDS, este archivo es necesario para la configuración y permitir que el dispositivo pueda comunicarse en la red con el maestro.

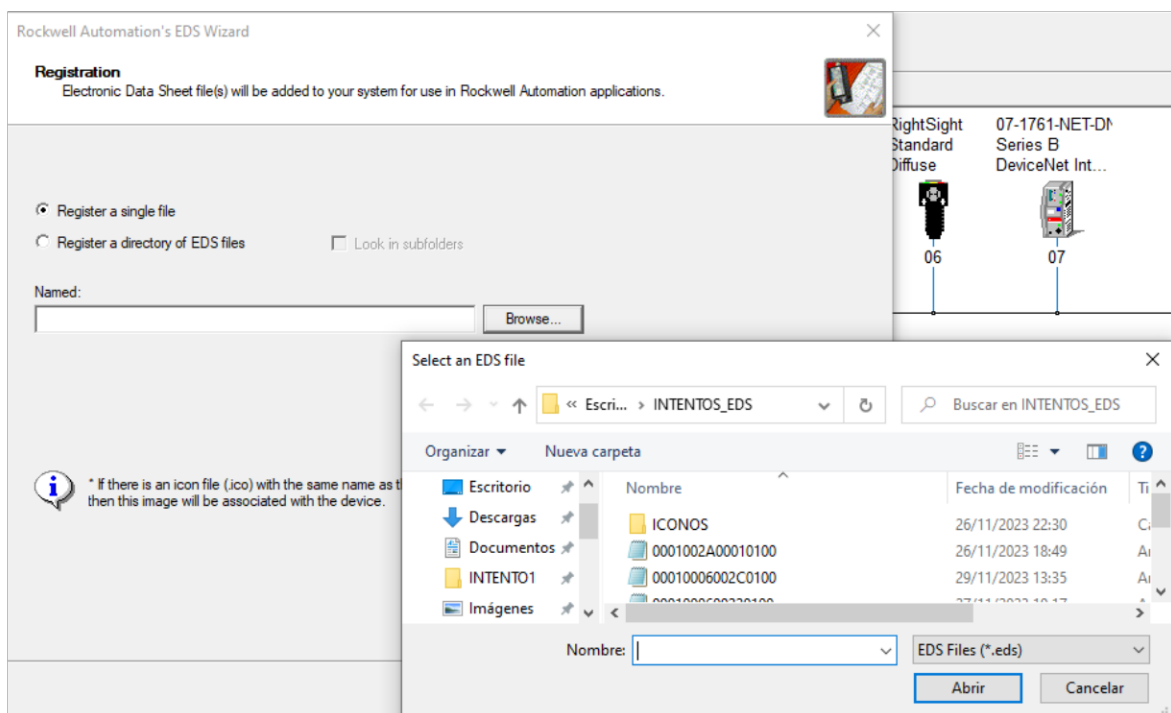


Figura 2.20. Selección de archivo EDS modificado para prototipo de sensor analógico.

Una vez instalado el archivo, el nuevo dispositivo está disponible en las carpetas de la red DeviceNet con su correspondiente ícono y nombre para ser identificado.

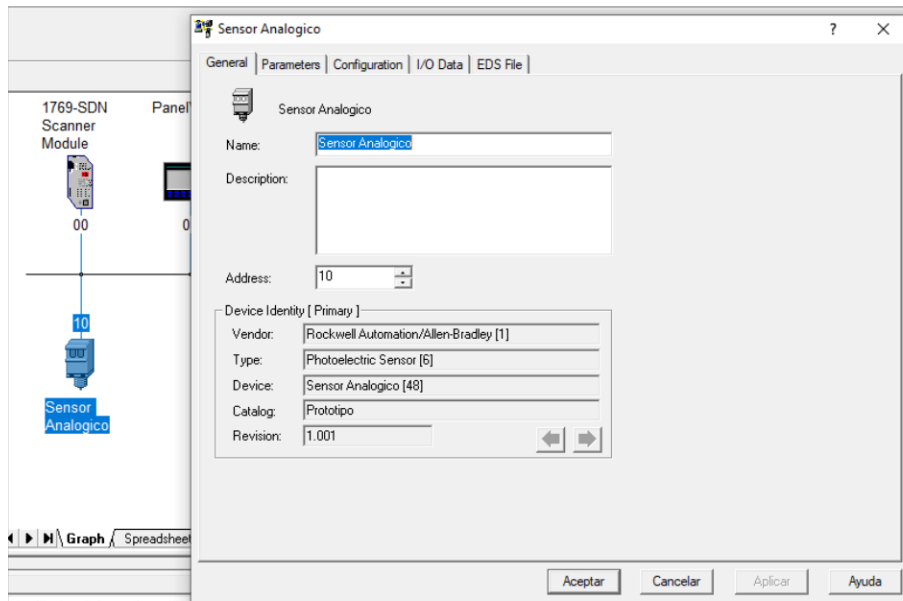


Figura 2.21. Nuevo dispositivo integrado a la red

Al contar con la posibilidad de realizar el diseño de la red de manera virtual una vez integrado todos los dispositivos, se verifica que el número de nodo de cada uno sea si equivalente a la real.

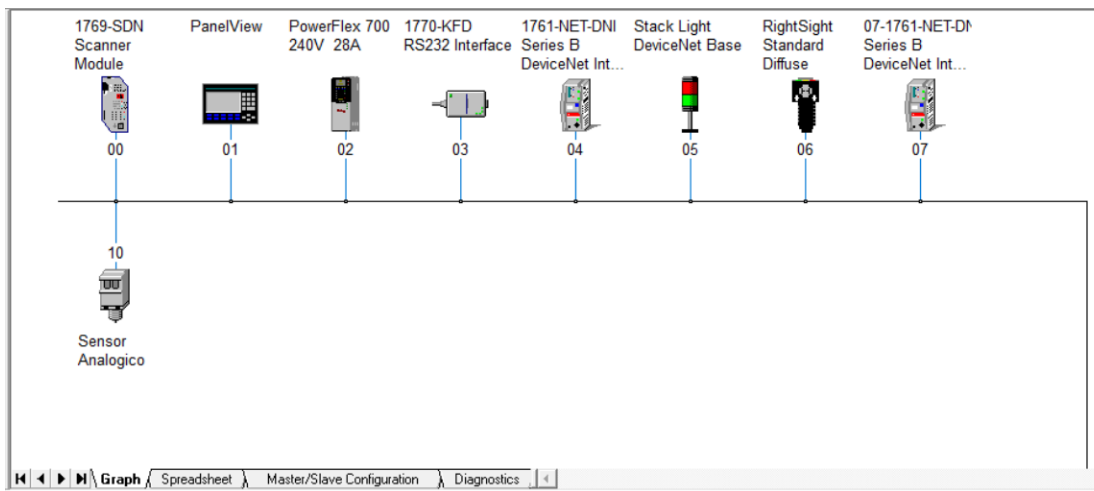


Figura 2.22. Rede DeviceNet diseñada con prototipo de sensor analógico integrado.

Para realizar la lectura de datos que son enviados desde el dispositivo esclavo hacia el maestro se ingresa en el scanner a la opción Scanlist y se selecciona los dispositivos con los que se va a realizar la comunicación.

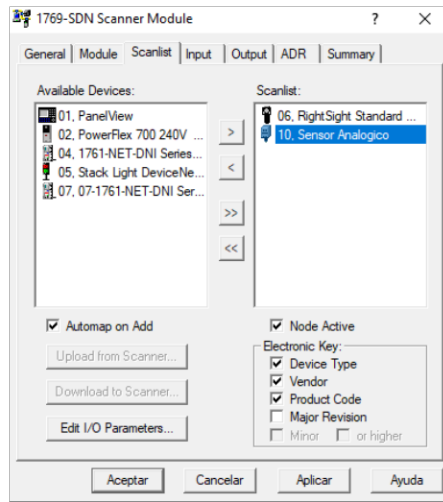


Figura 2.23. Selección de dispositivos esclavos para la comunicación

En la opción Input son los datos enviados desde los dispositivos esclavos hacia el maestro, el escáner permite ordenar todos los datos de entrada en espacios de memoria de 32 bits, en ocasiones la ubicación de los datos de los dispositivos es de manera automática debido a que el scanner está configurado para optimizar los espacios de memoria y no contar con bits sin utilizar.

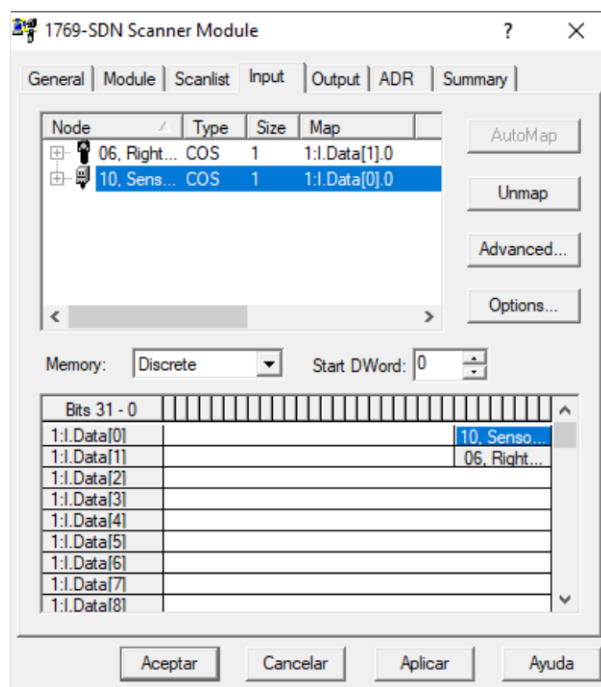


Figura 2.24. Ubicación de datos en los espacios de memoria del scanner

Para el caso de la configuración manual se da clic en la opción Advanced Mapping donde se configura el tipo de mensaje a recibir, el número de bits donde inicia y el número de bits a utilizar y en qué espacio de memoria se va a ubicar los datos.

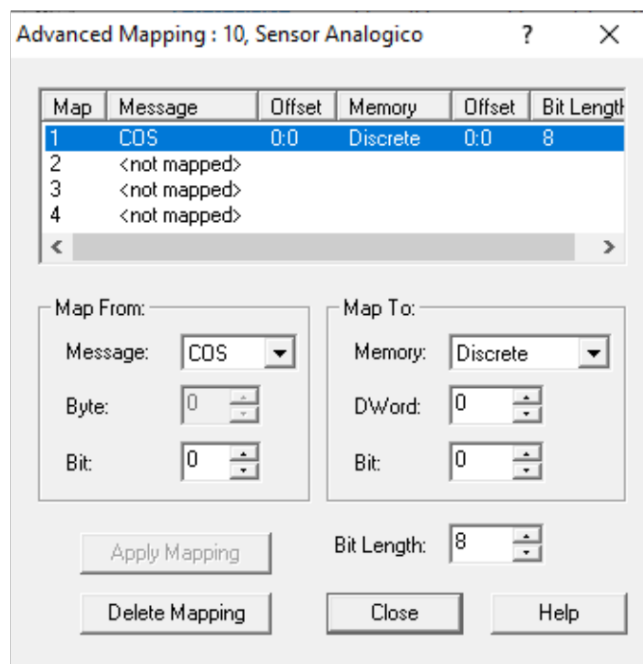


Figura 2.25. Configuración manual para el mapeo de datos recibidos

El prototipo de sensor analógico únicamente envía datos hacia el maestro, por ende, en la sección de salidas Output no es necesario la configuración, pero de igual manera que en el caso de las entradas se cuenta con la configuración automática del espacio de memoria para la salida de datos desde el maestro.

Una vez realizado todas las configuraciones se procede a subir la nueva red al escáner, eso cuando el sistema se encuentra en marcha u online para que el sensor pueda reconocer todos los cambios efectuados.

2.4.2.1 CONFIGURACION DEL ARCHIVO EDS

- **Encabezado:**

En esta sección de la hoja de datos se proporciona la información sobre el dispositivo, tales como: tipo y nombre del modelo, número de serie, datos del fabricante, fecha de elaboración del EDS, catálogo.

```

[File]
  DescText = "Sensor Analògico EDS File";

  CreateDate   = 07-30-2005;
  CreateTime   = 12:00:00;
  ModDate      = 10-18-2005;
  ModTime      = 11:40:00;
  Revision     = 2.01;

[Device]
  VendCode     = 1;
  VendName     = "Rockwell Automation/Allen-Bradley";    $ Nombre de la carpeta que contiene el dispositivo por defecto
  ProdType     = 6;
  ProdTypeStr  = "Photoelectric Sensor";                $ Nombre de la subcarpeta que contiene al dispositivo (sensor)
  ProdCode     = 44;                                    $ Còdigo para diferenciar el dispositivo
  MajRev       = 1;                                    $ Device Major Revision
  MinRev       = 1;                                    $ Device Menor Revision
  ProdName     = "Sensor Analògico";                    $ Nombre del dispositivo
  Catalog      = "Prototipo";                           $ Determina el catàlogo donde se debe buscar el sensor
  Icon         = "1734safe.ico";                        $ Determinar el icono del sensor

```

Figura 2.26. Sección de encabezado de archivo EDS

La estructura del encabezado es de carácter obligatorio al contener esencialmente el tipo de dispositivo que se va a conectar a la red, en el apartado Device se realiza la configuración del dispositivo a partir de un EDS ya existente en el software RSNetwork, es posible configurar el nombre del dispositivo, el catálogo y el ícono con que está representado en la red virtual.

Para el caso de un EDS para la red DeviceNet se tiene una sección donde se define la clase de red a la que se conectara el dispositivo.

```

[Device Classification]
Class1 = DeviceNet;

```

Figura 2.27. Definición de tipo de red

- **Datos de entrada y salida:**

En la configuración de entras en el bloque IO_Info se especifica el tipo de dato a enviar o recibir por parte del dispositivo, la especificación puede variar acorde al tipo de dispositivo integrado en la red. Se compara dos tipos de EDS de dos dispositivos.

```

[I0_Info]
  1_COSetPathNotSupported = 1;                $ no setting of connection path

  Default = 0x0004;                          $ COS perfered
  PollInfo = 0x0001, 1, 1;                  $ Not supported
  StrobeInfo = 0x0002, 1, 1;                $ Strobe is Input1 only, Interval counting not supported
  COSInfo = 0x0004, 1, 1;                   $ Use Input1

$tipo cos
  Input1 =
    1,                                       $ Se especifica que el dispositivo admite un parámetro de entrada de un Byte
    0,                                       $ Se especifica el número de bits a utilizar
    0x0000,                                  $ Especifica el tipo de datos que envía el dispositivo COS and Strobe
    "Sensor",                                $ Nombre del dispositivo
    6, "20 04 24 01 30 03",                 $ Tamaño de ruta al atributo de valor de ID

$ Para el caso del sensor digital se especifica las entradas utilizadas, además de la función que desempeña cada uno de los bits enviados hacia el scanner

  "The format for the COS and Strobe input byte follows: \n"
  "DB7 Not Used \n"
  "DB6 Not Used \n"
  "DB5 Counter Output \n"
  "DB4 Motion Output \n"
  "DB3 Margin Diagnostic 3x \n"
  "DB2 Margin Diagnostic 2x \n"
  "DB1 Sensor Output \n"
  "DB0 Sensor Output \n"
  "See specific parameter for more information.";

```

Figura 2.28. Configuración de entradas de un dispositivo digital

En la especificación de los datos de entrada o salida que puede soportar el dispositivo esclavo para el caso del prototipo de sensor se puede modificar el número de entradas y eliminar las salidas, se recalca que esto se lo realiza en criterio visto desde el scanner donde las entradas son los datos recibidos desde el dispositivo esclavo.

```

Input1 =
  9,                                       $ Size in bytes
  68,                                     $ Significant bits (0 = all)
  0x0000,                                  $ I/O Type Mask: Poll, COS, Cyclic
  "Safety + Channel Status",              $ Name Description
  8,                                       $ Connection Path Size
  "20 04 25 00 92 01 30 03",             $ Connection Path
  "Safety Inputs and Channel Status";     $ Help String

```

Figura 2.29. Configuración de entradas para un dispositivo analógico

El mayor cambio en comparación al dispositivo digital (ver figura 2.28) se encuentra en la configuración de las entradas donde no se especifica la función de cada bit, al contrario, los datos medidos el dispositivo de manera analógica (Ver figura 2.29) se envían en datos de 16 a 32 bytes dependiendo de la resolución del dispositivo.

- **Sección de parámetros:**

Las configuraciones de parámetros de ambos dispositivos difieren por el tipo de variable a manejar


```

[Params]
Param1 =
0, $ Light Source Detected
6, "20 0E 24 01 30 92", $ Data Placeholder
0x0012, $ 6 Bytes Read class 01 attribute x92
0xC1, 1, $ Enumerated Read-only
"Target", $ Boolean 1 Byte
"", $ No Units
"This indicates when the sensor's receiver is detecting light. \n"
"Detected' indicates that at least '1.0 x' the amount of light needed by \n"
"the sensor to activate is being received by the sensor. \n"
"Not Detected' indicates when the sensor detects an amount of light \n"
"equivalent to a '0.8 x' or less of the light needed for 'SENSOR OUTPUT'activation.",
0,1,0, $ min, max, default values
1,1,1,0, $ mult, div, base, offset scaling default to 1 xxxxx1xxx in discriptor
0,0,0,0, $ mult, div, base, offset links default to 0 xxxxx1xxx in discriptor
0; $ decimal places

```

Figura 2.30. Parámetros para un dispositivo Digital

```

[Params]
Param1 =
0, $ shall Equal 0
6,"20 49 24 01 30 02", $ path size
0x0030, $ path
0xC3, $ descriptor
2, $ data type=INT
"Ch0 Input Value", $ name
"", $ unit
"Value of Analog Input Channel 0", $ Help string
-32768,32767,0, $ min/ max/ default
,,,, $ mult/ div /base/ offset scaling
,,,, $ mult/ div /base/ offset links not used
0; $ decimal places

```

Figura 2.31. Parámetros para un dispositivo analógico

Para el caso del dispositivo analógico como se puede observar en la figura 2.31 es posible realizar un rango de medición más amplia debido a la cantidad de bits recibidos, en la sección marcada de la figura 2.30 que corresponde a las características de un dispositivo digital el rango de medición se encuentra limitada a tres estados lógicos.

Además, que es en esta sección que se asigna los bytes (nombre Ch0) que recibe el scanner por parte del sensor analógico.

- **Prueba de instalación de EDS:**

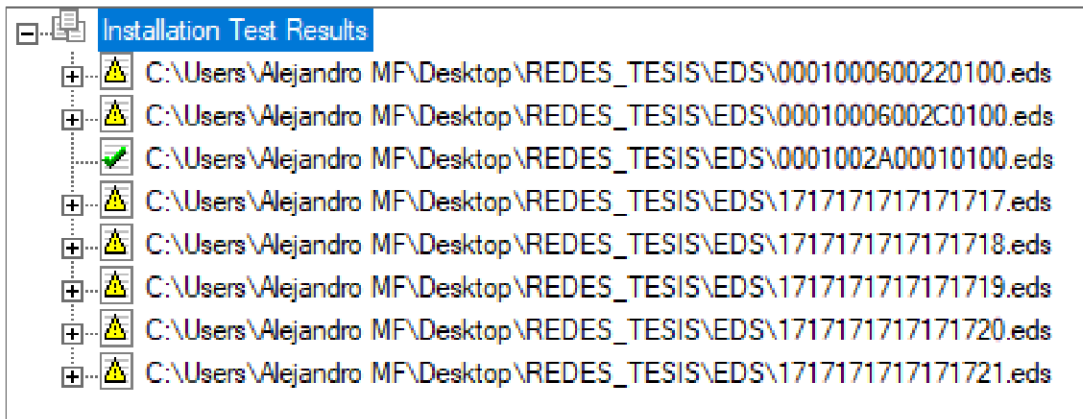


Figura 2.32. Instalación de archivos EDS

Al momento de realizar la instalación de los EDS modificados, estos son comparados con un EDS sin modificar, donde se verifica una serie de advertencias.

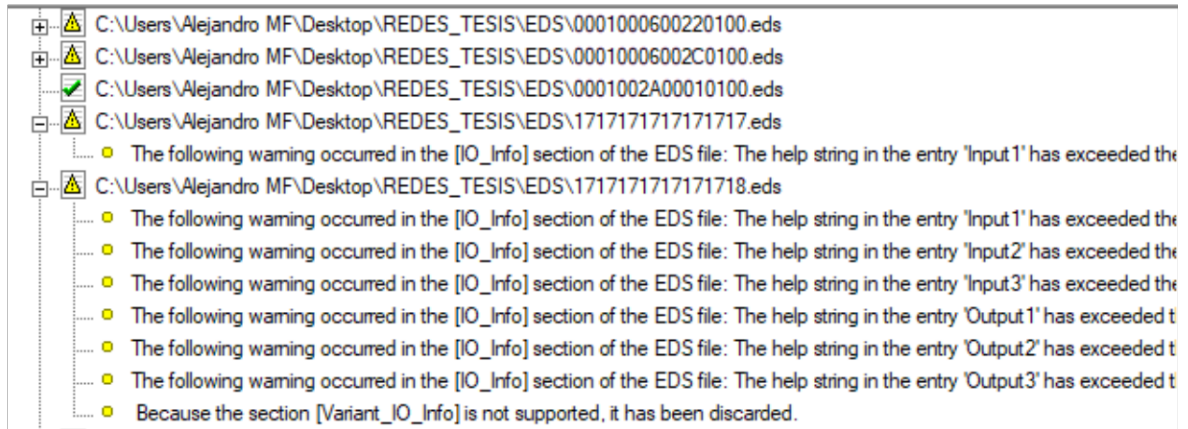


Figura 2.33. Lectura de advertencias presentes en EDS

Se compara un EDS del sensor fotoeléctrico con el prototipo donde ambos presentan advertencias, en el segundo caso los errores ocasionan que la sección de Varian_IO_Info sea descartado.

Un EDS que se encuentra con los correctos parámetros no debe presentar las advertencias al momento de cargar el archivo

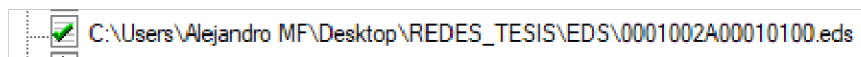


Figura 2.34. Archivo EDS instalado correctamente

2.4.3 CONFIGURACION DE PLC

Se realiza la configuración del PLC para la lectura de los datos enviados desde el prototipo de sensor analógico para ello se utiliza el programa RSLogix 5000 y se selecciona el PLC disponible en el laboratorio.

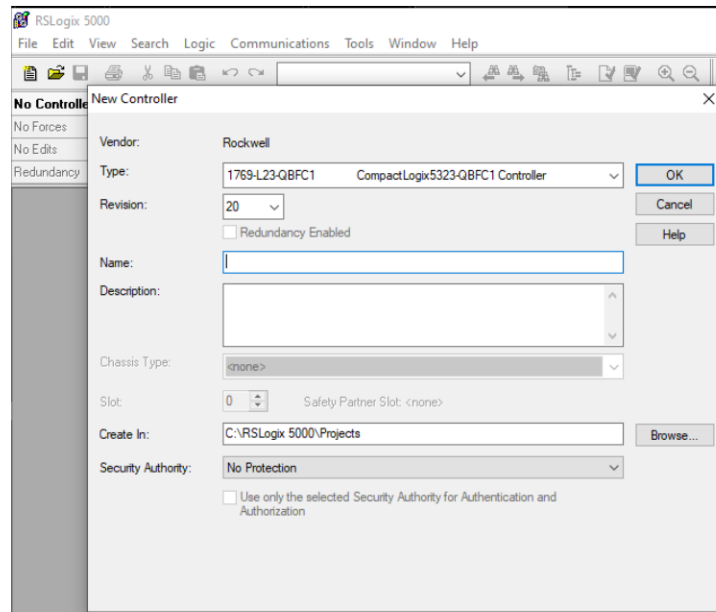


Figura 2.35. Selección de PLC a utilizar

Al guardar la selección del PLC se tiene en la ventana de “Controller Organizer” los módulos de entradas digitales y analógicas, los módulos de salidas digitales y analógicas integrados en el PLC.

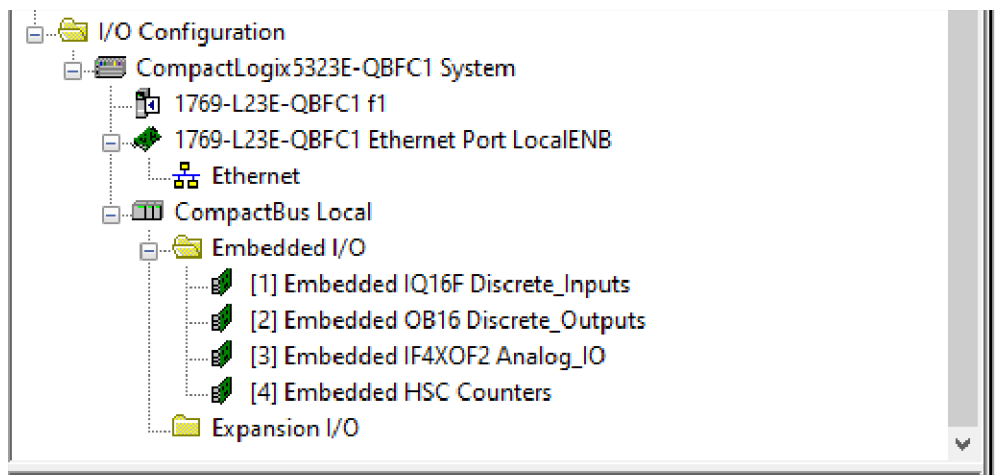


Figura 2.36. Módulos integrados de entradas y salidas digitales y analógicas

En la carpeta “Expansion I/O” es donde se debe añadir el módulo del scanner para tener la comunicación con la red DeviceNet y recibir los datos de los dispositivos configurados.

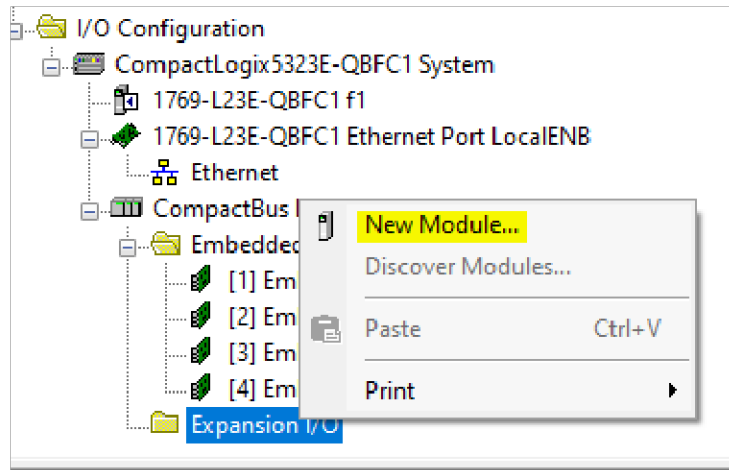


Figura 2.37. Expansión para Scanner para red DeviceNet

El modelo del scanner debe ser el mismo disponible en la red física, se pide el número de revisión de manera automática es 3.

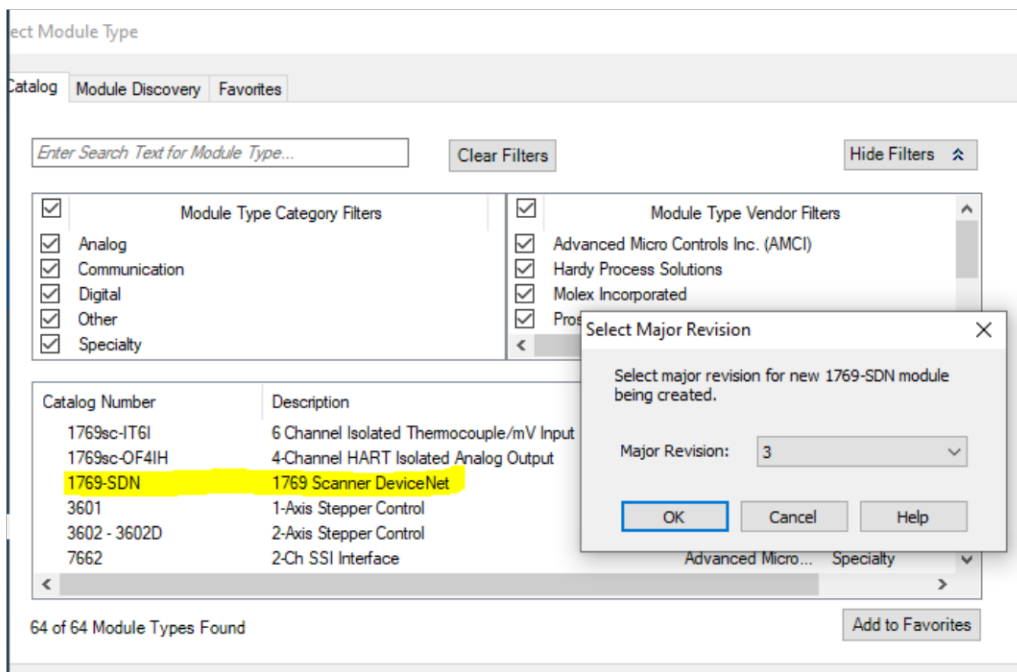


Figura 2.38. Selección de Scanner para Red DeviceNet

Al visualizar la figura 2.39 que pertenece a la ventana de configuración permite modificar el número de los espacios de memoria del scanner, número de bits de entrada y de salida.

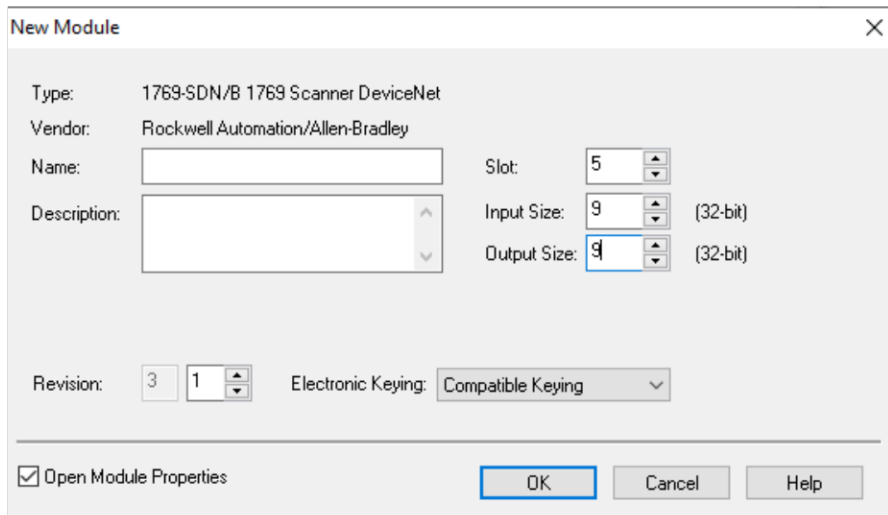


Figura 2.39. Configuración de espacios de memoria y bits de entrada y salida del scanner

Para tener una comunicación entre el PLC y la red DeviceNet el scanner sirve como intermediario ordenando los datos tanto para el envío o recepción, además se debe añadir la red modificada al módulo del scanner para asegurar la comunicación y transmisión de datos entre el PLC y los dispositivos que se encuentran en la red.

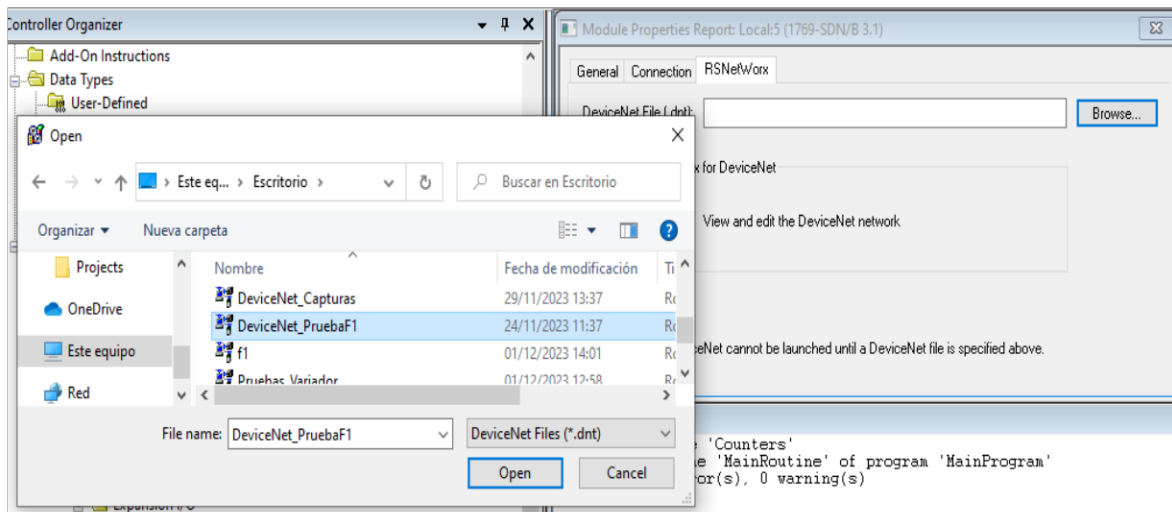


Figura 2.40. Selección de red DeviceNet diseñada

Los programas correspondientes a la lectura de cada uno de los datos que circulan por la red se lo hace mediante los registros del módulo de expansión tanto para la entrada y salida de los datos provenientes de los dispositivos esclavos.

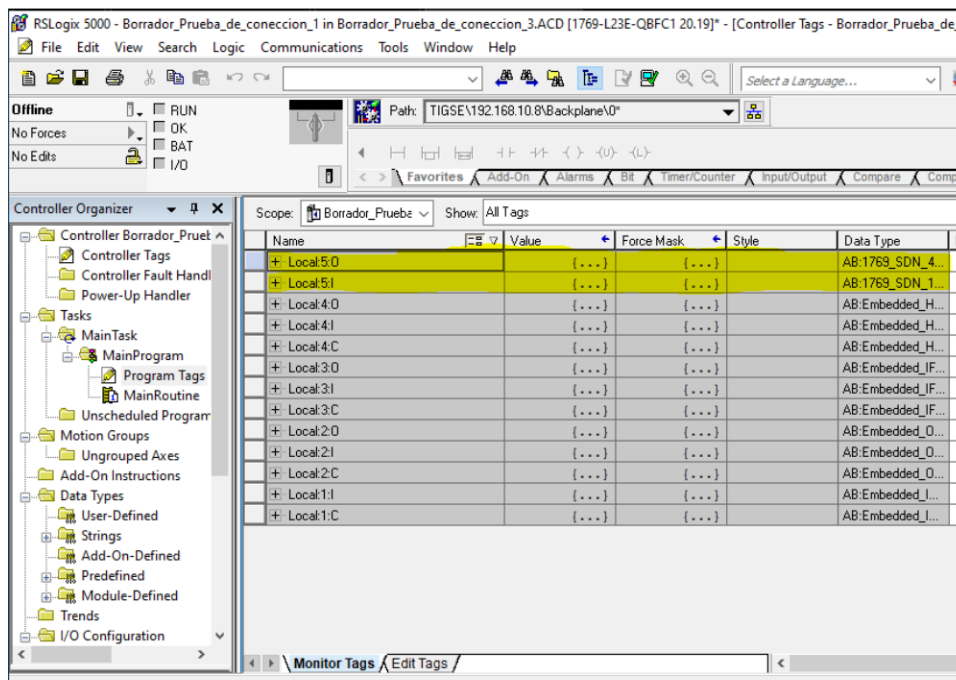


Figura 2.41. Registro para entrada y salidas de datos para red DeviceNet

Por último, al subir el programa al PLC el scanner no se encuentra en operación, para poner en marcha se debe acceder a un registro desde la localidad 5 del PLC configurada como salida para realizar un envío de dato hacia el scanner y cambiar a modo run.

Local:1:C	{...}	{...}		A
Local:1:I	{...}	{...}		A
Local:2:C	{...}	{...}		A
Local:2:I	{...}	{...}		A
Local:2:O	{...}	{...}		A
Local:3:C	{...}	{...}		A
Local:3:I	{...}	{...}		A
Local:3:O	{...}	{...}		A
Local:4:C	{...}	{...}		A
Local:4:I	{...}	{...}		A
Local:4:O	{...}	{...}		A
Local:5:I	{...}	{...}		A
Local:5:O	{...}	{...}		A
Local:5:O.CommandRegister	{...}	{...}		A
Local:5:O.CommandRegister.Run	1		Decimal	B
Local:5:O.CommandRegister.Fault	0		Decimal	B
Local:5:O.CommandRegister.DisableNetwork	0		Decimal	B
Local:5:O.CommandRegister.HaltScanner	0		Decimal	B
Local:5:O.CommandRegister.Reset	0		Decimal	B
Local:5:O.Data	{...}	{...}		D

Figura 2.42. Puesta en marcha del Scanner

2.5. VERIFICACIÓN DE CONEXIÓN A LA RED DE PROTOTIPO

Una vez efectuado todas las modificaciones tanto en la programación como en la configuración de la red DeviceNet se tiene en consideración que el dispositivo debe ser

capaz de realizar el envío de los datos de voltaje por medio de la red y el dispositivo maestro debe leer los datos y enviarlos al PLCs, para ello se tiene los correspondientes lineamientos de funcionamiento:

- Los niveles de voltaje que pueden ser ingresados a la entrada del sensor deben estar dentro de un rango de 0-10 voltios.
- Siempre al accionar la red física el dispositivo debe estar configurado en modo de operación normal.
- La nueva configuración de la red en el software RSNetWorx debe ser subido en el scanner.
- Al poner en marcha la red mediante el programa del PLC la lectura será de carácter entero por ende se requiere de los acondicionamientos necesarios para tener su equivalente en voltaje.

CAPÍTULO III

RESULTADOS, CONCLUSIONES Y RECOMENDACIONES

3.1. RESULTADOS

El dispositivo desarrollado es sometido a diversas pruebas de funcionamiento, las mismas que son detalladas a continuación:

Pruebas de niveles de voltaje

Se establece el nivel de voltaje para suministrar energía a la tarjeta de desarrollo Arduino Nano, además del correcto nivel de voltaje de entrada del conversor analógico dentro de un rango de 0-10 V DC. Con la finalidad de evitar quemar la tarjeta de desarrollo, tener suficientes niveles de voltaje para los diferentes módulos MCP2515 TJA1050, LCD 16x2 con módulo I2C, USB-TTL y comprobar la operación del divisor de voltaje con ganancia de 0.5 a la entrada del conversor analógico, se encuentre dentro de un rango de 0-5 VDC.

Además, se verifica los niveles de voltaje de los puertos de comunicación CAN-H y CAN – L se encuentre dentro de los rangos permitidos para la transmisión y lectura de datos de la red.

Prueba de lectura de número de nodo

El número de nodo se establece tanto en el dispositivo mediante los dipswitch y durante el diseño de la red en la plataforma de RSNetWorx, en ambos casos debe ser el mismo, con la finalidad proporcionar la matriz de Frame ID del esclavo y maestro acordes al número del nodo.

Prueba de establecimiento de conexión

El dispositivo realiza la lectura de los datos disponibles en la red DeviceNet, debe identificar y clasificar si el Frame ID corresponde al Scanner con la finalidad de proceder a iniciar el proceso de establecimiento de integración en la red al recibir el Frame ID del maestro con la estructura especificada en la Tabla 2.2.

Prueba de reconexión de dispositivo experimental

Se configura la red DeviceNet mediante el software RSNetWorx con un número de nodo distinto al número actual del dispositivo experimental, la señal de datos analógica no debe ser reconocida por el Scanner hasta localizar al dispositivo en el nuevo número de nodo configurado.

Prueba de envío de datos de señal analógica por medio de la red

Se propone el envío de diferentes niveles de voltaje que se encuentren dentro de los rangos permitidos de 0-10 V DC a través de la red, además de comprobar mediante equipos de medición de voltaje como por medio de la lectura de datos por medio del PLC CompactLogix.

Todas las pruebas realizadas permiten validar el dispositivo experimental al cumplir con los requerimientos necesarios de funcionamiento y operación al integrarse en la red DeviceNet.

3.1.1. PRUEBAS DE NIVEL DE VOLTAJE

Los niveles de voltaje de para suministrar la placa PCB de energía suficiente para cada uno de los componentes no debe ser menor a 10 V DC, eso se comprueba al medir en la salida del regulador de voltaje L7812GV es de 12 V DC.



Figura 3.1. Prueba de nivel de voltaje de alimentación para tarjeta de desarrollo Arduino Nano

El valor de voltaje en el pin número 7 de entrada del conversor analógico del Arduino Nano no supera los 5 V cuando el valor de la señal analógica es de 10 VDC como máximo.

Adicional se verifica los niveles de voltaje diferencial como se puede observar en la Figura 3.2 para determinar que se está realizando una correcta lectura de datos en el módulo MCP2515 TJA1050.

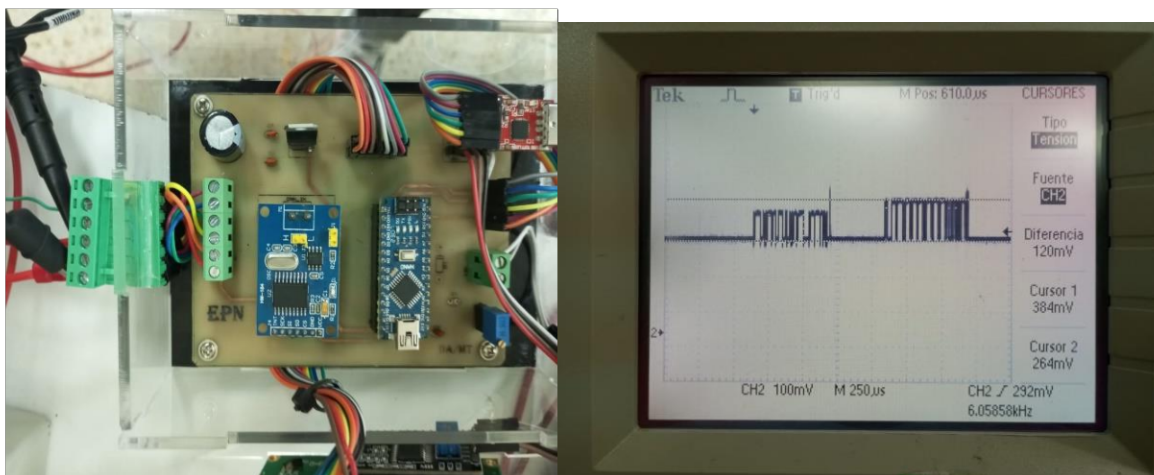


Figura 3.2. Prueba de niveles de voltajes en canales de comunicación CAN-L

El nivel de voltaje para los dos canales de comunicación debe de ser el mismo y encontrarse dentro del rango de operación tal como se puede observar en la Tabla 3.1

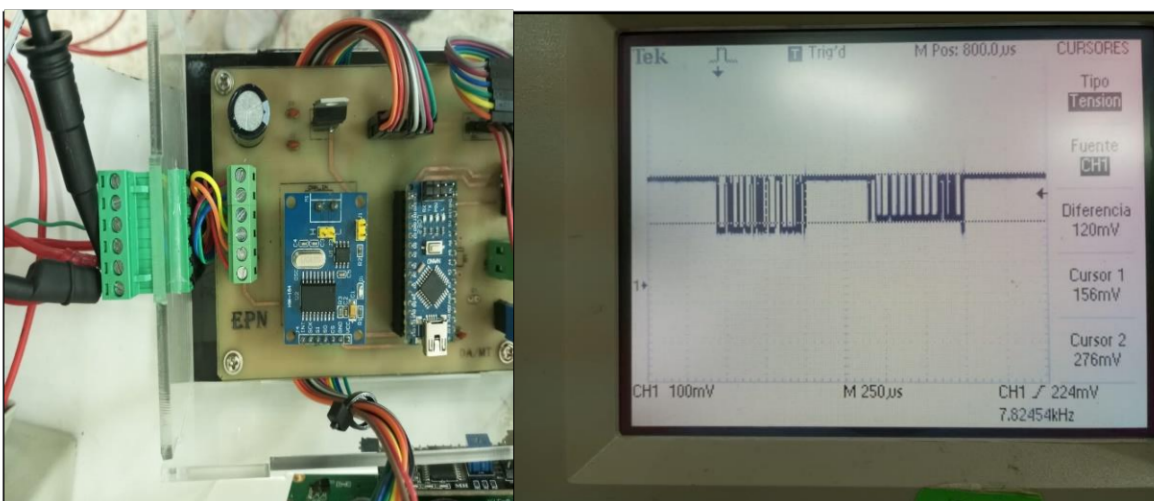


Figura 3.3. Prueba de niveles de voltajes en canales de comunicación CAN-H

En la Tabla 3.1 se determina las características de los niveles de voltaje mínimos y máximos que se deben encontrar al medir los canales de comunicación CAN.

Tabla 3.1. Valores permitidos de voltaje en los canales CAN-H y CAN-L

Valor de voltaje de canales CAN-H y CAN-L	
Voltaje Mínimo [V]	Voltaje Máximo [V]
1.5	3.5

3.1.2.PRUEBA DE LECTURA DE NÚMERO DE NODO

El dispositivo se encuentra de manera física por medio de los interruptores en el nodo número 10 para el caso de estudio, al iniciar la red la lectura se hace de manera inmediata y se crean las dos matrices una correspondiente al dispositivo esclavo y otra al maestro, donde ambos contienen el Frame ID para poder establecer conexión en la red.



Figura 3.4. Lectura de número de nodo

Al visualizar la Figura 3.4 el dispositivo muestra la información del Frame ID para el envío de datos desde el esclavo y el número de nodo actual.

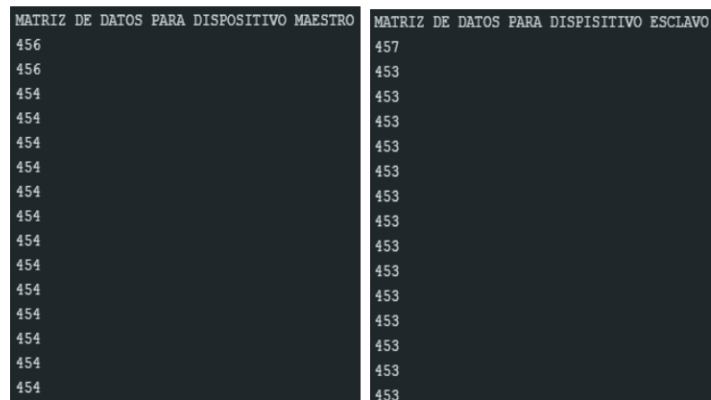


Figura 3.5. Creación de matrices de datos de Frame ID de maestro y esclavo para el proceso de establecimiento de conexión

Al visualizar la Figura 3.5 una vez que el dispositivo determina el número de nodo se crean las matrices. Cada matriz debe contener 15 datos que corresponden al proceso de establecimiento de conexión.

Las características de los datos creados para el maestro y el esclavo que se pueden visualizar en la Figura 3.5 se muestran en la Tabla 3.2

Tabla 3.2. Características de datos creados en la lectura del nodo del dispositivo

# Datos	Frame ID de esclavo	Características	Frame ID de maestro	Características
1	457	Respuesta para inicio de comunicación	456	Solicitud para inicio de comunicación
2	453	Respuesta a solicitud de información	456	solicitud de información de dispositivo
3	453		454	
4	453		454	
5	453		454	
6	453		454	
7	453		454	
8	453		454	
9	453		454	
10	453		454	
11	453	Respuesta de dato de entrada tipo encuesta	454	Solicitud de dato de entrada tipo encuesta
12	453	Respuesta de dato de entrada tipo cambio de estado	454	Solicitud de dato de entrada tipo cambio de estado
13	453	Respuesta de solicitud de datos de salida tipo encuesta	454	Solicitud de datos de salida tipo encuesta
14	453	Respuesta de solicitud de datos tipo cambio de estado	454	Solicitud de datos de salida tipo cambio de estado
15	453	Respuesta de tasa de paquetes estimada	454	Solicitud de tasa de paquetes

La siguiente Tabla 3.3 presenta tres ejemplos de cambio de número de nodo mediante los interruptores:

Tabla 3.3. Configuración del número de nodo mediante los interruptores físicos.

Bits de interruptores para número de nodo del dispositivo								Número de nodo
Orden de interruptores / estado lógico	7	6	5	4	3	2	1	
Ejemplo 1	1	0	0	1	0	1	0	10
Ejemplo 2	1	0	0	0	0	1	0	2
Ejemplo 3	1	0	0	1	1	1	0	14

El bit 7 correspondientes al último interruptor que se encuentra activado para mantener al dispositivo en modo de lectura, para comprobar los resultados se muestra en la pantalla LCD el número de nodo configurado y el Frame ID del esclavo para envío de datos.

En la siguiente Tabla 3.4 se muestra los cambios del Frame ID dependiendo del número de nodo del dispositivo

Tabla 3.4. Cambios de Frame ID inicial para establecer conexión usando 3 números de nodos diferentes

# Nodo	Frame ID Hexadecimal	Frame ID Binario									Descripción		
		G2		MAC ID						MENSAJE ID			
10	456	1	0	0	0	1	0	1	0	1	1	0	ID para inicio de comunicación
		2		10						6			
2	416	1	0	0	0	0	0	1	0	1	1	0	
		2		2						6			
14	476	1	0	0	0	1	1	1	0	1	1	0	
		2		14						6			

Al visualizar la Tabla 3.4 el Frame ID cambia conforme se modifica el número de nodo del dispositivo debido a que se modifica el MAC ID en la estructura del Frame ID.

3.1.3. PRUEBA DE ESTABLECIMIENTO DE CONEXIÓN



Figura 3.6. Dispositivo en modo de operación al desactivar interruptor 7

Una vez que el dispositivo se encuentra en modo de operación, espera para poder recibir el Frame ID para inicio de comunicación. El dispositivo maestro debe enviar mediante la red un Frame ID idéntico al de la Tabla 3.4 que corresponde a la solicitud de inicio de comunicación entre dispositivo maestro y esclavo.

```

N: 1 ID_EN_RED: 454 Número de dato: 4 No se encuentran mensaje en la red
ID_EN_RED: 456 Número de dato: 1 ID_Maestro: 454 NDatos: 5 0 E 1 1 2 N: 2
ID_Maestro: 456 NDatos: 6 0 4B 3 1 1 0 DatosEs: 0 8E 6 0 ID_EN_RED: 432 Número de dato: 8
DatosEs: 0 CB 0 N: 2 ID no pertenece a PRecepcion
N: 2 No se encuentran mensaje en la red N: 2
No se encuentran mensaje en la red ID_EN_RED: 454 Número de dato: 5 ID_EN_RED: 454 Número de dato: 8
N: 2 ID_Maestro: 454 NDatos: 5 0 0 0 0 0 ID_Maestro: 454 NDatos: 7 0 10 2B 1 1 10 0
DatosEs: 0 8E 2C 0 DatosEs: 0 90 10 0
N: 2 N: 2
ID no pertenece a PRecepcion ID_EN_RED: 432 Número de dato: 9
N: 2 ID no pertenece a PRecepcion
ID_EN_RED: 454 Número de dato: 2 ID_Maestro: 454 NDatos: 5 0 E 1 1 3 N: 2
ID_Maestro: 454 NDatos: 6 0 4B 3 1 10 0 DatosEs: 0 8E 1 6 ID_EN_RED: 454 Número de dato: 9
DatosEs: 0 CB 0 N: 2 ID_Maestro: 454 NDatos: 7 0 10 5 4 11 1 0
No se encuentran mensaje en la red DatosEs: 0 90 8 0
N: 2 N: 2
ID no pertenece a PRecepcion ID_EN_RED: 432 Número de dato: 7 ID no pertenece a PRecepcion
N: 2 ID no pertenece a PRecepcion ID_EN_RED: 454 Número de dato: A
ID_EN_RED: 432 Número de dato: 3 ID_Maestro: 454 NDatos: 6 0 10 5 1 C 3 ID_Maestro: 454 NDatos: 7 0 10 5 4 11 1 0
ID no pertenece a PRecepcion DatosEs: 0 94 16 FF
N: 2 N: 2
ID_EN_RED: 454 Número de dato: 3 ID_Maestro: 454 NDatos: 5 0 E 1 1 1 ID no pertenece a PRecepcion
ID_Maestro: 454 NDatos: 6 0 8E 1 0 ID_EN_RED: 454 Número de dato: B
DatosEs: 0 8E 1 0 ID_Maestro: 454 NDatos: 5 0 E 5 2 7
ID no pertenece a PRecepcion DatosEs: 0 8E 1 0
ID_EN_RED: 432 Número de dato: 8

```

Figura 3.7. Lectura de datos y respuesta de dispositivo en la red DeviceNet

El dispositivo rechaza cualquier dato que no pertenece al FrameID de la matriz de datos del maestro creada en la sección de lectura y que se puede visualizar en la Tabla 3.2., para el caso de estudio el prototipo se encuentra configurado en el nodo 10.

```

ID_EN_RED: 454 N_DE_DATOS: C
ID_MAESTR0: 454 N_DATOS_RECIBIDOS: 5 INFORMACION DE DATOS 0 E 5 2 8
TRANSMISION DATOS DEL ESCALVO: 0 94 16 FF
N: 2
ID_EN_RED: 432 N_DE_DATOS: D
ID NO PERTENECE A INFORMACION DE LA MATRIZ ID DEL MAESTRO
N: 2
ID_EN_RED: 454 N_DE_DATOS: D
ID_MAESTR0: 454 N_DATOS_RECIBIDOS: 5 INFORMACION DE DATOS 0 E 5 4 8
TRANSMISION DATOS DEL ESCALVO: 0 8E 0 0
N: 2
ID_EN_RED: 432 N_DE_DATOS: E
ID NO PERTENECE A INFORMACION DE LA MATRIZ ID DEL MAESTRO
N: 2
ID_EN_RED: 454 N_DE_DATOS: E
ID_MAESTR0: 454 N_DATOS_RECIBIDOS: 7 INFORMACION DE DATOS 0 10 5 4 9 FA 0
TRANSMISION DATOS DEL ESCALVO: 0 90 0 1
Conexión exitosa

```

Figura 3.8. Proceso de establecimiento de conexión

La secuencia de establecimiento de conexión debe ser de manera secuencial, si el dispositivo esclavo tarda mucho en contestar la petición del maestro, el scanner vuelve a intentar reconectarse nuevamente para realizar la secuencia de comunicación de manera exitosa.

En la siguiente Tabla 3.5 se detalla las características de los datos recibidos con el correcto Frame ID del maestro y datos de respuesta del dispositivo experimental:

Tabla 3.5. Características de datos recibidos del maestro y respuesta del dispositivo esclavo

# de proceso	Tipo de dispositivo	# de datos	Datos							Características
1	Scanner	7	0	4B	3	1	1	0	Solicitud y respuesta de inicio de conexión con dispositivo esclavo	
	Prototipo		0	1	0	2B	F5	A 50		
2	Scanner	7	0	4B	3	1	1	0	Solicitud de características de dispositivo esclavo con sus correspondientes respuestas	
	Prototipo		0	CB	0					
3	Scanner	8	0	4B	3	1	10	0		
	Prototipo		0	CB	0					
4	Scanner	5	0	E	1	1	1			
	Prototipo		0	8E	1	0				
5	Scanner	5	0	E	1	1	2			
	Prototipo		0	8E	6	0				
6	Scanner	5	0	E	1	1	3			
	Prototipo		0	8E	2C	0				
7	Scanner	5	0	E	1	1	4			
	Prototipo		0	8E	1	6				
8	Scanner	6	0	10	5	1	C	3		
	Prototipo		0	90						
9	Scanner	7	0	10	2B	1	1	10 0		
	Prototipo		0	90	10	0				
10	Scanner	7	0	10	5	4	11	1 0		
	Prototipo		0	90	8	0				
11	Scanner	5	0	E	5	2	7		Datos para solicitar y responder sobre el tipo de dato de entrada tipo encuesta	
	Prototipo		0	94	16	FF				
12	Scanner	5	0	E	5	4	7		Datos para solicitar y responder sobre el tipo de dato de entrada tipo cambio de estado	
	Prototipo		0	8E	1	0				
13	Scanner	5	0	E	5	2	8		Datos para solicitar y responder sobre el tipo de dato de salida tipo encuesta	
	Prototipo		0	94	16	FF				
14	Scanner	5	0	E	5	4	8		Datos para solicitar y responder sobre el tipo de dato de salida tipo encuesta	
	Prototipo		0	8E	0	0				

15	Scanner	7	0	10	5	4	9	FA	0	Datos para solicitar y responder para número de datos estimados
	Prototipo		0	90	0	1				

El dispositivo experimental realiza el proceso de establecimiento de conexión de manera satisfactoria, pero se observa que en la red diseñada en el software RSNetWorx, el dispositivo se encuentra como desconectado como se visualiza en la Figura 3.9.

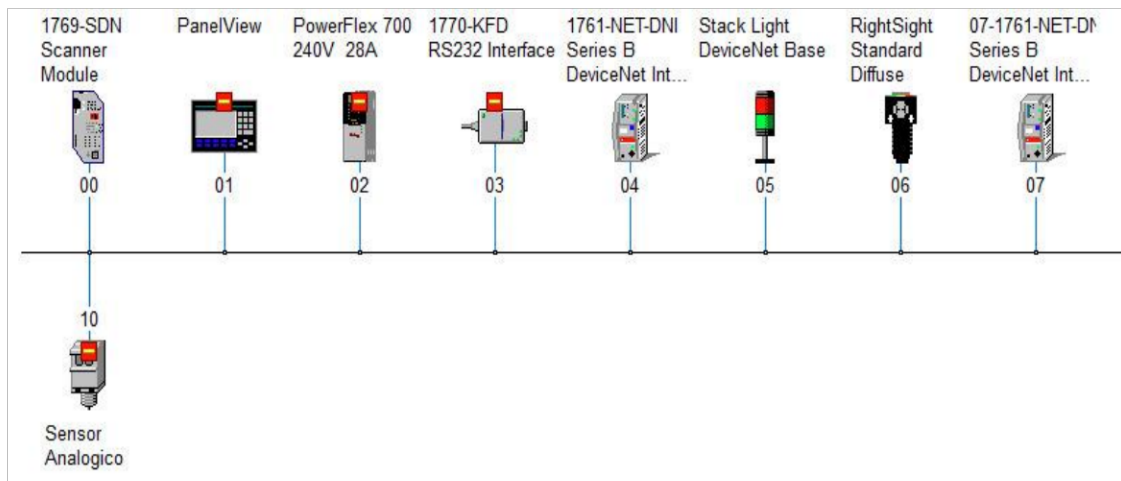


Figura 3.9. Monitoreo de dispositivo experimental en la red

Pero el scanner sí reconoce al dispositivo y lo integra en la red, esto se puede observar en la secuencia de código de errores en la pantalla del scanner como se puede visualizar en la Figura 3.10.

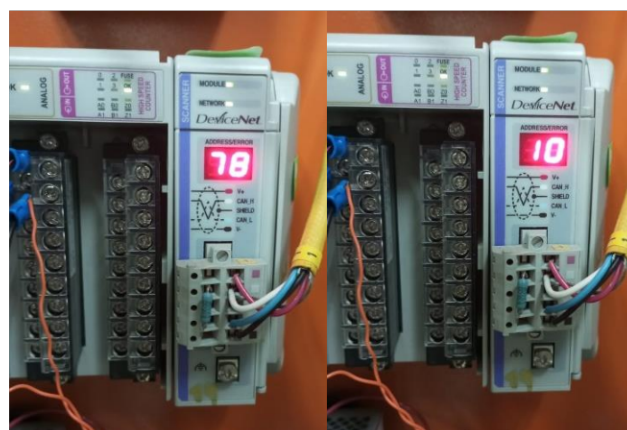


Figura 3.10. Secuencia de error de conexión de dispositivo de nodo número 10

En la siguiente Tabla 3.6 se especifican los errores más comunes identificados al momento de integrar el dispositivo en la red:

Tabla 3.6. Tabla de errores frecuentes al integrar el dispositivo en la red.

Número de error	Característica
78	Dispositivo no existente en la red
84	Error en establecimiento de conexión con dispositivo esclavo
80	Módulo scanner se encuentra inactivo

El dispositivo realiza el proceso completo de establecimiento de conexión el error 78 desaparece, si durante el proceso un dato enviado por el dispositivo esclavo no es identificado por el maestro se solicita nuevamente el dato, pero el sensor envía un dato erróneo entonces se origina el error 84. La solución al problema es cargar nuevamente la red en el scanner para que nuevamente realice el proceso de conexión con los dispositivos esclavos

3.1.4. PRUEBA DE RECONEXIÓN DE DISPOSITIVO EXPERIMENTAL

Para el proceso de reconexión se somete al dispositivo a tres tipos de números de nodo los que se pueden observar en la Tabla 3.4 mediante la configuración de la red por medio del software RSNetWorx. Además, una vez cargado el esquema de la nueva red en el scanner, el dispositivo puede volver a conectarse de forma automática si se produce algún corte de energía.

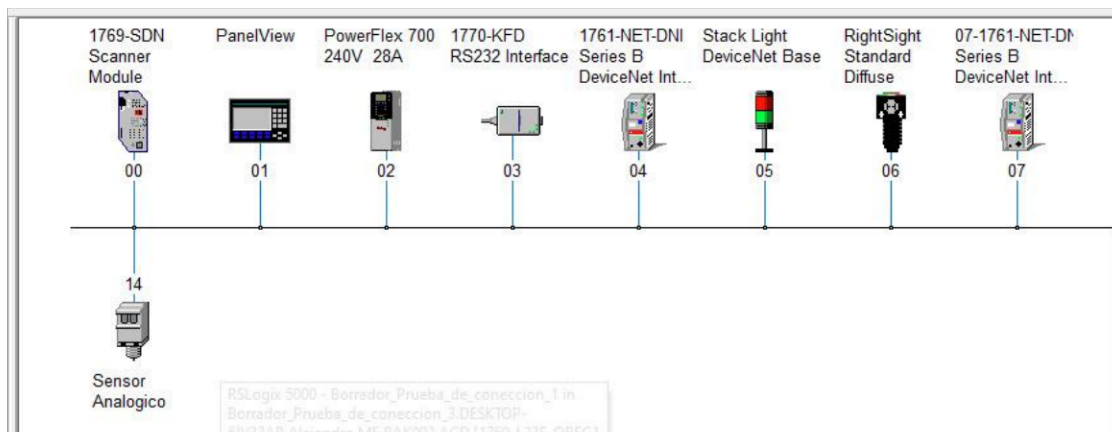


Figura 3.11. Cambio de número de nodo en dispositivo experimental

Una vez se carga la nueva configuración de la red al módulo scanner, los datos enviados por el dispositivo esclavo no se reconocen por el PLC.

Para iniciar nuevamente la comunicación entre dispositivo esclavo y maestro, se debe activar el interruptor 7 y realizar el cambio del número de nodo como se observa en la Figura 3.12

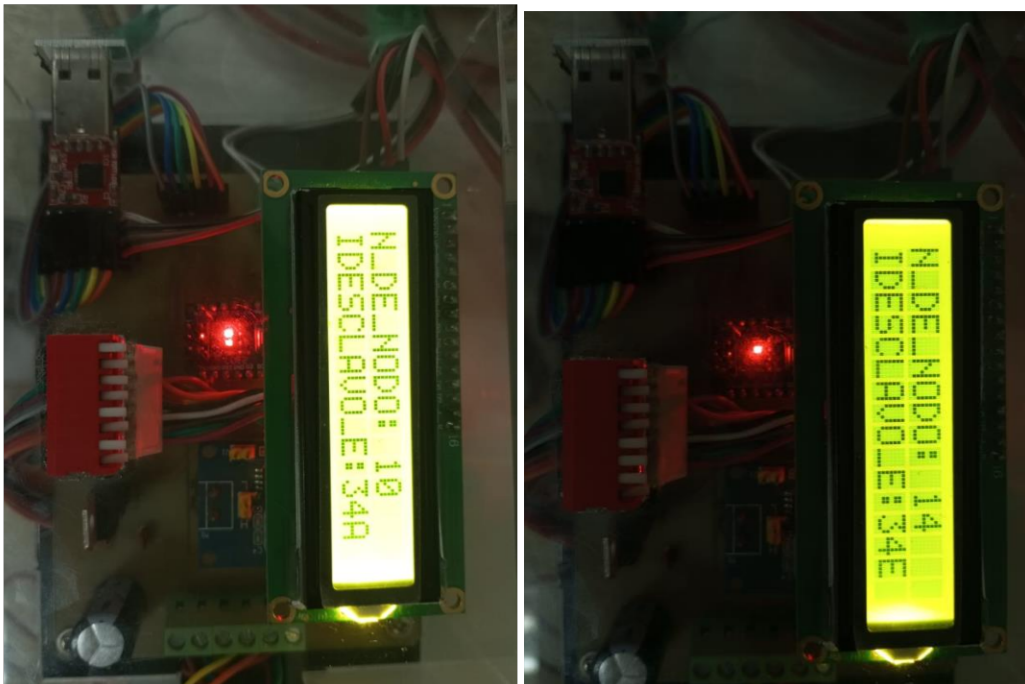


Figura 3.12. Cambio de número de nodo de dispositivo

Una vez se modifica el número de nodo se desactiva el interruptor 7 y el dispositivo espera a recibir el ID para iniciar comunicación con el maestro y repetir el proceso de establecimiento de conexión.

3.1.5. PRUEBA DE ENVIÓ DE DATOS DE SEÑAL ANALÓGICA POR MEDIO DE LA RED

Si el proceso de establecimiento de conexión del prototipo de dispositivo de sensor analógico con el maestro (scanner) se realiza de manera exitosa, el envío de datos de la señal analógica se realiza mediante el Frame ID descrito en la Tabla 2.3.

En la siguiente Tabla se tiene tres Frame ID para envío de datos según se modifica el número de nodo del dispositivo:

Tabla 3.7. Frame ID para envío de datos de señal analógica con distintos números de nodo

Frame ID Hexadecimal	Frame ID de esclavo										Descripción	
	G	MENSAJE ID				MAC ID						
0X34A	0	1	1	0	1	0	0	1	0	1	0	ID para envío de datos una vez integrado en la red
	1	13				10						
0X342	0	1	1	0	1	0	0	0	0	1	0	
	1	13				2						
0X34E	0	1	1	0	1	0	0	1	1	7	0	
	1	13				14						

Los datos que transmite el dispositivo experimental se envían de manera correcta a través de la red DeviceNet, el scanner actúa como dispositivo maestro para la red y como un intermediario para ordenar todos los datos y enviarlos al PLCs mediante los registros de localidad de memoria.

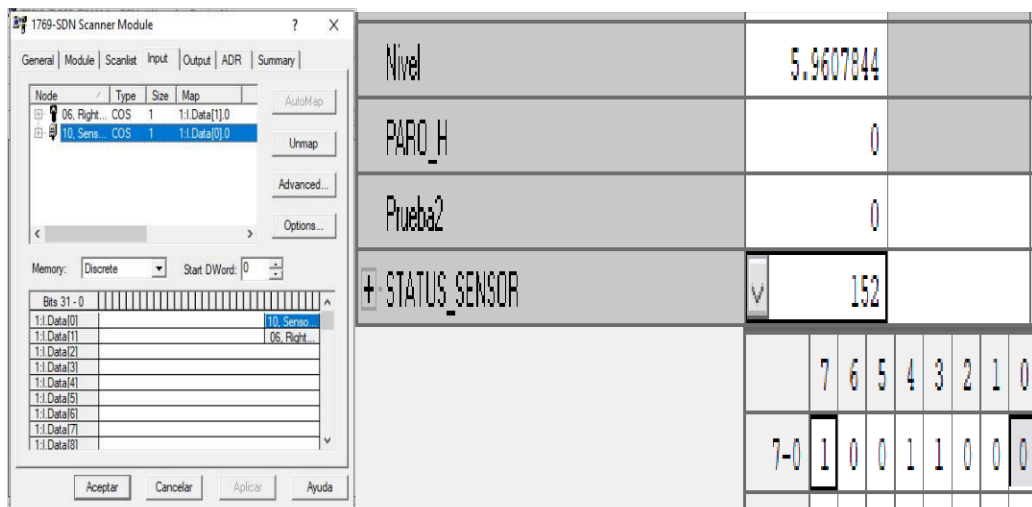


Figura 3.13. Lectura de datos del registro de memoria del scanner

Como se observo en la Figura 3.9 el dispositivo experimental no se reconoce en la red diseñada en el software RSNetWorx.

La conexión con el módulo scanner se realiza de manera exitosa y los datos enviados por el sensor prototipo son reconocidos y guardados en la localidad de memoria configurada en el software como se visualiza en la Figura 3.13

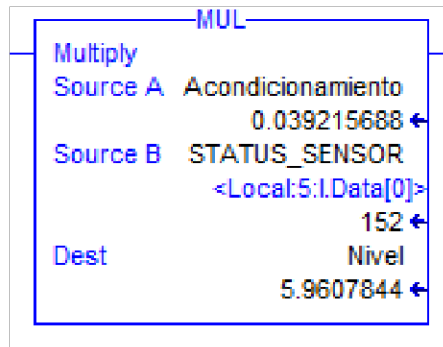


Figura 3.14. Lectura de datos mediante programa de PLC provenientes de la red DeviceNet

La lectura de los valores de voltaje realizadas por el dispositivo dependerá mucho de la resolución del conversor analógico interno del microcontrolador y de poder calibrar de manera eficiente mediante el potenciómetro de precisión el valor de la ganancia a 0.5.



Figura 3.15. Lectura de voltaje realizada por el multímetro

En la siguiente Tabla 3.8 se realiza una comparación entre los distintos valores de voltajes realizados mediante la prueba del dispositivo:

Tabla 3.8. Valores de niveles de voltaje:

Voltajes	Multímetro 1	Dispositivo	PLC
Medición 1	2,03	2,04	2,02
Medición 2	4,52	4,55	4,55
Medición 3	1,99	2	2

Al realizar el correspondiente análisis de los valores de voltaje siendo el valor del multímetro el valor de voltaje real se tiene los siguientes errores en porcentaje mostrados en la Tabla 3.9

Tabla 3.9. Cálculo de errores

Error de lectura %	Error de transmisión %
0.49	0.49
0.66	0.66
0.50	0.50

Los valores de los errores en las tres mediciones no superan el 1%, por lo tanto, las mediciones tanto por el dispositivo como por la transmisión de datos a través de la red DeviceNet son de un nivel tolerable.

Se menciona que la lectura de los datos del dispositivo es recibida a pesar de que el scanner no se encuentra en modo Run, eso se puede observar en la pantalla provista en el scanner con una secuencia de 80-00 para identificar que el scanner no está en marcha.

3.2. CONCLUSIONES

Se ha realizado la implementación e integración del prototipo de dispositivo analógico en la red DeviceNet y cumple con el principal objetivo de realizar la comunicación entre dispositivo maestro(scanner) y dispositivo esclavo (sensor analógico) y con el envío de la trama de datos correspondientes a la señal de voltaje de la entrada del dispositivo experimental a través de la red DeviceNet y de la lectura del dato en el PLC CompactLogix.

En el proceso de la programación del prototipo de dispositivo analógico se establece que la configuración del tipo de trama que debe manejar se encuentra relacionado con la que maneja la red DeviceNet, siendo del tipo de trama estándar que establece un número de 11 bits para la sección del identificador que permite diferenciar cada uno de los dispositivos que se encuentran en la red DeviceNet.

Uno de los elementos claves al momento diseñar y de programar el dispositivo es la configuración del FrameID para establecer la conexión entre maestro y esclavo, debido a que el FrameID está relacionado directamente al tipo de dispositivo, las capacidades, el tipo de datos a transmitir o a recibir cuando la conexión ha sido exitosa.

El prototipo de dispositivo analógico una vez que ha realizado el proceso de conexión y de manera exitosa los datos del valor de voltaje analógico son enviados a través de la red, cada dato posee un tiempo de espera, durante el envío de datos se verifica que la colisión de información no se da al visualizar que los datos se mantienen en el programa del PLC y únicamente cambian si varía el valor de voltaje a la entrada del sensor.

Al momento de cambiar de número de nodo a través de los dipswitch cuando ya el dispositivo se encuentra enviando datos al scanner, la conexión se pierde de manera inmediata y los datos del dispositivo experimental no son leídos por el scanner hasta volver a realizar el proceso de establecimiento de conexión tras colocar el número de nodo correcto en el dispositivo de manera física en los dipswitch.

Durante el desarrollo del prototipo, el mayor inconveniente al momento de realizar la comunicación del prototipo dentro de la red, esto debido a que el tiempo de reacción del microcontrolador usado para responder al mensaje del maestro esto por las propias limitantes del procesador y por la secuencia de programación que realiza internamente, los tiempos de retardo programados tanto para imprimir los datos en el LCD como por el puerto comunicación UART ocasionan pequeños retardos en la lectura y envío de los datos al momento de realizar el proceso de conexión. Para ello se reduce los tiempos de retardo en el programa del microcontrolador del dispositivo, de igual manera se elimina las impresiones de pantalla innecesarios del LCD y de la comunicación UART.

Es posible integrar nuevos dispositivos prototipos a la red DeviceNet siempre que cumplan con todos los requisitos tales como manejo de los protocolos CAN para la capa física y CPI para la capa de comunicación, de contar con un archivo EDS para agregar en los archivos de la red DeviceNet, así como los niveles de voltaje diferencial y velocidades de transmisión, además que es un requerimiento fundamental el contar con el archivo EDS para permitir que el scanner pueda reconocer al nuevo dispositivo y sobre todo que el dispositivo experimental maneje a nivel de capa física el protocolo CAN y a nivel de capa de transmisión el protocolo CIP que juntos integran la comunicación por DeviceNet.

En el diseño del reductor de voltaje que va desde un nivel de 0-10 y pasa a un nivel de 0-5, se descarta el uso de un amplificador operacional con ganancia de 0.5 al tener que contar con una segunda fuente de voltaje de valor negativo para asegurar un nivel de voltaje cero cuando ningún valor de lectura del sensor por lo tanto con la finalidad de utilizar la fuente de voltaje de la red para alimentar el dispositivo experimental y contar con el nivel de voltaje necesario para mantener en operación el sensor se utiliza un divisor de voltaje en la entrada analógica del microcontrolador.

3.3. RECOMENDACIONES

Durante el proceso de implementación el nuevo archivo EDS del prototipo de dispositivo diseñado, se debe considerar que parte del EDS de un dispositivo ya existente en los archivos del software RSNetWorx y esto actualmente origina conflicto al momento de poner

en línea la red modificada, por lo que se recomienda desarrollar una nueva estructura de EDS para el caso de la implementación de un dispositivo totalmente nuevo.

Al momento de seleccionar los elementos para el diseño de dispositivos experimentales es necesario verificar que la tarjeta de desarrollo y el módulo de expansión puedan trabajar con la misma velocidad de la red la se integrará el dispositivo con la finalidad de poder realizar la lectura de los datos durante la conexión.

Previo a la conexión del dispositivo desarrollado a la red verificar todas las conexiones, el buen estado de los cables y el correcto funcionamiento de la fuente de 24 Voltios para evitar el ingreso de ruido por cables sueltos, cortocircuitos y que el dispositivo experimental no se conecte a la red.

Considerar que el dispositivo se encuentra en un ambiente lleno de señales de ruido y que pueden perjudicar la lectura de los valores de voltaje de entrada hacia el sensor analógico originando falsos resultados de mediciones de voltaje y por ende datos erróneos que son enviados a través de la red, para ello se debe agregar capacitores a la entrada del pin A7 de la tarjeta del Arduino Nano con la finalidad de poder reducir las señales invasivas.

Evitar el uso de circuitos integrados reductores de voltaje debido a que por acción del ruido que genera la propia red pueden operar de manera incorrecta y causar daños al dispositivo experimental y a la propia red DeviceNet.

Para evitar manipular y dañar algún componente de la placa al momento de realizar tanto la conexión de los cables de alimentación y de transmisión de datos como para realizar los cambios del número de nodo mediante los interruptores físico es necesario el implementar borneras que se encuentren separados de la placa.

Cargar la configuración de la red elaborado en el software RSNetWorx hacia el scanner antes de establecer comunicación con el dispositivo para evitar generar errores en el proceso de conexión y que el scanner interprete una mal conexión y descarte toda la información del dispositivo.

Es posible utilizar un microcontrolador con mejores características de procesamiento de datos y que pueda utilizar de manera directa el protocolo de comunicación CAN como es el caso de una tarjeta de desarrollo de la familia STM32 que posee puertos dedicados a la comunicación CAN y solamente requiere de un módulo para operar con voltajes diferenciales correspondiente al puerto de comunicación CAN-L y CAN-H.

Para obtener una mejor resolución de la señal analógica enviada desde el prototipo de dispositivo analógico se puede considerar el envío de dos mensajes que contienen los

datos de la señal de voltaje divididos en 8 bits cada uno con la finalidad de reducir el error, además de evitar comprimir únicamente en 8 bits los datos de la señal analógica y por ende tener una menor resolución.

REFERENCIAS BIBLIOGRÁFICAS

- [1] N. Ruddy, "Diseño y configuración de una red DeviceNet manual del estudiante", jul, 2017. Consultado: el 15 de noviembre de 2023. [En línea]. Disponible en: https://es.slideshare.net/niloruddy/diseo-y-configuracin-de-una-red-device-net-manual-del-estudiante?from_action=save
- [2] A.Lopez, E. León, E. Bravo, "Comunicación entre un escáner DeviceNet autónomo y una aplicación software mediante Modbus/TCP", may, 2012. Consultado: el 17 de noviembre de 2023. [En línea]. Disponible en: https://www.researchgate.net/publication/305296654_Comunicacion_entre_un_escaner_DeviceNet_autonomo_y_una_aplicacion_software_mediante_MODBUSTCP
- [3] Logicbus, "DeviceNet", jun, 2019. Consultado: el 17 de noviembre de 2023. [En línea]. Disponible en: <https://www.logicbus.com.mx/blog/devicenet/>
- [4] WEG, "Manual de comunicación DeviceNet", mar, 2008. Consultado: el 18 de noviembre de 2023. [En línea]. Disponible en: <https://static.weg.net/medias/downloadcenter/h09/hfe/WEG-ssw07-manual-de-la-comunicacion-devicenet-10000046974-manual-espanol.pdf>
- [5] G. Gamboa, "Implementación de prototipos de drivers de comunicación y hojas de datos electrónicas (EDS) DeviceNet de bajo costo para sensores industriales Analógicos y discretos", Nov, 2018. Consultado: el 14 de noviembre de 2023. [En línea]. Disponible en: <https://bibdigital.epn.edu.ec/bitstream/15000/19856/1/CD-9264.pdf>
- [6] Rockwell Automation "Protocolos de comunicación industrial", 2023. Consultado: el 20 de noviembre de 2023 [En línea]. Disponible en: <https://www.rockwellautomation.com/es-mx/capabilities/industrial-networks.html#:~:text=CIP%20es%20un%20protocolo%20a,control%20de%20movimiento%20e%20informaci%C3%B3n>.
- [7] G. Prado, H. Toro, "Protocolos de comunicación industrial", jul. 2018. Consultado: el 25 de noviembre de 2023. [En línea]. Disponible en: <https://repositorio.ecci.edu.co/bitstream/handle/001/1874/Informe%20de%20seminario.pdf?sequence=1&isAllowed=y>
- [8] KVASER, "El Tutorial Del Protocolo De Bus CAN", 2023. Consultado: el 30 de noviembre de 2023. [En línea]. Disponible en: <https://www.kvaser.com/can-protocol-tutorial/#>
- [9] A. Martínez, "Introducción a CAN bus: Descripción, ejemplos y aplicaciones de tiempo real", Universidad Politécnica de Madrid, jul. 2017. Consultado: el 13 de noviembre de 2023. [En línea]. Disponible en: https://oa.upm.es/48061/10/TFM_ADRIAN_MARTINEZ_REQUENA.pdf

- [10] S. Bonilla, R. Mesias, “ Construcción de un modelo didáctico para la iluminación del vehículo controlado con sistema CAN BUS, para el laboratorio de la Escuela de Ingeniería Automotriz”, mar. 2012. Consultado: el 21 de noviembre de 2023. [En línea]. Disponible en: <http://dspace.espace.edu.ec/handle/123456789/2279>
- [11] A. Fuambuena, “Guestion de un interfaz USB-CAN (Configuración)”, jul. 2012. Consultado: el 5 de diciembre de 2023. [En línea]. Disponible en: <https://riunet.upv.es/bitstream/handle/10251/16934/Memoria.pdf?sequence=1>
- [12] D. Chavez, “SISTEMA ELECTRÓNICO DE CONTROL Y MONITOREO VEHICULAR A DISTANCIA MEDIANTE TECNOLOGÍA INALÁMBRICA.”, oct. 2015. Consultado: el 1 de diciembre de 2023. [En línea]. Disponible en: https://repositorio.uta.edu.ec/bitstream/123456789/15104/1/Tesis_t1067ec.pdf
- [13] ARM MBED, “STM32F103C8T6”, jul. 2016. Consultado: el 10 de diciembre. [En línea]. Disponible en: https://os.mbed.com/users/hudakz/code/STM32F103C8T6_Hello/
- [14] MICROCHIP, “MCP2551”, 2016. Consultado: el 22 de diciembre. [En línea]. Disponible en: <https://ww1.microchip.com/downloads/en/devicedoc/20001667g.pdf>
- [15] GRINI, “Arduino con ESP32”, dic. 2018. Consultado: el 21 de diciembre. [En línea]. Disponible en: <http://blog.espol.edu.ec/gimi/ide-arduino-con-esp32/>
- [16] MARKERS, “Arduino Nano con cargador CH340”. 2023. Consultado: el 13 de diciembre. [En línea]. Disponible en: <https://makerselectronics.com/product/arduino-nano-with-ch340-uploader-usb-type-c>
- [17] EBAY, “1PC MCP2515 TJA1050”, 2023. Consultado: el 5 de noviembre. [En línea]. Disponible en: <https://www.ebay.com/itm/225713108199?chn=ps>
- [18] TECmikro, LCD 16X2 CON I2C “”, 2023. Consultado: el 21 de diciembre. [En línea]. Disponible en: <https://tecmikro.com/led-lcd/351-lcd-16x2-i2c.html>
- [19] TESLAELECTRONIC, “L7812 Regulador de Voltaje Positivo”, 2023. Consultado: el 17 de diciembre. [En línea]. Disponible en: <https://www.teslaelectronic.com.pe/producto/l7812-regulador-de-voltaje-positivo/>
- [20] ALPHARTRONIC, “USB to TTL Convertidor CP2102”, 2023. Consultado: el 20 de diciembre. [En línea]. Disponible en: <https://alphatronic.lk/product/usb-to-ttl-converter/>
- [21] Allen Bradley, “CompactLogix 1769-L35E PA4 IQ32 OB32 IQ1 IF4 SDN ECR”, 2023. Consultado: el 19 de diciembre. [En línea]. Disponible en: https://literature.rockwellautomation.com/idc/groups/literature/documents/um/1769-um011_-en-p.pdf

ANEXOS

ANEXO I. Características de elementos que conforman el dispositivo

ANEXO II. Descripción de la placa PCB del dispositivo.

ANEXO III. Guía de procedimiento de integración de prototipo de dispositivo en la red DeviceNet

ANEXO IV. Descripción de procedimiento ante falla de proceso de comunicación del dispositivo.

ANEXO V. Programa de prototipo de dispositivo en tarjeta de desarrollo Arduino Nano.

ANEXO I

Características de elementos que conforman el dispositivo

Las características de operación para el funcionamiento de los elementos que integran el dispositivo se presentan en la siguiente tabla:

Tabla 4.1. Especificaciones de operación de elementos que conforman el dispositivo

Nombre del elemento	Características
Tarjeta de desarrollo Arduino Nano	Procesador: Atmega328 Alimentación: 7-12 VDC Pines digitales: 14 (D0-D13) Pines Analógicos: 6 (A0-A5) Reloj: 16 MHZ Fuente: 3.3 VDC y 5 VDC
Módulo MCP2515 TJA1050	Alimentación: 5VDC Comunicación: CAN Control de protocolo: interfaz SPI Velocidad de comunicación: 1MB/s
LCD 16x2 con módulo I2C	Alimentación: 5 VDC Compatibilidad con Arduino Protocolo de comunicación: I2C
Regulador de voltaje L7812GV	Voltaje de entrada: 14-35VDC Voltaje de salida: 12VDC Corriente de salida: 1.5 A
USB-TTL	5 pines: 3.3VDC, TX, RX, GND y 5VDC Comunicación: UART
Potenciómetro de precisión	10K Ohm
Capacitores	Electrolítico: 1000uF 50VDC Cerámico: 1x104 y 2x103

ANEXO II

Descripción de la placa PCB del dispositivo.

Para facilitar el cambio de componentes que se encuentren averiados en la placa, a excepción de los capacitores, todos los demás elementos cuentan con sócalos para su remoción sin la necesidad de volver a desoldar el elemento como se visualiza en la Figura 4.1.

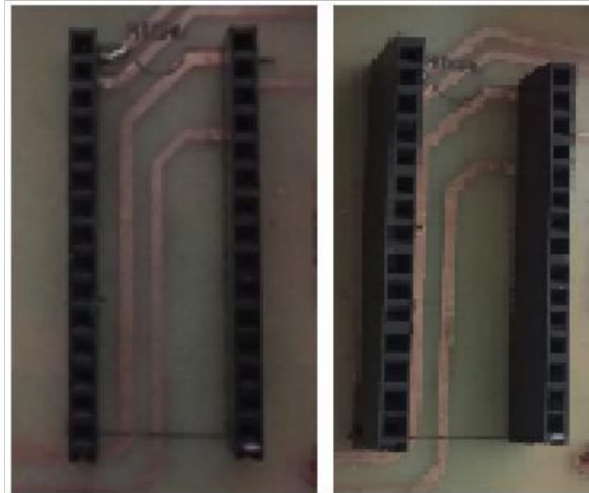


Figura 4.1. Ejemplo de sócalo para tarjeta de Arduino Nano para remoción ante alguna avería

Para evitar la manipulación de la placa al momento de realizar la conexión de los pines tanto de la fuente de 24VDC como de los pines de comunicación CAN-L y CAN-H de la red DeviceNet y la entrada de voltaje para el sensor analógico, las borneras son de igual manera removibles.

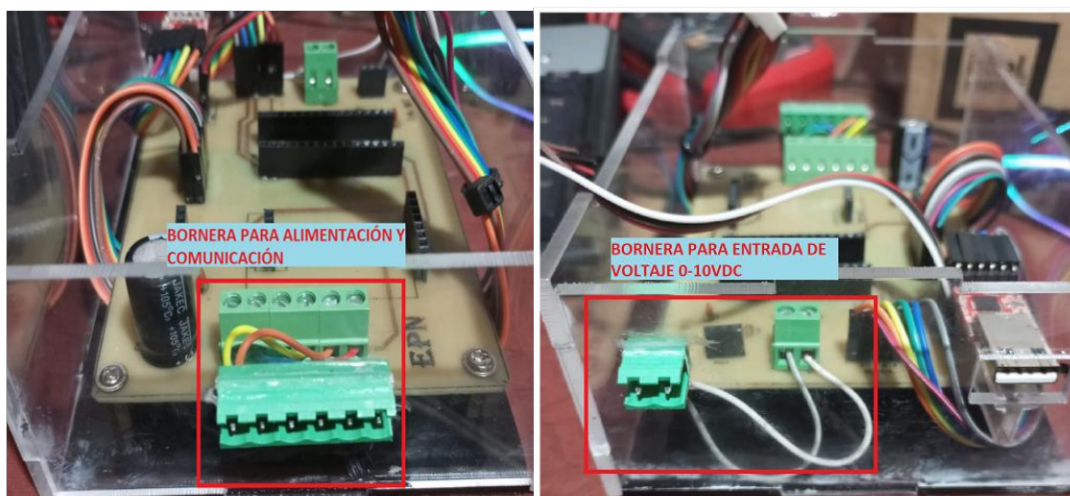


Figura 4.2. Extensión para borneras removibles

Como se puede observar en la Figura 4.3, el resultado final de la placa resulta en que en la parte inferior izquierda se encuentra las entradas de alimentación de la fuente de la red DeviceNet como los pines de comunicación CAN-H y CAN-L que van directamente al módulo MCP2515 TJA1050.

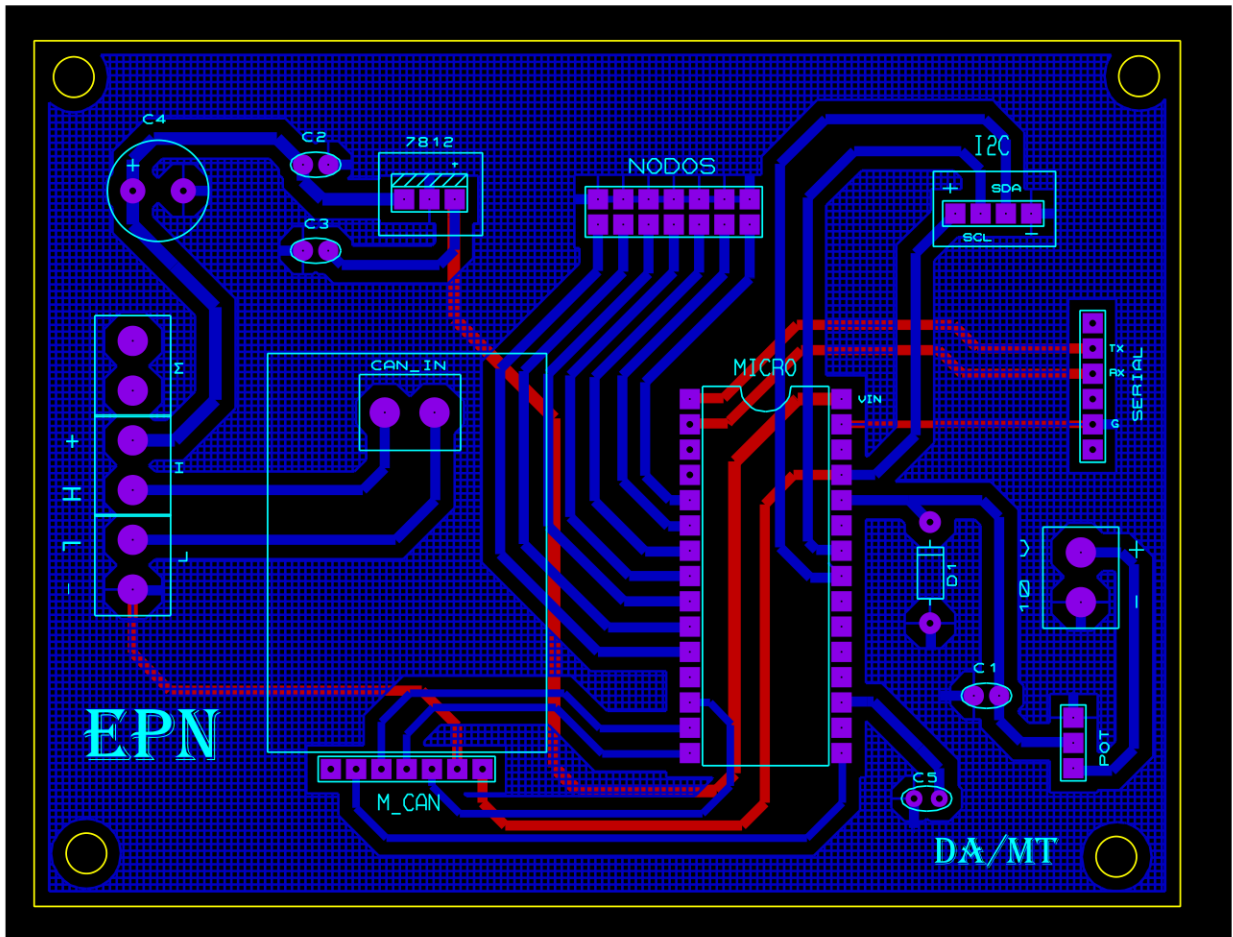


Figura 4.3. Diseño de PCB de dispositivo prototipo

Una vista general de la placa implementada acorde al diseño presentado en la Figura 4.3, se muestra en la Figura 4.4.

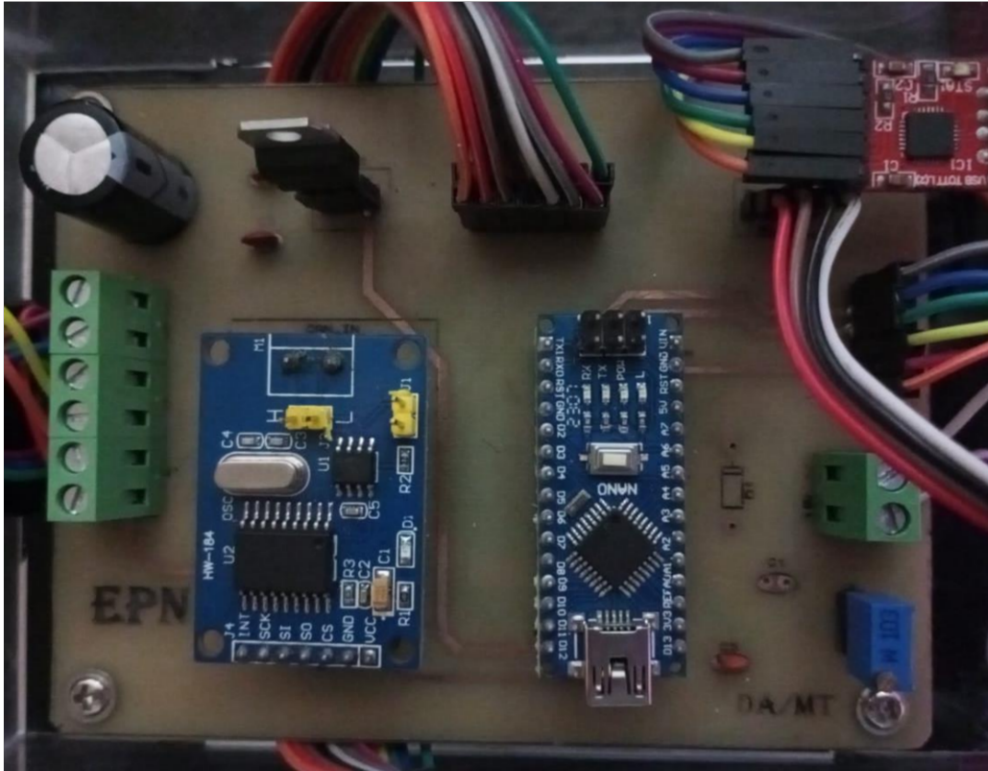


Figura 4.4. Implementación de placa PCB

Las pistas dedicadas para la comunicación entre la red DeviceNet con el módulo deben ser lo más cortas posibles para evitar el ingreso de ruido externo. De igual manera entre el módulo MCP2515 TJA1050 y los pines de comunicación SPI de la Arduino Nano se encuentran lo más ceca.



Figura 4.5. Distancias de pistas para comunicacion DeviceNet y SPI

El diseño abarca una gran masa de tierra en toda la placa con la finalidad de disminuir el ruido que perjudica en tanto en la comunicación con la red DeviceNet como entre los distintos componentes de la placa.



Figura 4.6. Masa de tierra en la PCB diseñada

ANEXO III

Guía de procedimiento de integración de prototipo de dispositivo en la red DeviceNet

Como primer paso se requiere del archivo EDS con las modificaciones necesarias para el reconocimiento del dispositivo en la red DeviceNet, el archivo EDS modificado se encuentra de manera digital.

```
[IO_Info]
  1_COSetPathNotSupported = 1;                $ no setting of connection path

  Default      = 0x0004;                      $ COS perfered
  PollInfo     = 0x0001, 1, 1;                $ Not supported
  StrobeInfo   = 0x0002, 1, 1;                $ Strobe is Input1 only, Interval counting not supported
  COSInfo      = 0x0004, 1, 1;                $ Use Input1

$tipo cos
  Input1 =
    1,
    0,
    0x0000,
    "Sensor",
    6, "20 04 24 01 30 03",
    $ Se especifica que el dispositivo admite un parámetro de entrada de un Byte
    $ Se especifica el número de bits a utilizar
    $ Especifica el tipo de datos que envia el dispositivo COS and Strobe
    $ Nombre del dispositivo
    $ Tamaño de ruta al atributo de valor de ID

$ Para el caso del sensor digital se especifica las entradas utilizadas, además de la función que desempeña cada uno de los bits enviados hacia el scanner

  "The format for the COS and Strobe input byte follows: \n"
  "DB7 Not Used \n"
  "DB6 Not Used \n"
  "DB5 Counter Output \n"
  "DB4 Motion Output \n"
  "DB3 Margin Diagnostic 3x \n"
  "DB2 Margin Diagnostic 2x \n"
  "DB1 Sensor Output \n"
  "DB0 Sensor Output \n"
  "See specific parameter for more information.";
```

Figura 4.7. Archivo EDS de prototipo de dispositivo para integrar en la red

Una vez modificado el EDS, el documento debe adjuntarse en los archivos de la aplicación RSNetWorx mediante el software EDS Hardware Installation Tool. La plataforma de RSNetWorx debe encontrarse cerrada con la finalidad de evitar que el nuevo archivo no sea reconocido.

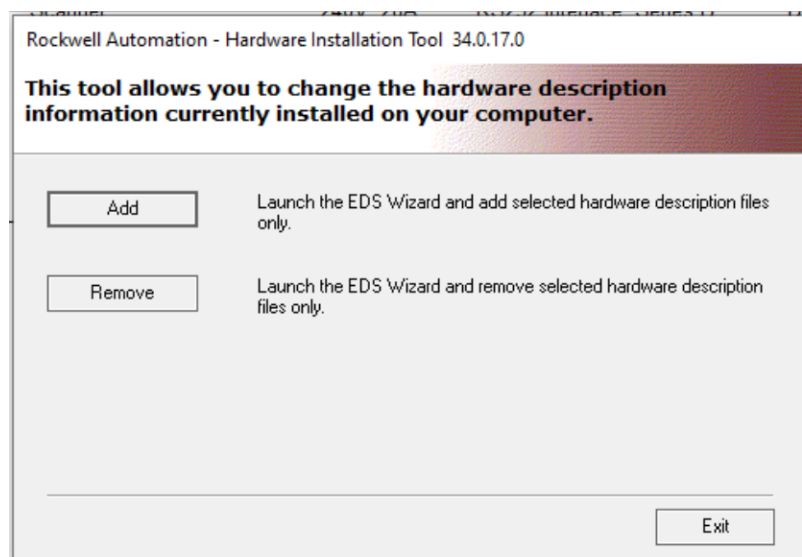


Figura 4.8. Software EDS Hardware Installation

Se elige la opción Add para añadir un nuevo archivo EDS que corresponde al dispositivo a integrar en la red DeviceNet.

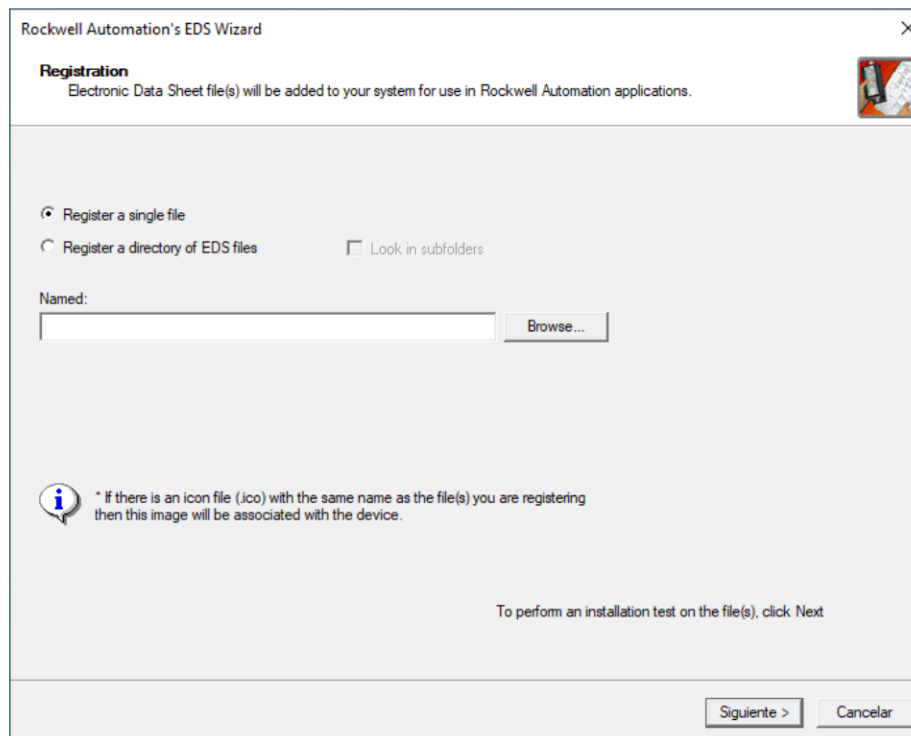


Figura 4.9. Ventana para cargar archivo EDS

Si se requiere subir toda una carpeta de archivos EDS se elige la segunda opción, para el caso del dispositivo experimental solamente se requiere de un archivo por lo tanto se escoge la primera opción y se busca la carpeta que contiene el archivo EDS.

Una vez seleccionado el archivo se carga el dispositivo, en esta ocasión se presentan advertencias, pero el archivo en general es aceptado como válido en el software RSNetWorx.

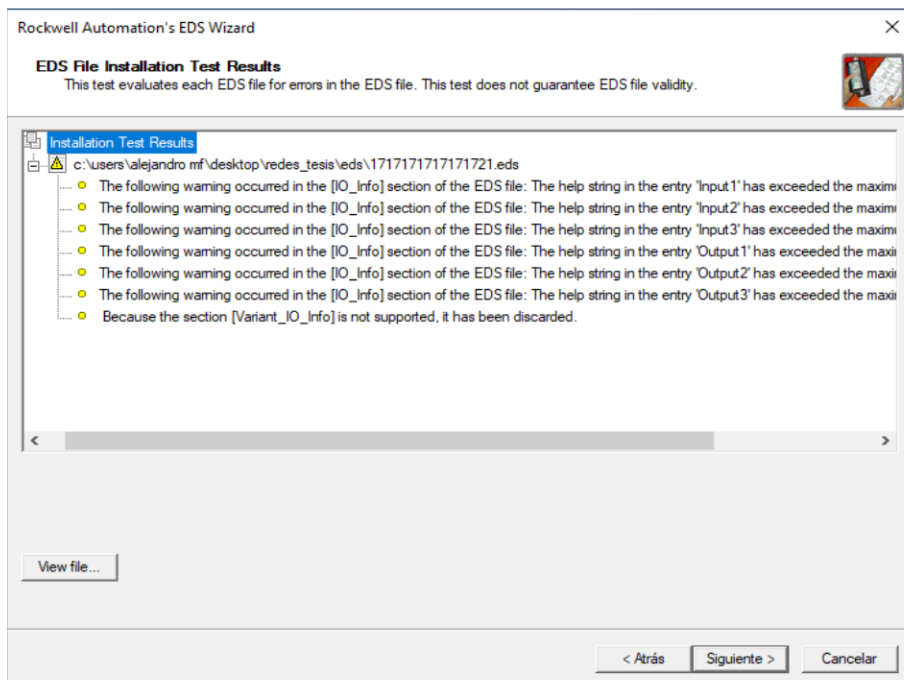


Figura 4.10. Registro de nuevo archivo EDS en los archivos de la aplicación RSNetWorx

En la siguiente ventana se verifica tanto el nombre del dispositivo con el que se encontrará en la red virtual y del ícono correspondiente de este modo se verifica que el archivo EDS corresponde al nuevo dispositivo.

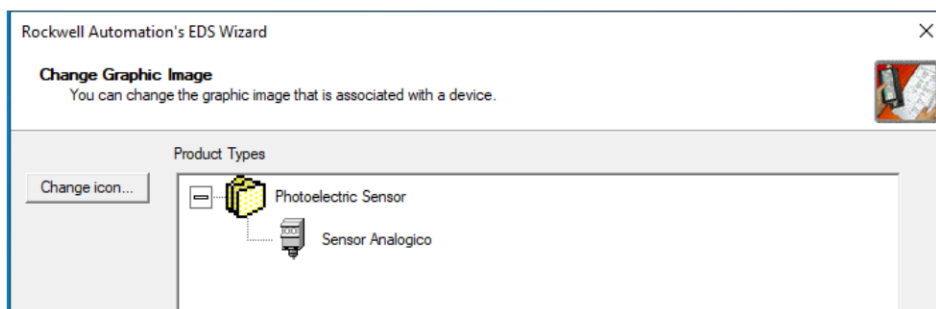


Figura 4.11. Ícono y nombre del nuevo dispositivo para la red DeviceNet

Al final si todo el proceso se ejecuta de manera correcta el archivo EDS es validado y debe salir un mensaje como se muestra en la Figura 4.12.

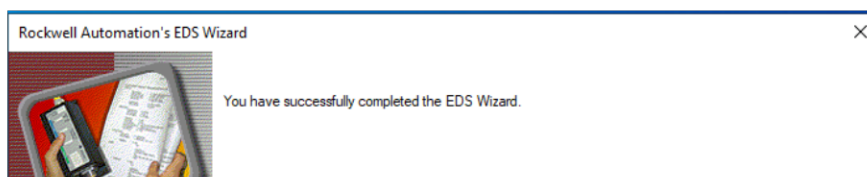


Figura 4.12. Validación de archivo EDS

Una vez instalado el archivo EDS se procede a realizar la configuración y diseño de la red DeviceNet de manera virtual en el software RSNetWorx. Se debe asignar a cada dispositivo disponible un número determinado de nodo, el valor 00 es netamente reservado para el dispositivo maestro (scanner).

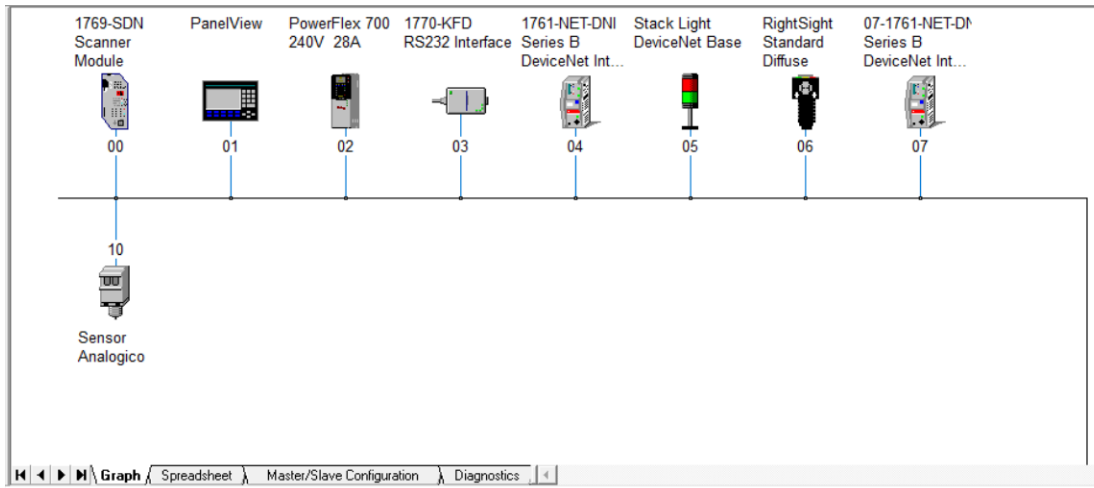


Figura 4.13. Red DeviceNet configurada

Todos los dispositivos tienen un número de nodo asignado a excepción del prototipo como se puede visualizar en la figura, por lo que se solicita se verifique su número correspondiente al momento configurar la red.

Selección de cada uno de los dispositivos para diseño de la red DeviceNet

- Selección de dispositivo maestro (scanner):

Ir a la carpeta “Communications Adapter” y buscar el modelo del scanner disponible en el laboratorio de redes industriales siendo el modelo 1769-SDN Scanner Module Rev 03.

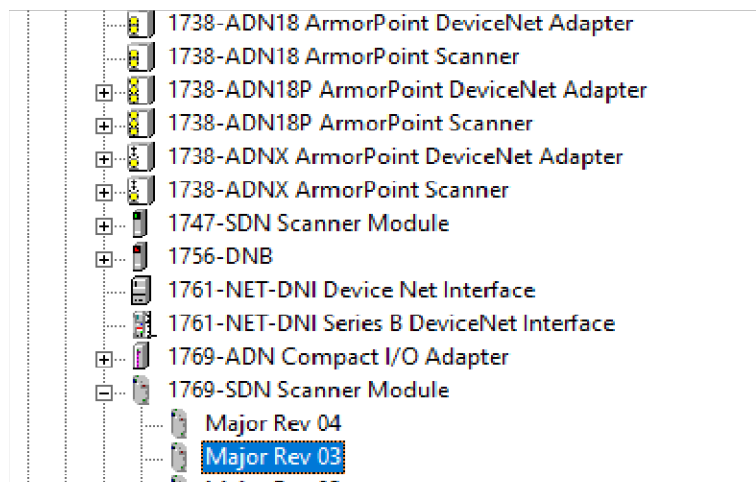


Figura 4.14. Selección de versión de scanner

Se arrastra el dispositivo a la red y por defecto se asigna un número de nodo 00, únicamente para este dispositivo se deja asignado el número de nodo por defecto ya que el scanner actúa como dispositivo maestro en la red DeviceNet.

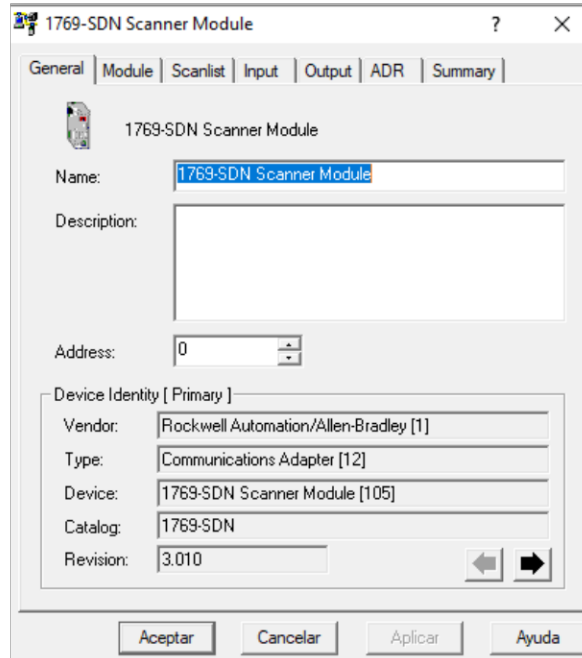


Figura 4.15. Propiedades de dispositivo maestro scanner

- Variador de velocidad:

Dirigirse a la carpeta “DPI to DeviceNet” y seleccionar el modelo del dispositivo que corresponde a “PowerFlex 700 240V 28A” debe ser el modelo que no tenga ninguna otra opción de selección de versión.

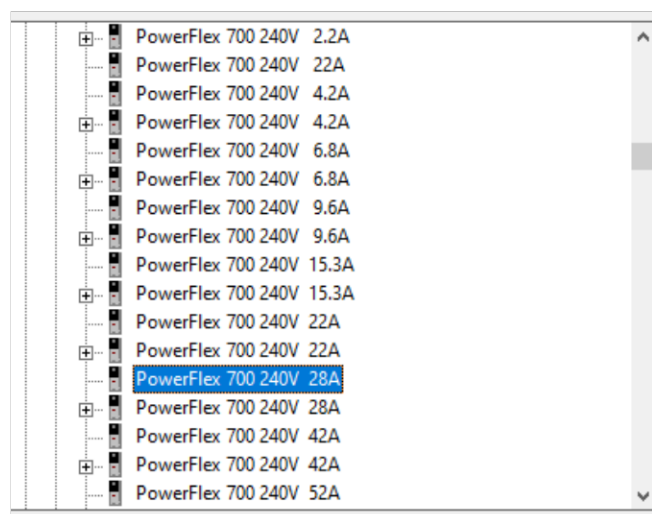


Figura 4.16. Selección de Variador de velocidad

Al arrastrar el dispositivo el número de nodo asignado por defecto es el 01, para el dispositivo disponible en la red el número debe ser 02 por lo que se requiere modificar el número en la red a diseñar mediante la opción “Address” al dar doble clic en el ícono del dispositivo.

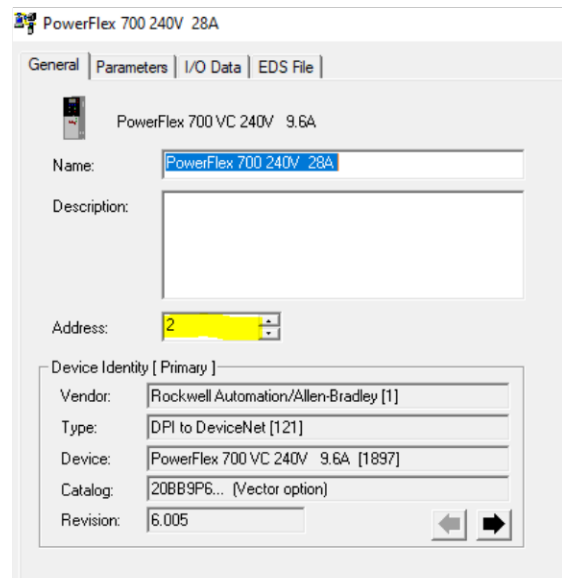


Figura 4.17. Cambio de número de nodo de variador de velocidad

- Luces indicadoras:

Dirigirse a la carpeta “General Purpose Discrete I/O” y buscar el nombre del dispositivo “Stack Light DeviceNet Base Rev 02”.

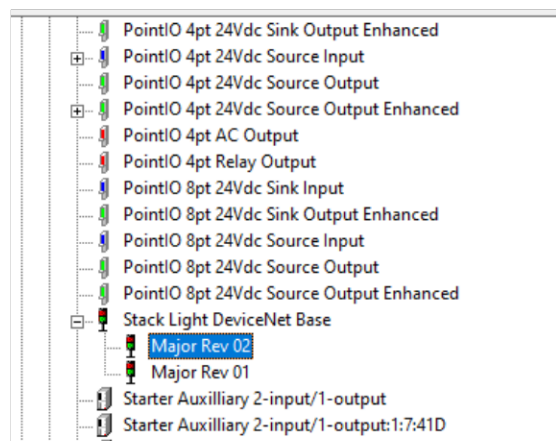


Figura 4.18. Selección de luces indicadoras

Al igual que el caso del variador se asigna un número de nodo por defecto, pero se requiere modificar al número de nodo asignado en el laboratorio de redes industriales que corresponde a 05.

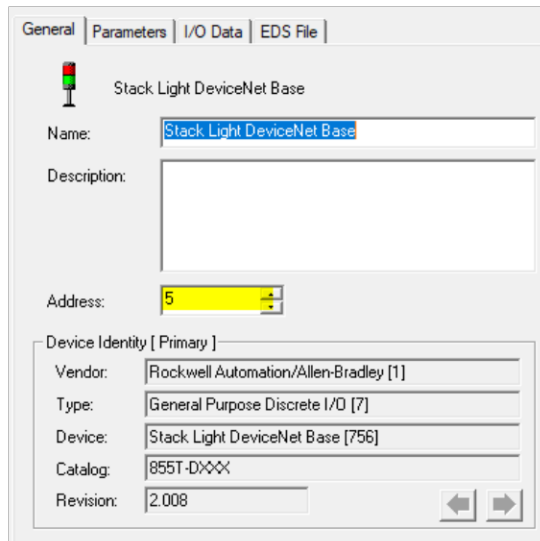


Figura 4.19. Cambio de número de nodo para luces indicadoras

- Sensor fotoeléctrico:

Dirigirse a la carpeta “Photoelectric Sensor” y seleccionar el modelo “RightSight Standard Diffuse”

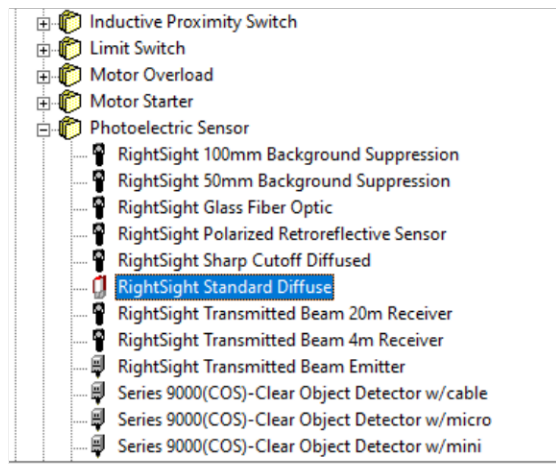


Figura 4.20. Selección de dispositivo de sensor fotoeléctrico

Se debe modificar el número de nodo que corresponde al número asignado en el laboratorio de redes industriales siendo de un valor de 06.

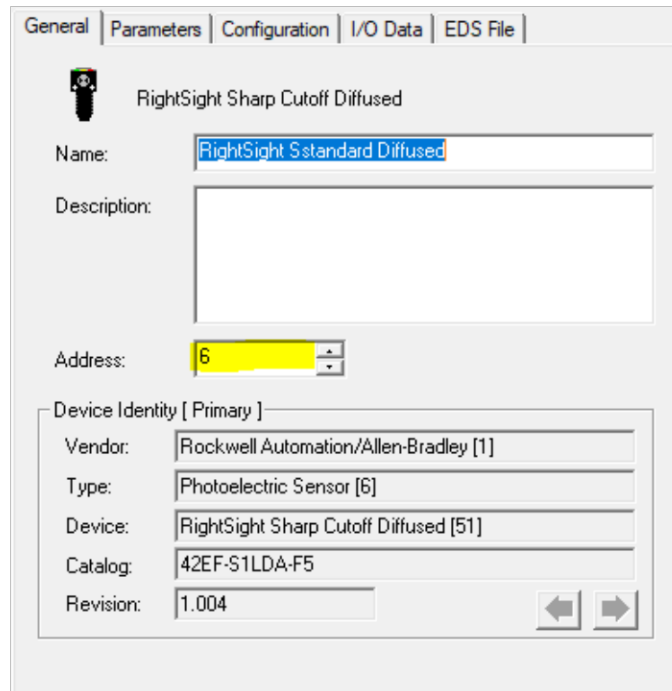


Figura 4.21. Cambio de número de nodo para sensor fotoeléctrico

- Dispositivo experimental:

El prototipo de dispositivo analógico se encuentra en la misma carpeta del sensor fotoeléctrico con el nombre “RightSight Transmitted Beam Emitter” la mayor diferencia radica en que al momento de arrastrar el dispositivo el nombre cambia al modificado y asignado en el archivo EDS que se cargó con anterioridad de manera automática.

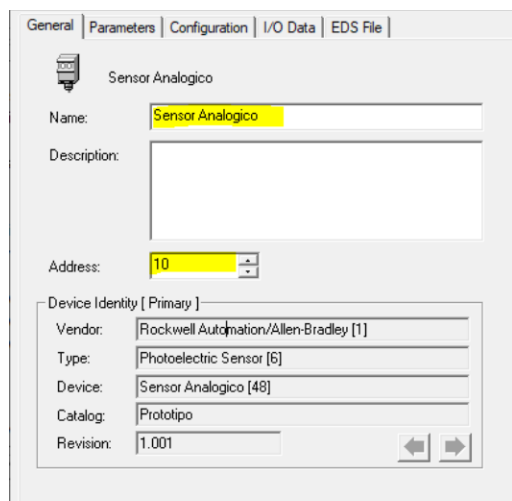


Figura 4.22. Selección de dispositivo experimental

Para el nuevo dispositivo la dirección de nodo puede ser cualquiera menos las direcciones que ya se encuentren ocupadas en la red DeviceNet, para este caso en específico se asigna la dirección de nodo 10.

Asignación de espacios de memoria en el dispositivo maestro (Scanner)

En las propiedades del scanner en la sección Scanlist se encuentran todos los dispositivos que en la sección anterior fueron agregados a la red DeviceNet, para que el scanner pueda reconocer los datos se debe asignar los dispositivos deseados a la lista del scanner tal y como se puede observar en la Figura 4.23.

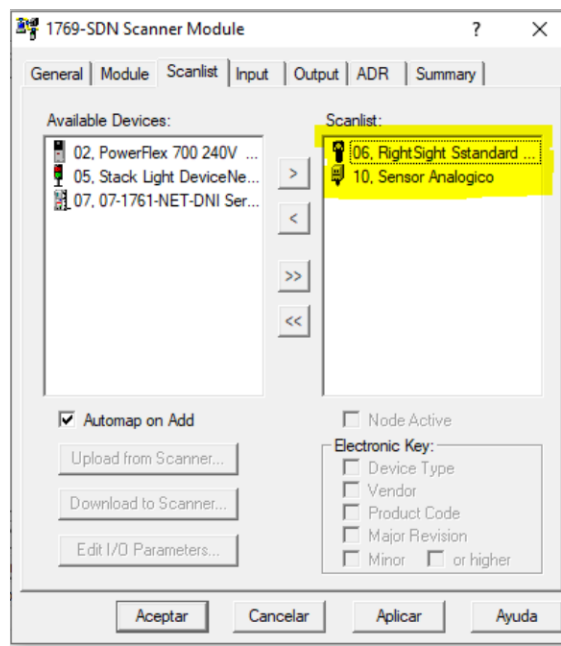


Figura 4.23. Asignación de dispositivo en la lista del scanner

Luego en la pestaña "Input" para los dos dispositivos que fueron agregados a la lista del scanner se asigna un espacio de memoria, para mayor facilidad de lectura de datos cada dispositivo cuenta con un espacio de memoria de 32 bits. Esto representa la opción menos recomendada debido a que se hace uso de mayor recurso de espacio de almacenamiento y que a su vez representa un mal uso de la memoria del scanner.

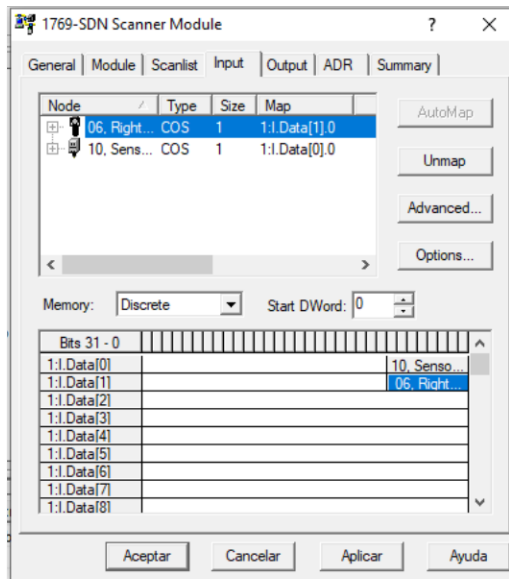


Figura 4.24. Asignación de espacio de memoria para datos de entrada hacia el dispositivo maestro

Los espacios de memoria se asignan de manera automática, pero mediante la opción “Advanced Mapping” se puede ubicar de manera manual los espacios de memoria que se desea utilizar y desde que número de bit.

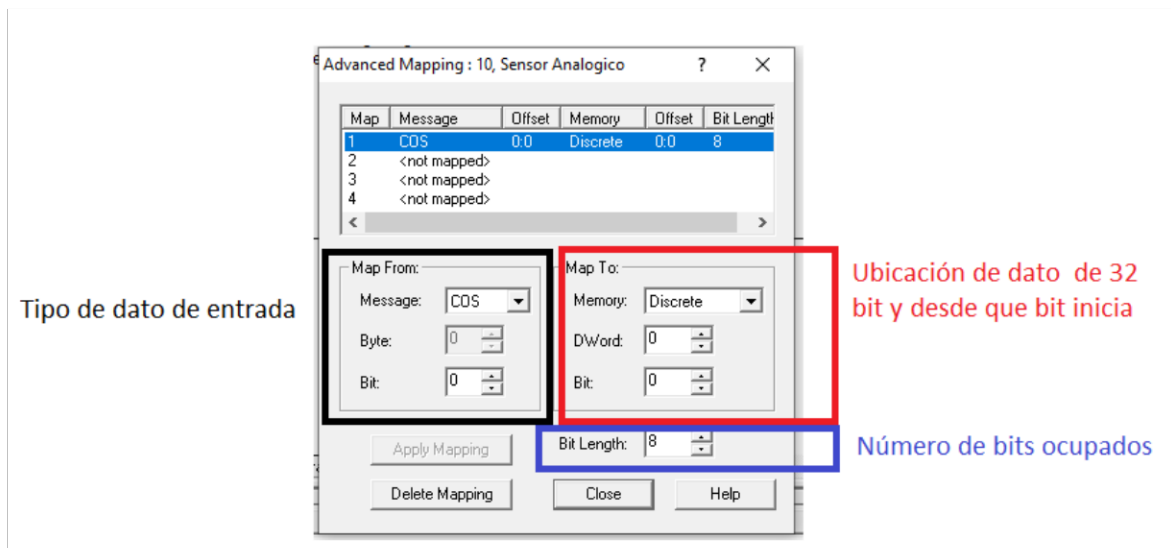


Figura 4.25. Asignación de espacio de memoria de manera manual

Una vez que los espacios de memoria son asignados se guarda todos los cambios y se inicia la red poniendo en línea.

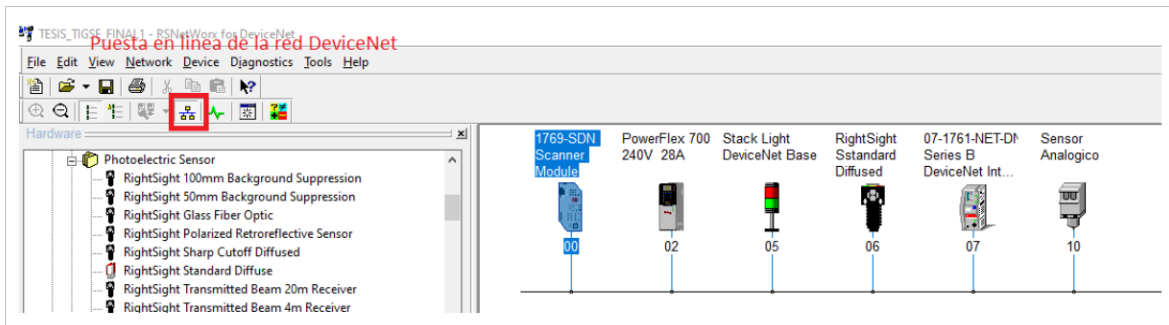


Figura 4.26. Puesta en línea de la red DeviceNet diseñada

Si es primera vez que se pone en línea la red se debe seleccionar la dirección de PLC correspondiente y seleccionar el módulo del scanner y finalmente seleccionar la red tipo DeviceNet.

La nueva red de ser cargada al scanner caso contrario no se detectará ningún cambio o modificación realizada, puesto que el scanner debe tener toda la configuración realizada de la red para aceptar como valido los datos que son enviados y recibidos en la red DeviceNet.

Lectura de datos de prototipo de sensor analógico en PLC CompactLogix

Es necesario crear un nuevo programa en el software RSLogix-5000, se debe seleccionar el modelo del PLC disponible en el laboratorio de Redes Industriales como se visualiza en la figura

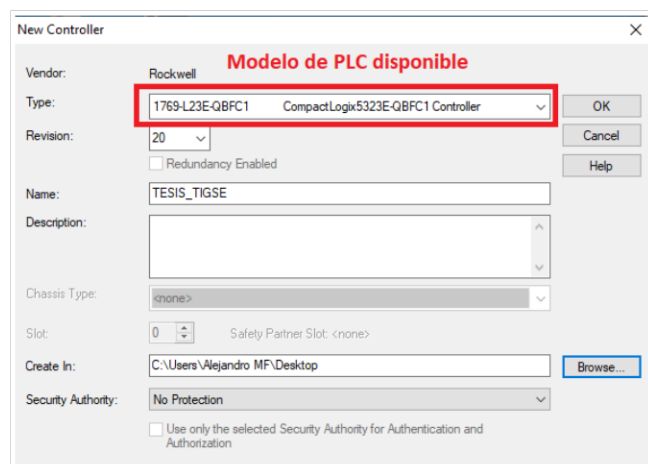


Figura 4.27. Selección de modelo de PLC

En la parte inferior izquierda se cuenta con los 4 slot que corresponden a las entradas y salidas digitales, entradas y salidas analógicas y un contador, es necesario añadir un slot que contenga al módulo del scanner.

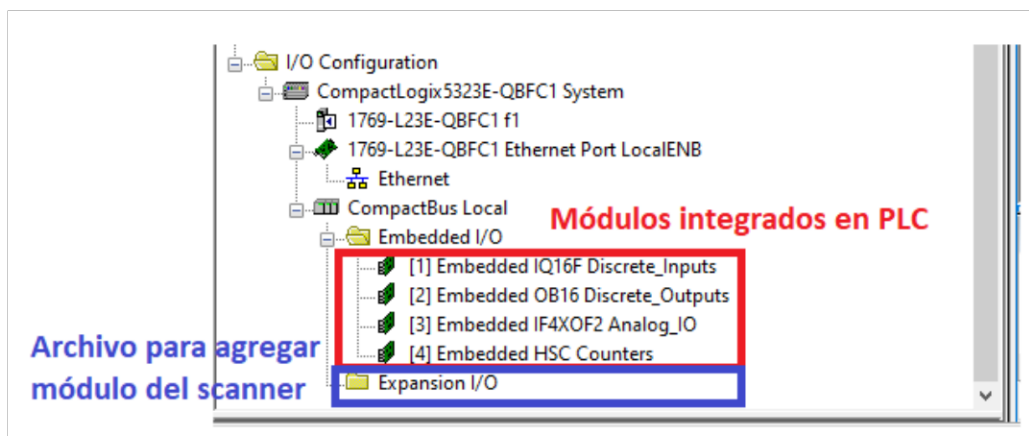


Figura 4.28. Módulos integrados de entradas y salidas digitales y analógicas

En la carpeta “Expansion I/O” es donde se debe añadir el módulo del scanner para tener la comunicación con la red DeviceNet y recibir los datos de los dispositivos configurados.

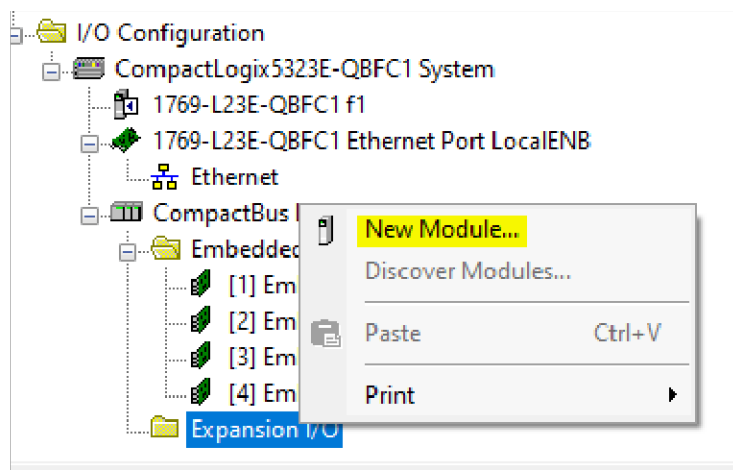


Figura 4.29. Expansión para Scanner para red DeviceNet

El modelo del scanner debe ser el mismo disponible en la red física, el número de revisión para este caso es de manera automática y corresponde al número 3.

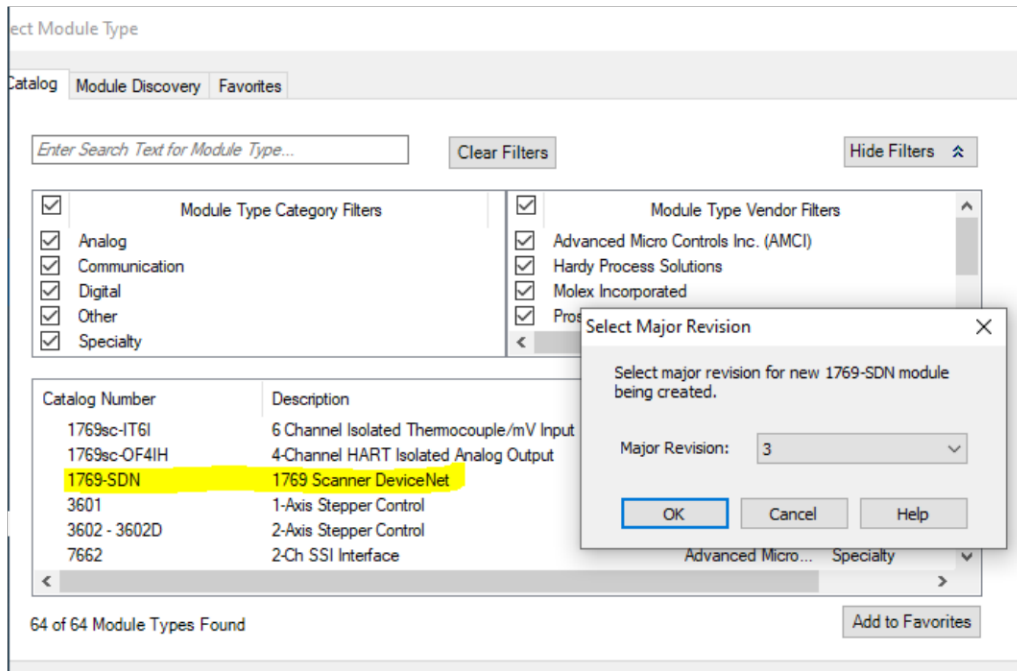


Figura 4.30. Selección de Scanner para Red DeviceNet

Al visualizar la Figura 4.30, la ventana de configuración permite modificar el número de los espacios de memoria del scanner, número de bits de entrada y de salida. Además, se tiene en cuenta que contando desde el lado izquierdo el número de slot del módulo del scanner es el 5.

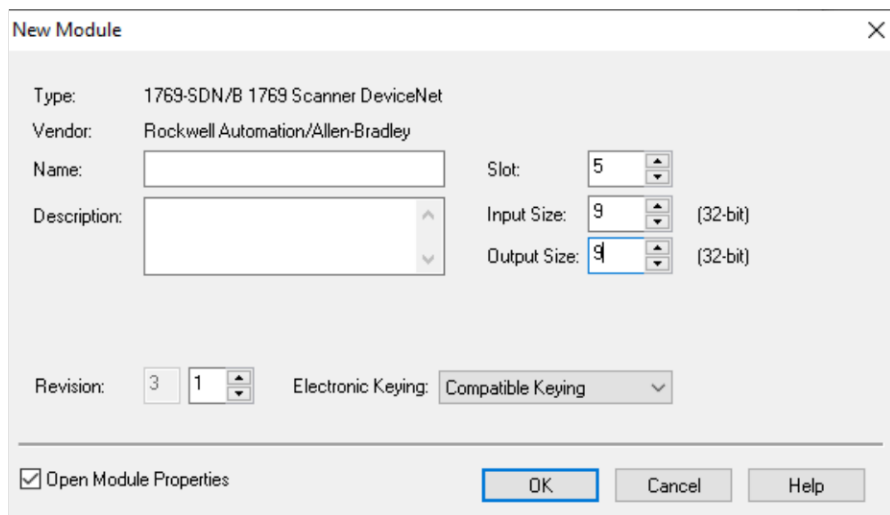


Figura 4.31. Configuración de espacios de memoria y bits de entrada y salida del scanner

Para tener una comunicación entre el PLC y la red DeviceNet el scanner sirve como intermediario ordenando los datos tanto para el envío o recepción, además se debe añadir

la red modificada al módulo del scanner para asegurar la comunicación y transmisión de datos entre el PLC y los dispositivos que se encuentran en la red.

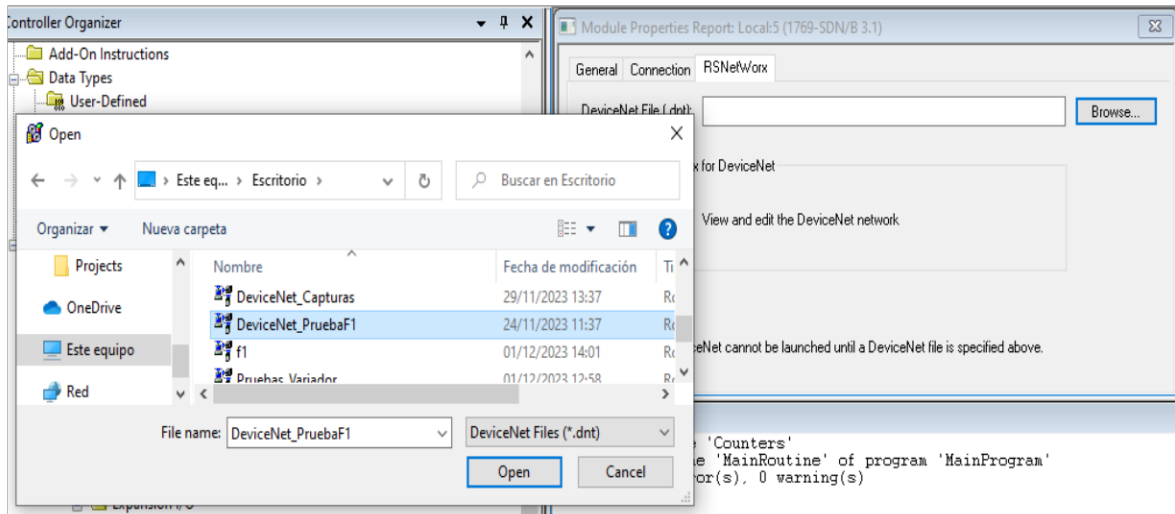


Figura 4.32. Selección de red DeviceNet diseñada

Los programas correspondientes a la lectura de cada uno de los datos que circulan por la red se lo hace mediante los registros del módulo de expansión tanto para la entrada y salida de los datos provenientes de los dispositivos esclavos.

+ Local:1:C	{...}	{...}	AB:Embedded_I...
+ Local:1:I	{...}	{...}	AB:Embedded_I...
+ Local:2:C	{...}	{...}	AB:Embedded_O...
+ Local:2:I	{...}	{...}	AB:Embedded_O...
+ Local:2:O	{...}	{...}	AB:Embedded_O...
+ Local:3:C	{...}	{...}	AB:Embedded_IF...
+ Local:3:I	{...}	{...}	AB:Embedded_IF...
+ Local:3:O	{...}	{...}	AB:Embedded_IF...
+ Local:4:C	{...}	{...}	AB:Embedded_H...
+ Local:4:I	{...}	{...}	AB:Embedded_H...
+ Local:4:O	{...}	{...}	AB:Embedded H...
+ Local:5:I	{...}	{...}	AB:1769_SDN_1...
+ Local:5:O	{...}	{...}	AB:1769_SDN_4...

Localidad de memoria del módulo del scanner

Figura 4.33. Registro para entrada y salidas de datos para red DeviceNet

Por último, al subir el programa al PLC el scanner no se encuentra en operación, para poner en marcha se debe acceder a un registro desde la localidad 5 del PLC configurada como salida para realizar un envío de dato hacia el scanner y cambiar a modo run.

+	Local1:C	{...}	{...}		A
+	Local1:I	{...}	{...}		A
+	Local2:C	{...}	{...}		A
+	Local2:I	{...}	{...}		A
+	Local2:O	{...}	{...}		A
+	Local3:C	{...}	{...}		A
+	Local3:I	{...}	{...}		A
+	Local3:O	{...}	{...}		A
+	Local4:C	{...}	{...}		A
+	Local4:I	{...}	{...}		A
+	Local4:O	{...}	{...}		A
+	Local5:I	{...}	{...}		A
-	Local5:O				A
-	Local5:O.CommandRegister				A
	Local5:O.CommandRegister.Run	1		Decimal	B
	Local5:O.CommandRegister.Fault	0		Decimal	B
	Local5:O.CommandRegister.DisableNetwork	0		Decimal	B
	Local5:O.CommandRegister.HaltScanner	0		Decimal	B
	Local5:O.CommandRegister.Reset	0		Decimal	B
+	Local5:O.Data	{...}	{...}	Decimal	C

Registro para puesta en marcha del scanner

Figura 4.34. Puesta en marcha del Scanner

En la localidad 5.0 Data es donde se va a encontrar todos los datos que ingresan al scanner desde la red DeviceNet. Para el caso propuesto los datos del dispositivo experimental se encuentran en el espacio de memoria "Local 5.0 Data [0]".

Localidad asignado a datos del sensor analógico					
-	Local:5:0.Data	{...}	{...}	Decimal	DINT[9]
+	Local:5:0.Data[0]	0		Decimal	DINT
+	Local:5:0.Data[1]	0		Decimal	DINT
+	Local:5:0.Data[2]	0		Decimal	DINT
+	Local:5:0.Data[3]	0		Decimal	DINT
+	Local:5:0.Data[4]	0		Decimal	DINT
+	Local:5:0.Data[5]	0		Decimal	DINT
+	Local:5:0.Data[6]	0		Decimal	DINT
+	Local:5:0.Data[7]	0		Decimal	DINT
+	Local:5:0.Data[8]	0		Decimal	DINT

Figura 4.35. Espacio de memoria para datos del dispositivo experimental

El espacio de memoria asignado en el PLC debe coincidir con el espacio configurado en la red DeviceNet de la figura.

Las variables que se crean pueden ser asignadas a un bit en específico de la localidad de memoria del scanner o toda la palabra de 32 bits en conjunto como se puede observar en la Figura 4.36.

Variable asignada a dato de 1 bit					
Prueba2	Local:5:I.Data[1]...	Local:5:I.Data[1]...	BOOL		Read/Write
+ STATUS_SE...	Local:5:I.Data[0](C)	Local:5:I.Data[0](C)	DINT		Read/Write

Variable asignada a dato de 32 bits

Figura 4.36. Datos asignados a variables para la lectura de voltaje analógico del dispositivo experimental

Los datos enviados desde el sensor analógico por medio de la red DeviceNet son en formato de bit, por lo que se requiere de un proceso de conversión de datos para expresar en valor decimal el voltaje censado por el dispositivo y que debe encontrarse en un rango de 0-10 VDC.

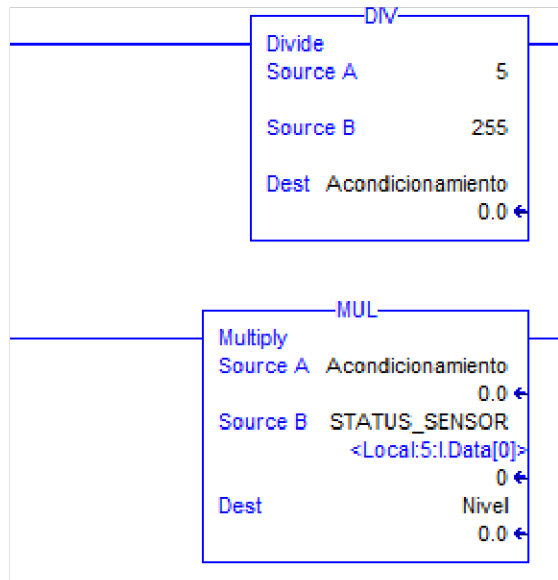


Figura 4.37. Sección de conversión de datos digitales a valor de voltaje censado

ANEXO IV

Descripción de procedimiento ante falla de proceso de comunicación del dispositivo.

En la siguiente tabla se detalla los errores más comunes que se puede presentar al momento de integrar el sensor analógico a la red DeviceNet.

Tabla 4.2. Descripción de errores y soluciones al integrar el nuevo dispositivo a la red DeviceNet

Error-Nodo	Descripción	Solución
80-00	Módulo scanner se encuentra inactivo	Poner en marcha el módulo scanner desde los registros de localidad en el programa del PLC como en la figura
75-10	El dispositivo maestro no recibe ningún mensaje del esclavo trascurrido 10s	Revisar si el dispositivo esclavo se encuentra encendido, y con el número de nodo correcto al configurado en la red DeviceNet. Caso contrario colocar el número de nodo correcto y realizar nuevamente el proceso de comunicación.
78-10	El dispositivo esclavo no existe en la red DeviceNet configurada	Revisar el número de nodo del dispositivo y poner en modo de comunicación nuevamente al dispositivo.
84-10	Error durante el proceso de establecimiento de conexión con el dispositivo esclavo al presentar un tiempo de retardo de respuesta por parte del dispositivo esclavo.	Colocar el dispositivo en modo de lectura de número de nodo, en siga colocar nuevamente al dispositivo en modo de establecimiento de comunicación. Otra posible solución es cargar nuevamente la red DeviceNet en el scanner

ANEXO V

Programa de prototipo de dispositivo en tarjeta de desarrollo Arduino Nano

La programación de la tarjeta de desarrollo Arduino Nano se encuentra de manera digital.