



**UNIVERSIDAD POLITÉCNICA SALESIANA
SEDE CUENCA**

CARRERA DE ELECTRÓNICA Y AUTOMATIZACIÓN

DESARROLLO DE UN SISTEMA DE MONITOREO Y CONTROL REMOTO PARA
UN VARIADOR DE FRECUENCIA DELTA INTEGRADO A UN PLC S7-1200 DE
SIEMENS POR MEDIO DE UNA PASARELA INTELIGENTE BASADO EN
RASPBERRY PI

Trabajo de titulación previo a la obtención del
título de Ingeniero en Electrónica

AUTORES: WILLIAM ALEXIS FERNÁNDEZ LOJANO

DIEGO FERNANDO IDROVO VAZCONEZ

TUTOR: ING. JULIO CESAR ZAMBRANO ABAD, PhD.

Cuenca - Ecuador

2024

**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE
TITULACIÓN**

Nosotros, William Alexis Fernández Lojano con documento de identificación N° 0107789265 y Diego Fernando Idrovo Vazconez con documento de identificación N° 0105667570; manifestamos que:

Somos los autores y responsables del presente trabajo; y, autorizamos a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Cuenca, 10 de febrero del 2024

Atentamente,



William Alexis Fernández Lojano

0107789265



Diego Fernando Idrovo Vazconez

0105667570

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Nosotros, William Alexis Fernández Lojano con documento de identificación N° 0107789265 y Diego Fernando Idrovo Vazconez con documento de identificación N° 0105667570, expresamos nuestra voluntad y por medio del presente documento cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del Proyecto técnico: “Desarrollo de un sistema de monitoreo y control remoto para un variador de frecuencia Delta integrado a un PLC S7-1200 de Siemens por medio de una pasarela Inteligente basado en Raspberry Pi”, el cual ha sido desarrollado para optar por el título de: Ingeniero en Electrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 10 de febrero del 2024

Atentamente,



William Alexis Fernández Lojano

0107789265



Diego Fernando Idrovo Vazconez

0105667570

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Julio Cesar Zambrano Abad con documento de identificación N° 0301489696, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DESARROLLO DE UN SISTEMA DE MONITOREO Y CONTROL REMOTO PARA UN VARIADOR DE FRECUENCIA DELTA INTEGRADO A UN PLC S7-1200 DE SIEMENS POR MEDIO DE UNA PASARELA INTELIGENTE BASADO EN RASPBERRY PI, realizado por William Alexis Fernández Lojano con documento de identificación N° 0107789265 y Diego Fernando Idrovo Vazconez con documento de identificación N° 0105667570, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 10 de febrero del 2024

Atentamente,

**JULIO CESAR
ZAMBRANO
ABAD**

Firmado digitalmente por JULIO CESAR
ZAMBRANO ABAD
DN: cn=JULIO CESAR ZAMBRANO ABAD,
o=EC, o=BANCO CENTRAL DEL ECUADOR,
ou=ENTIDAD DE CERTIFICACION DE
INFORMACION-ECIBCE,
email=juliozambranoabad@gmail.com
Fecha: 2024.03.18 17:44:00 -05'00'

Ing. Julio Cesar Zambrano Abad, PhD.

0301489696

AGRADECIMIENTOS

Agradecimientos de William Alexis Fernandez Lojano

En el umbral de esta travesía académica, debo honrar a mis queridos padres, cuyo apoyo inquebrantable ha sido el pilar que ha sostenido mis esfuerzos en este arduo sendero hacia la consecución de este noble propósito. A todos aquellos maestros cuya sabiduría ha iluminado mi camino, les debo una deuda de gratitud infinita, pues sus enseñanzas han sido el fundamento sobre el cual se erige este logro. Y a mí mismo, fiel compañero de esta jornada, me enorgullezco por mi perseverancia incansable ante la adversidad, por nunca claudicar ante los embates del destino y por enfrentar cada desafío con temple y coraje.

Agradecimientos de Diego Fernando Idrovo Vazconez

Después de este emocionante recorrido, quiero expresar mi más sincero agradecimiento a mis padres y familia, cuyo apoyo inquebrantable ha sido mi mayor fortaleza. Agradezco también a la universidad y a mis profesores por su dedicación y sabiduría compartida, que han sido fundamentales en mi formación académica. Y no puedo olvidar reconocerme a mí mismo por superar cada desafío con trabajo arduo y dedicación.

DEDICATORIAS

Dedico este proyecto a todas las personas que han creído en mí y que, a lo largo de los años, han compartido su conocimiento conmigo, ayudándome a alcanzar este punto. Vuestra confianza y colaboración han sido fundamentales en este camino.

Por Diego Fernando Idrovo Vazconez

A aquellos que en los pasillos de la universidad tejieron conmigo el tapiz de la vida, en las alegrías y las penas, en las victorias y derrotas, les dedico con profunda admiración este pequeño tributo. Vuestra amistad ha sido el sustento que ha alimentado mi alma durante esta travesía académica. Que esta tesis sea el eco de nuestra unión indeleble.

Por William Alexis Fernandez Lojano

Índice general

Agradecimientos	I
Dedicatorias	II
Índice General	III
Índice de figuras	VII
Índice de tablas	VIII
Resumen	IX
Abstract	X
Antecedentes	1
Justificación	4
Objetivos	7
Introducción	8
1. Comunicación Modbus y VFD	9
1.1. Comunicación Modbus RTU.....	9
1.1.1. Que es Modbus RTU ?	9
1.1.2. Comunicación bajo el estándar RS485	12
1.1.3. Implementación en S7-1200	15
1.2. Variador de frecuencia (VFD)	19
<i>ÍNDICE GENERAL</i>	IV

1.3. Integración VFD y PLC mediante Modbus RTU	22
2. Pasarela inteligente	29
2.1. Pasarela inteligente de Siemens.....	30
2.2. Raspberry pi como pasarela inteligente.....	31
2.2.1. Node-RED en Raspberry pi.....	32
2.3. Plataforma IoT.....	36
2.3.1. Elementos de una infraestructura IoT	37
2.4. Integración de Raspberry Pi y el PLC.....	39
2.4.1. Arquitectura de la comunicación entre el PLC y el Raspberry Pi .	39
2.4.2. Configuración de la Comunicación Bidireccional en el PLC.....	39
3. Implementación del sistema de monitoreo y control remoto	45
3.1. Arquitectura de del sistema de monitoreo y control remoto.....	45
3.2. Configuración y programación del dashboard de Ubidots.....	46
3.3. Envío y recepción de datos en Node-Red.....	49
3.4. Pruebas y resultados	57
4. Conclusiones y Trabajos Futuros	69
Glosario	72
Referencias	76

Índice de figuras

1.1. Ciclo de mensajes en el campo de mensajes de modbus RTU	11
1.2. Conexion RS-485	13
1.3. PLC S71200	15
1.4. Módulo de Comunicación CM1241.....	16
1.5. Configuración del Módulo	17
1.6. Bloque Modbus.....	17
1.7. Data Response	18
1.8. Diagrama eléctrico del VSI [13]	20
1.9. Tabla direcciones 1.....	23
1.10. Tabla direcciones 2.....	24
1.11. Distribución De Lectura y Escritura.....	25
1.12. Bloque de carga de la comunicación	25
1.13. Bloque para establecer frecuencia.....	25
1.14. Comando para el control del motor.....	26
1.15. Comando para la lectura de datos.....	27
1.16. Datos Obtenidos Del Variador	27
1.17. Bloque de envió de datos.....	28
2.1. Editor de Node-RED	33
2.2. Boton nodos.....	35
2.3. Arquitectura de comunicación entre el PLC y el Raspberry Pi	39
2.4. Nodos Programados para el intercambio de datos entre PLC y la nube .	40
2.5. Nodo Inject.....	41
2.6. Nodo Change	41

2.7. Nodo Function	42
2.8. Nodo Para enviar datos a Ubidots.....	42
2.9. Nodo Para leer datos de Ubidots	42
2.10. Nodo para extraer datos del PLC.....	42
2.11. Nodo para escribir en el PLC.....	43
2.12. Bloque S7 Comms	43
2.14. Array De datos	44
2.13. Bloques Put y Get del bloque S7-Comms	44
3.1. Arquitectura de del sistema de monitoreo y control remoto.....	46
3.2. PLataforma de inicio de Ubidots.....	47
3.3. Devices Ubidots.....	47
3.4. Devices Ubidots.....	48
3.5. Dashboards	48
3.6. Widgets.....	49
3.7. S7 Node	49
3.8. Node-Ubidtos	50
3.9. PLC Node	50
3.10. Ip PLC.....	51
3.11. Configuración Ip del nodo	51
3.12. Datos PLC S7	52
3.13. Nodos PLC S7	52
3.14. Conversion de datos.....	52
3.15. Variables en Ubidots	53
3.16. Variables para recibir datos desde Ubidots.....	53
3.17. Nodos Ubidtos	54
3.18. Boton nodos	54
3.19. Variables en Ubidots.....	55
3.20. Nodo Ubidots In.....	55
3.21. Nodos PLC.....	56
3.22. Nodos Función.....	56
3.23. Dashboard Ubidots.....	57

3.24. Datos Node-RED	58
3.25. RUN VFD	59
3.26. RUN en Ubidots.....	59
3.27. STOP VFD.....	60
3.28. STOP en Ubidots	61
3.29. Motor puesto a prueba	61
3.30. Ubidots giro horario.....	62
3.31. VFD giro horario.....	62
3.32. Ubidots giro Antihorario	63
3.33. VFD giro Antihorario	63
3.34. Ubidots Frecuencia 6Hz.....	64
3.35. VDF Frecuencia 6Hz.....	64
3.36. Ubidots Frecuencia 24Hz.....	65
3.37. VFD Frecuencia 24Hz	65
3.38. Ubidots Frecuencia 60Hz.....	66
3.39. VFD Frecuencia 60Hz	66
3.40. Frecuencia de salida.....	67
3.41. Lectura 720 RPM.....	68
3.42. Lectura 1440 RPM	68
3.43. Lectura 2280 RPM	68

Índice de tablas

1.1. Comparación entre RS-232 y RS-485	15
1.2. Parámetros a Configurar en el Variador de Frecuencia	22
1.3. Motor Control Commands	23
1.4. Address Calculations.....	23
1.5. Parámetros eléctricos y sus direcciones	24
2.1. Comparación entre IoT2020 y IoT2040.....	30
2.2. Comparación de modelos de Raspberry Pi.....	31
2.3. Funcionalidades de las plataformas.....	38

Resumen

El objetivo principal de este proyecto es integrar un variador de frecuencia de la marca Delta con un controlador lógico programable (PLC) de la serie Siemens S7-1200. El propósito fundamental es establecer un sistema eficiente de supervisión y gestión para el variador de frecuencia mencionado, implementando una pasarela inteligente basada en Raspberry Pi.

Este proyecto se enfoca en la posibilidad de poner en línea el PLC mediante Raspberry Pi, con el fin de optimizar costos al prescindir de complementos exclusivos de la misma marca del PLC. La elección de Raspberry Pi no solo permite ahorros significativos, sino que también habilita la conexión del PLC a internet. El control y monitoreo de los variadores de frecuencia se logra a través de la plataforma web Ubidots.

La integración de Raspberry Pi como una pasarela inteligente simplifica el acceso remoto desde cualquier dispositivo con conexión a la red. Esta elección estratégica no solo optimiza costos sino que también proporciona flexibilidad y accesibilidad. La plataforma web Ubidots se presenta como un medio eficiente para llevar a cabo el control y monitoreo de los variadores de frecuencia, permitiendo el acceso desde cualquier ubicación con conexión a internet, cumpliendo así con los requisitos necesarios para gestionar los variadores de frecuencia de manera remota y efectiva.

Palabras clave: PLC; UBIDOTS; RASPBERRY PI; VARIADOR DE FRECUENCIA

Abstract

The main purpose of this project is the integration of a Delta variable frequency drive with a Siemens S7-1200 series programmable logic controller (PLC). The main purpose is to establish an effective monitoring and management system for the mentioned frequency inverter, implementing an intelligent gateway based on Raspberry Pi.

This project focused on the possibility of bringing the PLC online through Raspberry Pi, in order to optimize costs by dispensing with exclusive complements of the same brand of PLC. The choice of Raspberry Pi not only allows significant savings, but also enables the connection of the PLC to the internet. The control and monitoring of the variable frequency drives is achieved through the Ubidots web platform.

The integration of Raspberry Pi as an intelligent gateway simplifies remote access from any device with a network connection. This strategic choice not only optimizes costs but also provides flexibility and accessibility. The Ubidots web platform is presented as an efficient means to carry out the control and monitoring of variable frequency drives, allowing access from any location with internet connection, thus fulfilling the requirements needed to manage variable frequency drives remotely and effectively.

Keywords: PLC;UBIDOTS; RASPBERRY PI; FREQUENCY CONVERTER

Antecedentes

La automatización industrial es un campo que ha experimentado un desarrollo constante desde sus inicios durante la Revolución Industrial. El continuo avance tecnológico ha llevado a las empresas a adoptar sistemas automatizados para optimizar sus procesos de producción. En este contexto, el control de motores industriales ha desempeñado un papel fundamental, y la introducción de sistemas basados en Controladores Lógicos Programables del ingles (PLC) ha marcado un punto de inflexión significativo en la automatización industrial.

Los Controladores Lógicos Programables (PLC) han surgido como componentes esenciales en una amplia gama de sistemas de control en la industria, desempeñando un papel fundamental en el ámbito de la automatización [1]. Su posición central en el panorama de los sistemas de control industrial es testimonio de su versatilidad y eficacia en diversas aplicaciones.

En la industria, encontramos una variedad impresionante de PLC, incluyendo PLC de marcas reconocidas como Allen Bradley, Siemens , Mitsubishi, Schneider, etc. Estos conjuntos de herramientas ponen a disposición una extensa variedad de aplicaciones y funcionalidades particulares, a la vez que ostentan propiedades singulares que los hacen sobresalir. Entre estas cualidades sobresalen su flexibilidad, escalabilidad y facilidad de programación.

Sin embargo, un aspecto esencial que contribuye a su preeminencia en el campo de la automatización industrial es su diversidad de protocolos de comunicación compatibles. Estos protocolos incluyen Modbus RTU, Ethernet, Profibus, Profinet, entre otros. Esta capacidad de adaptación y comunicación de los PLC ha permitido su integración efectiva en una variedad de entornos industriales, lo que resulta en un mayor control y eficiencia en los procesos industriales.

En el presente, una considerable porción de la población del planeta goza de la posibilidad de acceder a la internet, lo que ha revolucionado notablemente la manera en que interactuamos con sistemas y datos. En el ámbito industrial, la capacidad de controlar y monitorear equipos a través de la web se ha convertido en una posibilidad esencial.

En un contexto donde el acceso a Internet es ampliamente disponible, numerosas empresas han identificado oportunidades significativas para aprovechar esta conectividad en múltiples aspectos de su operación. Entre estas posibilidades, destaca de manera significativa la capacidad de monitorear y controlar en tiempo real. La habilidad de llevar a cabo un monitoreo en tiempo real no solo introduce una nueva dimensión de eficiencia operativa, sino que también fomenta enfoques innovadores para la creación de transparencia energética en la industria [2].

En la era actual, se ha evidenciado un progreso ininterrumpido en las estrategias utilizadas para habilitar la conectividad de equipos industriales a la red de Internet. En este contexto, los Controladores Lógicos Programables (PLC) han desempeñado un papel activo y en constante evolución. Diversos módulos y soluciones han surgido para conectar eficazmente estos dispositivos a la red global.

Un aspecto notable es que muchas de las soluciones para habilitar la conectividad de los PLC provienen de los mismos fabricantes de PLC de renombre, como Siemens, Schneider, y otros líderes de la industria. Estos fabricantes han desarrollado módulos y tecnologías específicas que facilitan la integración de los PLC en redes de datos y su acceso a Internet. Esta tendencia es una respuesta a la creciente demanda de una mayor visibilidad y control en los procesos industriales, aprovechando las ventajas de la conectividad global.

En el contexto de la investigación de tesis, nos complace anunciar que todos los insumos y recursos necesarios para llevar a cabo este proyecto serán generosamente proporcionados por una empresa colaboradora. Esta colaboración no solo simplifica la realización de la investigación, sino que también alivia cualquier preocupación relacionada con los costos asociados. Gracias a esta asociación, el desarrollo de la tesis se ejecutará de forma eficiente sin la dependencia de los recursos universitarios ni la generación de egresos personales. Este respaldo resulta crucial para asegurar el

logro de los objetivos de la exploración y su consecuente impacto en la ampliación del conocimiento dentro del campo de estudio.

Justificación

En el contexto industrial actual, la selección precisa de un Controlador Lógico Programable o por sus siglas del inglés (PLC) constituye un elemento determinante para el desempeño y la optimización de las operaciones. La amplia variedad de PLC disponibles en el mercado presenta características específicas que los hacen idóneos para aplicaciones particulares. Por tanto, es esencial seleccionar el PLC que se adapte de manera óptima a las necesidades específicas de la empresa y del proceso industrial en cuestión.

Cada PLC tiene sus propias capacidades y funcionalidades diseñadas para satisfacer requerimientos particulares. La elección del PLC adecuado depende del uso previsto, ya que diferentes aplicaciones requieren características específicas, como velocidad de procesamiento, capacidad de E/S, protocolos de comunicación compatibles y capacidades de programación. Por lo tanto, es imperativo evaluar detenidamente las demandas particulares de la empresa y considerar cómo el PLC seleccionado se alinea con esas necesidades específicas.

Asimismo, la necesidad de conectar los PLC a la red global plantea desafíos financieros significativos. En muchos casos, la solución preferida implica la adquisición de módulos específicos de la misma marca que el PLC, diseñados para facilitar esta conectividad. Sin embargo, estos módulos suelen tener un costo considerable que puede ser justificable en proyectos de gran envergadura con altas demandas de procesamiento de datos. En tales situaciones, las empresas pueden asumir estos costos como parte integral de su inversión en tecnología para proyectos de amplio alcance.

No obstante, existen escenarios en los que la necesidad de conectividad global no requiere soluciones tan costosas. En estas circunstancias, es viable considerar

opciones alternativas que sean más accesibles y financieramente eficientes. Un ejemplo destacado es el uso de un Raspberry Pi, una opción versátil y económicamente viable que puede cumplir con éxito la función de los módulos mencionados anteriormente. El Raspberry Pi se convierte en una solución factible para conectar el PLC a Internet, lo que, en última instancia, representa un ahorro económico significativo para las empresas.

La elección de utilizar un Raspberry Pi como puente de conectividad es una estrategia inteligente para habilitar la integración de PLCs en la red de Internet. En particular, se enfoca en el PLC Siemens S7-1200 que destaca por su óptima relación calidad-precio, posicionándose como una alternativa ampliamente seleccionada en el ámbito industrial. La propuesta incluye la utilización de la comunicación Modbus RS485, una solución robusta para vincular el PLC con un variador de frecuencia de la marca Delta MS300. Lo significativo de esta estrategia reside en que no establece la limitante de que el convertidor de frecuencia deba pertenecer a la misma marca que el PLC, puesto que la comunicación Modbus RS485 facilita una interoperabilidad sin contratiempos entre dispositivos de diferentes procedencias.

Adicionalmente, la conectividad a través de Internet brinda la posibilidad de tener el PLC en línea, lo que habilita el monitoreo y control remoto desde cualquier ubicación. La elección de la plataforma Ubidots se basa en su sencillez de uso y costos asequibles en comparación con otras alternativas. Esto permite un control y monitoreo efectivos del variador de frecuencia, contribuyendo significativamente a la reducción de costos en la operación industrial.

En última instancia, esta propuesta desempeñará un papel esencial al proporcionar un recurso fundamental para futuros proyectos de integración entre PLC y variadores de frecuencia de diferentes marcas. Al documentar de manera detallada este proceso, nuestro objetivo es establecer una base sólida y comprensible que sirva como punto de partida valioso. Esta guía abordará tanto los aspectos técnicos como los desafíos comunes relacionados con la interoperabilidad de dispositivos industriales de diversas marcas. Además, al brindar una referencia completa, esperamos simplificar el camino de investigación para futuros estudiantes, ahorrándoles tiempo y esfuerzo en investigaciones repetitivas. Con este enfoque, aspiramos a fomentar un mayor avance

en la integración de tecnologías industriales, propiciando de esta manera el desarrollo sostenido de la automatización y el control de procesos en el sector industrial.

Objetivos

Objetivo General

- Desarrollar un sistema de monitoreo y control remoto para un variador de frecuencia Delta integrado a un PLC S7-1200 de Siemens por medio de una pasarela inteligente basado en Raspberry pi

Objetivos específicos:

- Establecer una red de comunicación Modbus RTU basado en el estandar RS485 para comunicar el PLC S7-1200 de Siemens y el variador de frecuencia Delta MS300
- Realizar la instalación y configuración completa de los componentes necesarios en el dispositivo pasarela, (Raspberry PI), con el fin de establecer la conexión del PLC con la nube para la supervisión y control de datos de manera remota.
- Crear un panel de control personalizado en Ubidots para visualizar los parámetros de operación del variador de frecuencia y emitir acciones de control básicas para la operación del motor.

Introducción

En el actual paradigma de la automatización industrial, la eficiencia y la flexibilidad constituyen pilares fundamentales para el éxito operativo de las compañías. En este escenario, la implementación de tecnologías avanzadas no solo busca la optimización del rendimiento en los procesos de producción, sino que también se erige como un medio propicio para implementar soluciones inteligentes de bajo costo destinadas al monitoreo y el control instantáneo de grupos de trabajo a través de la red mundial de comunicación, comúnmente conocida como la red de internet.

La búsqueda constante de las mejoras en la eficiencia operativa ha situado a la automatización industrial en el centro de las estrategias empresariales. La convergencia de tecnologías avanzadas desempeña un papel esencial al proporcionar soluciones que no solo optimizan la productividad, sino que también ofrecen alternativas económicamente viables para la gestión remota de procesos.

En el contexto de este trabajo, se enfoca la atención en la pasarela inteligente Raspberry Pi, que desempeña un papel esencial como puente tecnológico que conecta el convertidor de frecuencia Delta con el autómatas programable S7-1200. Esta pasarela no solo establece una interfaz de comunicación, sino que también abre las puertas a la conectividad remota, permitiendo así el monitoreo y control a distancia de los dispositivos vinculados al PLC. La Raspberry Pi, gracias a su capacidad de procesamiento y versatilidad, se posiciona como una herramienta esencial para la configuración de un sistema integral.

Capítulo 1

Comunicación Modbus y VFD

Este capítulo proporciona una visión general de la comunicación Modbus RTU, resaltando sus características clave y beneficios. Además, se explora el estándar RS-485, fundamental para la implementación de Modbus RTU, y se expone la razón detrás de su elección en este contexto.

La relevancia de estas tecnologías se destaca mediante la evaluación de su aplicación en una red de comunicación diseñada para facilitar la interconexión entre un PLC S7-1200 y un variador de frecuencia Delta MS300. Este análisis sienta las bases para comprender la utilidad práctica de Modbus RTU y RS485 en la automatización industrial.

1.1. Comunicación Modbus RTU

El protocolo MODBUS se ha convertido en un estándar ampliamente utilizado en la industria debido a su capacidad para facilitar la comunicación entre dispositivos en entornos industriales diversos. Este capítulo inicial ofrecerá una mirada profunda a las características y la cobertura de la comunicación MODBUS, así como una guía práctica para implementarla con un variador de frecuencia Delta [3].

1.1.1. Que es Modbus RTU ?

Desde su desarrollo por Modicon en 1979, el protocolo MODBUS se ha erigido como un pilar fundamental en la industria. Su creación marcó un hito

en la comunicación industrial, proporcionando una solución universal, abierta y fácil de usar que ha perdurado a lo largo de las décadas. A lo largo de los años, MODBUS ha demostrado ser un estándar de confianza en la automatización industrial, adaptándose a los avances tecnológicos y manteniendo su relevancia en un entorno industrial en constante evolución [4].

En el paisaje actual de la industria, donde los PLC, PAC, dispositivos e instrumentos E/S pueden tener interfaces Ethernet o inalámbricas, MODBUS continúa siendo el protocolo preferido por muchos. Su versatilidad es una de sus características más destacadas, ya que puede funcionar con una amplia gama de medios de comunicación, desde cables de par trenzado hasta conexiones inalámbricas. Esta capacidad lo hace adecuado para una variedad de entornos industriales y aplicaciones, proporcionando una solución confiable y flexible para la comunicación entre dispositivos.

MODBUS se ha establecido como un elemento fundamental en la industria. Su creación marcó un hito en la comunicación industrial, ofreciendo una solución universal, abierta y de fácil implementación que ha perdurado a lo largo de las décadas. Con el paso del tiempo, MODBUS ha demostrado ser un estándar confiable en la automatización industrial, adaptándose a los avances tecnológicos y manteniendo su relevancia en un entorno industrial en constante cambio [5].

MODBUS RTU es un protocolo de comunicación serie abierto punto a punto utilizado para desarrollar la comunicación denominada Multi-Master, Slave/Server-Cliente entre dispositivos inteligentes.

MODBUS RTU es compatible con diferentes tipos de conexiones físicas, incluyendo RS-232, RS-422 y RS-485, cada una con sus propias especificaciones de cableado y conectores. Por ejemplo, RS-232 funciona bajo un esquema dúplex, lo que significa que permite la transferencia de datos en ambos sentidos de forma simultánea, mientras que RS-485 es semidúplex y realiza estas operaciones secuencialmente. Además, cada estándar puede diferenciarse por la velocidad de baudios y la tasa de transferencia de datos que soporta, lo que influye en la eficiencia y velocidad de la comunicación [6].

La capa física en MODBUS RTU es responsable de varios aspectos, como la

dirección esclava, los bits de inicio y parada en los datos, el cálculo de CRC, el tiempo de espera y la detección de errores de trama. Por otro lado, la capa de enlace de datos maneja el reconocimiento o rechazo de códigos de función, datos ocupados o datos de reagrupación [6].

En cuanto al código de función en MODBUS RTU, se utiliza una representación en coma flotante de 32 bits, la cual se codifica en un formato entero. Este código es fundamental para la comunicación entre dispositivos. El funcionamiento del código de función es el siguiente: los dispositivos maestros envían consultas a los dispositivos esclavos, especificando el código de función correspondiente. A su vez, los dispositivos esclavos responden a estas consultas del maestro de acuerdo con el código especificado.

La Figura 1.1 muestra el formato en el que se comparten mensajes en MODBUS RTU, ilustrando la estructura y los campos relevantes de los mensajes en la comunicación.

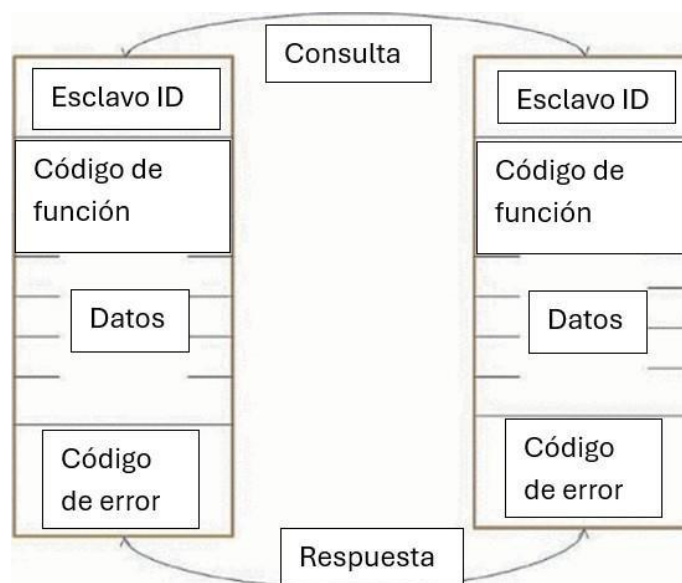


Figura 1.1: Ciclo de mensajes en el campo de mensajes de modbus RTU

1.1.2. Comunicación bajo el estándar RS485

El RS-485, denominado también como EIA/TIA-485, constituye un avance notable en el ámbito de la comunicación física, ofreciendo un sistema robusto y confiable para la transmisión de señales, el RS-485 emergió como una solución a la imperiosa necesidad de extender las funcionalidades físicas del puerto RS-232, ofreciendo un rendimiento superior en entornos industriales y comerciales [7].

La conexión en serie RS-485 se configura a través de un tendido que posee dos o tres conductores, en el que uno de ellos transmite la información de forma invertida y, usualmente, se integra un cable de tierra (0V). Esta configuración posibilita el intercambio de datos entre transmisores y receptores mediante un par trenzado de hilos rígidos de calibre 22 o 24 AWG, lo que asegura una transmisión eficiente y fiable.

El RS-485 sobresale por su habilidad para conducir un mensaje a través de dos conductores: uno para la información original y otro para su réplica invertida. Esta técnica de comunicación presenta una notable inmunidad a las perturbaciones en modo común, lo que lo convierte en la opción preferida para entornos donde la integridad de la señal es crucial. Gracias a esto, se posibilita el envío de información a distancias considerables y a velocidades relativamente elevadas, llegando a alcanzar hasta 100 kbits/s a una distancia máxima de 4000 pies (o aproximadamente 1200 metros) en comunicaciones RS-485 [8].

Además, el uso de cables de par trenzado en RS-485 permite adaptarse a diferentes necesidades como se muestra en la figura 1.2, ya que estos cables pueden ser blindados o no, dependiendo del entorno y las condiciones específicas de aplicación. Esta versatilidad asegura que RS-485 sea una solución adaptable y confiable para una variedad de escenarios de comunicación.

Protocolo de comunicación RS-485

Luego de que el canal de comunicación RS-485 se halla completamente preparado para funcionar a nivel tangible, es fundamental establecer un protocolo de transferencia de datos. Esto implica que las unidades del sistema definan una guía estándar para la transferencia de información fragmentada

Las características de la interfaz RS-485 impiden que los dispositivos transmitan y reciban datos al mismo tiempo, puesto que esto genera la posibilidad

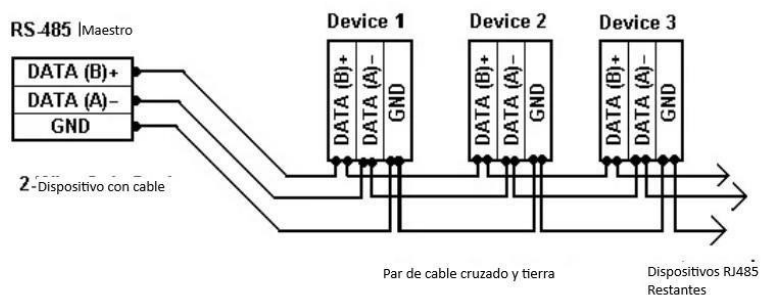


Figura 1.2: Conexión RS-485

de choques entre unidades de envío. En consecuencia, se vuelve fundamental la implementación de un comportamiento predefinido para prevenir enfrentamientos en la comunicación de bloques de datos.

En el protocolo de comunicación RS-485, el nodo maestro es el responsable de la emisión de comandos, mientras que los nodos subordinados conectados al maestro reciben la información a través de los puertos RS-485. Sin embargo, es importante destacar que varios nodos pueden responder al maestro dependiendo de la información enviada. Sin embargo, esta clase de interacción no configura la única alternativa viable para la transferencia de información dentro del sistema en cuestión. No obstante, existen otras modalidades de redes RS-485 donde cada nodo posee la capacidad de iniciar una transferencia de datos de forma autónoma. Sin embargo, en estas redes, la probabilidad de incurrir en pérdidas de información suele ser considerablemente mayor.

■ **Atributos sobresalientes de la interfaz RS-485:**

- Comunicación bidireccional mediante un par de hilos trenzados.
- Habilidad de redes mediante la conexión de múltiples transceptores
- Comunicación a distancias considerables
- Transmisión de información a ritmo acelerado

Entre las principales características de la comunicación bajo el estándar RS-485 se encuentran las siguientes:

- **Transmisión de datos bidireccional semidúplex:** En este tipo de comunicación, el flujo de datos serie puede transportarse en una única dirección a la vez. Por lo

tanto, para que la transferencia de datos ocurra en ambas direcciones, se requiere un transceptor, comúnmente conocido como controlador. Este dispositivo, o circuito eléctrico, forma una señal física en el lado del transmisor para facilitar la comunicación en ambas direcciones.

- **Canal de comunicación simétrico:** En este caso se tiene la necesidad de dos hilos de señal equivalentes para la transmisión y recepción de datos.

Estos cables se utilizan para enviar y recibir datos en ambas direcciones a través de un cable que tiene dos conductores entrelazados, estos son los encargados de la conducción de la información de forma bidireccional

El uso de un canal simétrico sirve en gran manera ayudando a estabilizar dicha señal además de que reduce la interferencia electromagnética que pueda ser generada por la señal útil.

- **Multipunto:** En este caso, la conexión RS-485 puede operar con varios dispositivos receptores y transceptores enlazados al mismo tiempo. Esto significa que es posible conectar tanto transmisores como receptores a la línea de comunicación al mismo tiempo. No obstante, cualquier otro dispositivo emisor que desee conectarse deberá aguardar hasta que el canal de comunicación esté desocupado para el envío de datos.

A lo largo de más de medio siglo, los protocolos de comunicación serie RS-485 y RS-232 han sido empleados y siguen siendo ampliamente utilizados en una variedad de empresas e industrias en la actualidad. El desarrollo del protocolo RS-485 se basó en la expansión de las capacidades del estándar RS-232, lo que resultó fundamental para su evolución. A continuación, la tabla 1.1 proporciona una serie de las características que presentan ambos estándares para la comunicación serie.

Protocolo	RS232	RS485
Tipo de protocolo	Dúplex	Semi-dúplex
Tipo de señal	Desbalanceado	Balanceado
Número de dispositivos	1 transmisor y receptor	Hasta 32 transmisores y 43 receptores
Transmisión máxima de datos	19,2 Kbps a 15 m	10 Mbps a 15 m
Longitud máxima del cable	Aprox. 15,25 m a 19,2 Kbps	Aprox. 1220 m a 100 Kbps
Corriente de salida	500mA	250mA
Voltaje mínimo de entrada	+/- 3V	0,2V diferencial

Tabla 1.1: Comparación entre RS-232 y RS-485

1.1.3. Implementación en S7-1200

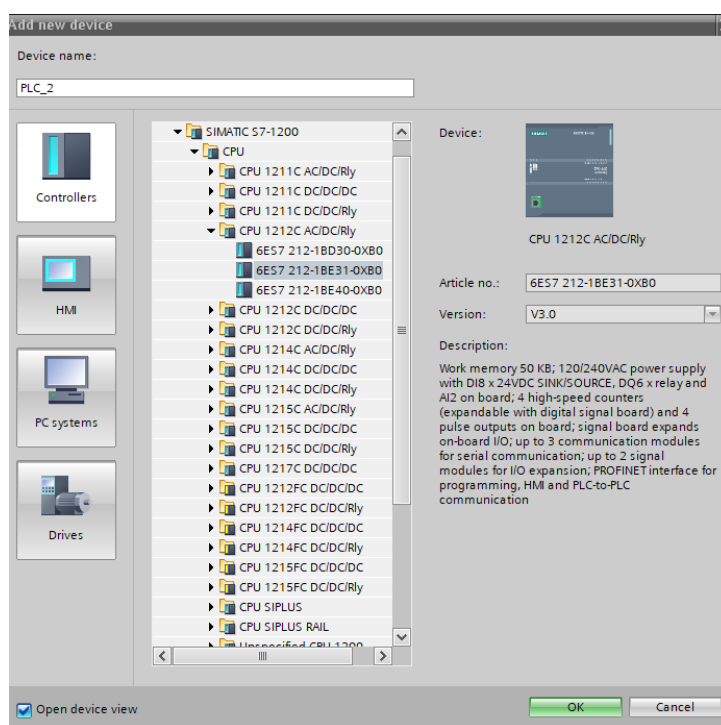


Figura 1.3: PLC S71200

En primera instancia se debe crear un proyecto en TIA Portal, para esto se debe seleccionar el PLC S7-1200 1212C AC/DC RLY en el programa de TIA Portal, siguiendo los siguientes pasos:

Abrir TIA Portal: Inicia el software TIA Portal en tu computadora.

Crear Nuevo Proyecto: Selecciona la opción para crear un nuevo proyecto.

Seleccionar PLC: En la ventana de creación de proyecto, elige el PLC adecuado para tu aplicación. En este caso, selecciona el PLC S7-1200 1212C AC/DC RLY.

Configuración del Proyecto: Define los parámetros del proyecto, como el nombre del proyecto, la ubicación de almacenamiento, etc. Asegúrate de proporcionar una descripción detallada de tu proyecto para una mejor comprensión.

Añadir Dispositivos: Una vez creado el proyecto, procede a añadir los dispositivos necesarios. En este caso, el PLC S7-1200 1212C AC/DC RLY. Esto se puede hacer desde la pestaña "Devices & Networks"(Dispositivos y Redes).

Configuración del PLC: Configura el PLC de acuerdo a tus requisitos específicos. Esto puede incluir la asignación de direcciones IP, configuración de módulos de expansión, etc.

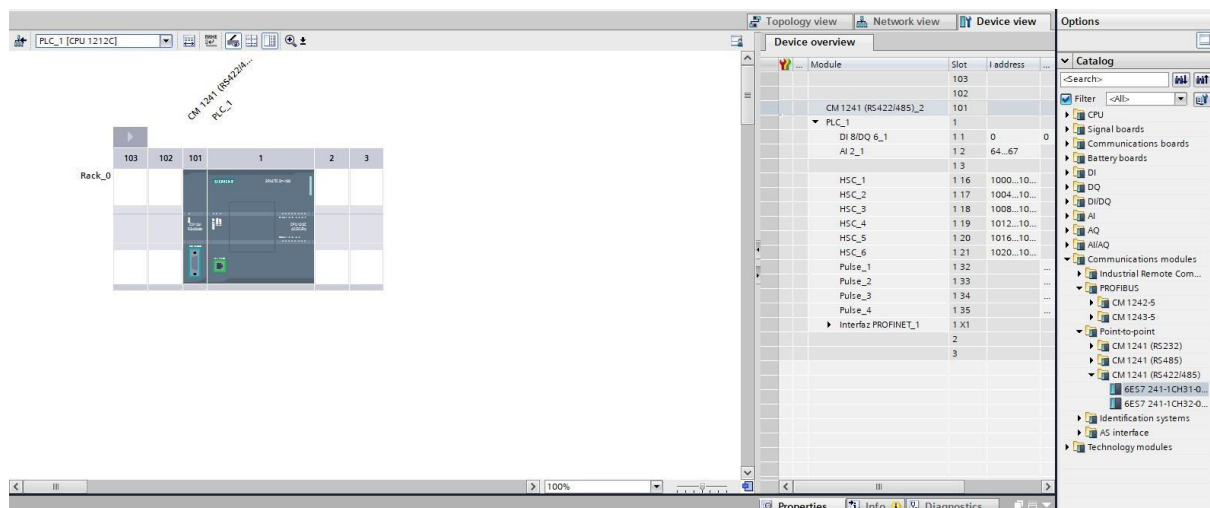


Figura 1.4: Módulo de Comunicación CM1241

Agregamos al proyecto el módulo de comunicación que vamos a utilizar en este caso el CM1241, como se observa en la figura 1.4.

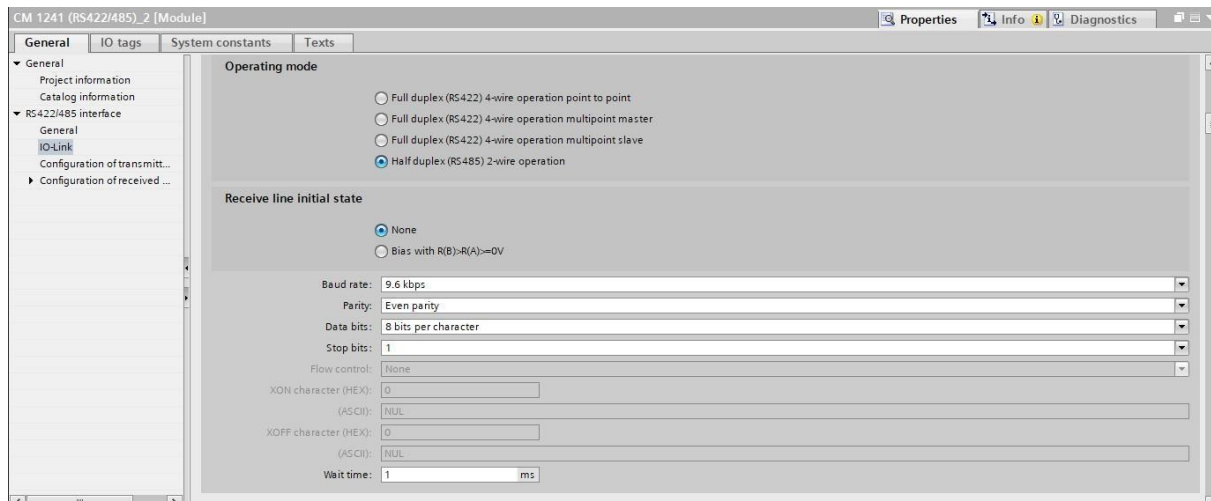


Figura 1.5: Configuración del Módulo

Se selecciona el módulo de comunicación CM1241 para complementar el PLC S7-1200 1212C AC/DC/RLY elegido previamente. Este módulo es esencial para facilitar la comunicación entre el PLC y el variador de frecuencia, utilizando el protocolo RS485, lo cual es crucial para el control y la gestión eficiente de la maquinaria, refiérase a la figura 1.5.

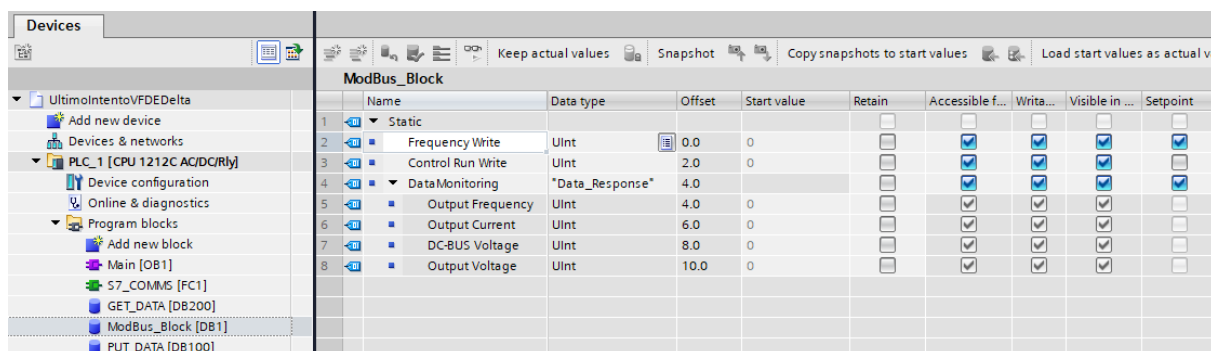


Figura 1.6: Bloque Modbus

En este espacio se crea el bloque de control de encendido, control de frecuencia y monitoreo de datos, con el nombre Modbusblock(DB1), esto debe ser creado similar a la figura 1.6.

The screenshot shows the 'Data_Response' configuration window in SIMATIC Manager. The left pane shows a project tree with the following structure:

- UltimolentoVFDEDelta
 - Add new device
 - Devices & networks
 - PLC_1 [CPU 1212C AC/DC/Rly]
 - Device configuration
 - Online & diagnostics
 - Program blocks
 - Technology objects
 - External source files
 - PLC tags
 - PLC data types
 - Add new data type
 - Data_Response (selected)
 - System data types
 - Watch and force tables
 - Online backups
 - Device proxy data
 - Program info
 - PLC alarm text lists

The right pane displays the 'Data_Response' table with the following columns: Name, Data type, Default value, Accessible f..., Writa..., Visible in..., and Setpoint.

	Name	Data type	Default value	Accessible f...	Writa...	Visible in ...	Setpoint
1	Output Frequency	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
2	Output Current	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
3	DC-BUS Voltage	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	Output Voltage	UInt	0	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Figura 1.7: Data Response

En el bloque data response tendremos los distintos datos que recopilara el PLC desde el variador de frecuencia, estos son la frecuencia de salida, corriente de salida, Voltaje de salida en el bus, y el voltaje de salida del variador, se puede observar el funcionamiento de los mismos en la figura 1.7.

1.2. Variador de frecuencia (VFD)

Los Variadores de Frecuencia, conocidos también como Variable Frequency Drives (VFD), representan una tecnología fundamental en la regulación precisa de la velocidad y la fuerza de los motores de corriente alterna (CA), logrando esto mediante la modulación de la frecuencia y el voltaje suministrado [9]. Estos dispositivos son esenciales para regular tanto la aceleración como la desaceleración durante el arranque y la parada del motor. Un VFD se compone principalmente de tres componentes significativos: un rectificador, un inversor y un sistema de control.

La versatilidad del VFD lo convierte en una herramienta indispensable en una amplia gama de aplicaciones industriales, incluyendo sistemas de ventilación, ascensores, y escaleras mecánicas, entre otros. Su capacidad para proporcionar un control preciso y eficiente del motor lo hace altamente ventajoso en comparación con otros dispositivos de control, como los arrancadores suaves o los arrancadores de línea [10].

Por otro lado, la diversidad en los tipos de variadores de frecuencia refleja la variedad de aplicaciones que abarcan. Estos dispositivos se clasifican en diferentes categorías según su topología, que incluyen el inversor de voltaje (VSI), el inversor de corriente (CSI) y el inversor de carga conmutada (LCI) [11].

Además, la clasificación de los variadores de frecuencia también puede basarse en las características de la carga que alimentan, como el par variable, el par constante o la potencia constante. Esta variedad de clasificaciones permite adaptar los variadores de frecuencia a una amplia gama de necesidades y requisitos específicos de aplicación.

El VFD más común que se puede encontrar es el variador VSI (Figura 1.8), el cual puede convertir una entrada de CA de voltaje fijo, al igual que la frecuencia también fija, en una salida de CA de voltaje y frecuencia variable. En este caso, el rectificador convierte la entrada de CA en CC, el enlace de CC suaviza la salida de CC mediante un condensador, y el inversor conmuta la salida de CC para producir una salida de CA, que se acerca mucho a una forma sinusoidal, utilizando dispositivos electrónicos de potencia como los IGBT o GTO [10].

El rectificador se encarga de convertir la corriente alterna de entrada en corriente continua (CC), mientras que el enlace de CC, compuesto por un capacitor,

suaviza la salida de CC [12]. Finalmente, el inversor, utilizando dispositivos electrónicos de potencia como los IGBT (Transistores Bipolares de Puerta Aislada) o los GTO (Gate Turn-Off Thyristors), conmuta la salida de CC para producir una salida de CA que se asemeja mucho a una onda sinusoidal.

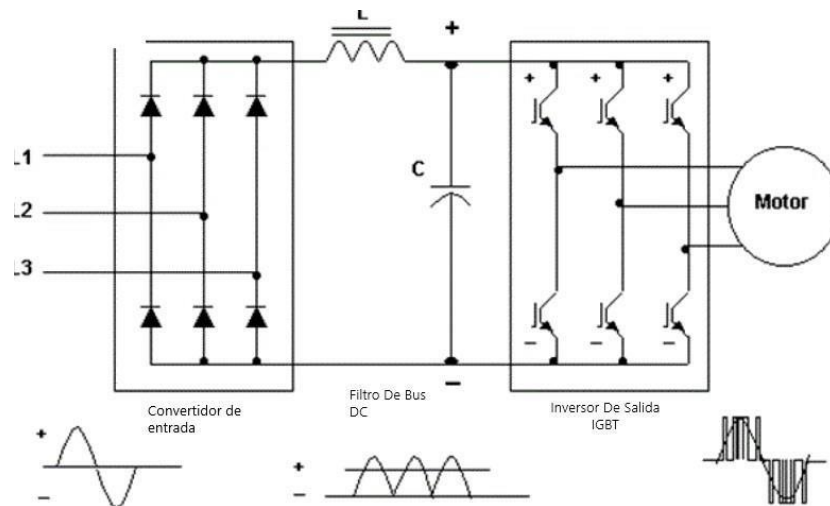


Figura 1.8: Diagrama eléctrico del VSI [13]

- **Al utilizar un Variador de Frecuencia (VFD), se obtienen diversas ventajas en comparación con otros métodos de control de velocidad. Algunas de estas ventajas incluyen:**

 - **Ahorro de energía:** Un variador de frecuencia variable (VDF) puede contribuir a la disminución del consumo de energía al ajustar la velocidad y el caudal del motor en función de la carga y las necesidades del proceso. Esto no solo ayuda a reducir el desperdicio de energía y a mejorar el sistema, sino que también un VDF es capaz de ayudar a mejorar el factor de potencia del sistema al reducir la potencia reactiva consumida por el motor.
 - **Mayor Confiabilidad:** Esto significa que nos puede ayudar a mejorar la confiabilidad del motor y del sistema al proporcionar un control suave y preciso de la velocidad y la fuerza de giro del motor. Esto puede eliminar las tensiones mecánicas y los impactos causados por el arranque o parada directa en línea del motor, lo que podría dañar los devanados, cojinetes, correas, entre otros. Además, puede proteger al motor de fallas eléctricas

como la sobretensión, subtemperatura, cortocircuito, etc., monitorizando y ajustando el voltaje y la corriente de salida. Un VFD también es capaz de reducir los niveles de ruido y vibración del motor y del sistema, lo que ayuda a mejorar el entorno de trabajo y prolongar la vida útil del equipo.

- **Variaciones de velocidad:** Se refiere a la capacidad de ajustar la velocidad del motor desde 0 hasta superar la velocidad nominal según las cargas y los requisitos del proceso.
- **Arranque suave:** Es capaz de realizar esto aumentando gradualmente la frecuencia y el voltaje del motor desde 0 hasta el valor deseado, lo que reduce la corriente del motor que normalmente es de seis a diez veces su corriente nominal cuando se arranca directamente en línea.
- **Mayor vida útil de la máquina y menos mantenimiento:** Gracias a las ventajas mencionadas anteriormente, el VFD es capaz de extender la vida útil de la máquina y con ello reducir los costos de mantenimiento. El VFD puede disminuir el deterioro del motor y del sistema al ofrecer un control suave y preciso de la velocidad y el torque. Asimismo, puede supervisar y diagnosticar la condición en la que se encuentra un motor.

El mercado de los Variadores de Frecuencia (VFD) ofrece una amplia gama de opciones, con numerosas marcas reconocidas que ofrecen productos específicos con características y aplicaciones particulares. Entre las marcas más destacadas se encuentran Siemens, Allen Bradley, Schneider Electric, Delta, Omron, ABB, WEG y Danfoss. Estas marcas líderes ofrecen una variedad de opciones que abarcan desde aplicaciones industriales generales hasta aplicaciones especializadas en sectores específicos. La elección del VFD adecuado depende en gran medida de la aplicación específica para la cual se destinará. Cada marca y modelo de VFD tiene sus propias características y capacidades, por lo que es importante considerar aspectos como la compatibilidad con los protocolos de comunicación requeridos, la capacidad de control y monitoreo, así como también el costo y la disponibilidad de soporte técnico.

Para el trabajo propuesto, se ha optado por utilizar el VFD de la marca Delta debido a su compatibilidad con los protocolos de comunicación necesarios y su costo

competitivo. Esta elección se basa en la relación costo-beneficio y en la capacidad del VFD Delta para cumplir con los requisitos específicos del proyecto. Sin embargo, es importante destacar que la elección final de un VFD debe basarse en un análisis exhaustivo de las necesidades y requisitos de la aplicación en cuestión.

1.3. Integración VFD Delta y PLC S7-1200 mediante Modbus RTU bajo el estándar RS-485

Se necesita establecer los parámetros mínimos de comunicación como se muestra en la tabla 1.2. Para esto entramos en el modo de configuración de parámetros en el variador de frecuencia y ajustamos los parámetros según se indica en la Tabla 1.2. Cabe recalcar que estos parámetros fueron los que se usaron para la elaboración de este proyecto. El operador puede modificar estos a su conveniencia.

Parameter	Value	Description
Parámetro 00.20	1	Establecer la frecuencia por Modbus
Parámetro 00.21	2	Controlar el motor por Modbus
Parámetro 09.00	1	Direccion = 1
Parámetro 09.01	9.6	9600 Bps
Parámetro 09.02	0	Sin Uso
Parámetro 09.03	1.0	Tiempo de error en la comunicacion
Parámetro 09.04	14	Elegir el Protocolo Modbus

Tabla 1.2: Parámetros a Configurar en el Variador de Frecuencia

Una vez establecidos los parámetros en el variador de frecuencia, se configura los registros destinados al control de direcciones. Esta tarea se lleva a cabo directamente en el software TIA Portal. La Figura 1.9 muestra la tabla de registros correspondiente.

Después de analizar los registros de la figura 1.9 se define los registros que serán utilizados para el control de dirección del motor y de la frecuencia. Estos registros se sintetizan en las tabla 1.3.

Se realizan los cálculos para el monitoreo de datos a partir de la tabla proporcionada en la figura 1.10

Todas las tablas de direcciones se sacaron del manual del variador delta

4. Address list

Content	Address	Function	
AC motor drive parameters	GGnnH	GG is the parameter group, nn is the parameter number; for example, the address of Pr.04-10 is 040AH.	
Command write only	2000H	bit 1-0	00B: No function 01B: Stop 10B: Run 11B: JOG + RUN
		bit 3-2	Reserved
		bit 5-4	00B: No function 01B: FWD 10B: REV 11B: Change direction
		bit 7-6	00B: 1 st accel. / decel. 01B: 2 nd accel. / decel. 10B: 3 rd accel. / decel. 11B: 4 th accel. / decel.
		bit 11-8	000B: Master speed
			0001B: 1 st step speed frequency
			0010B: 2 nd step speed frequency
			0011B: 3 rd step speed frequency
			0100B: 4 th step speed frequency
			0101B: 5 th step speed frequency
			0110B: 6 th step speed frequency
			0111B: 7 th step speed frequency
			1000B: 8 th step speed frequency
			1001B: 9 th step speed frequency
			1010B: 10 th step speed frequency
1011B: 11 th step speed frequency			
1100B: 12 th step speed frequency			
1101B: 13 th step speed frequency			
1110B: 14 th step speed frequency			
1111B: 15 th step speed frequency			
bit 12	1: Enable bit 06-11 function		
bit 14-13	00B: No function		
	01B: Operated by the digital keypad		
	10B: Operated by Pr.00-21 setting		
bit 15	11B: Change the operation source		
	Reserved		
2001H	Frequency command (XXX.XX Hz)		
2002H	bit 0	1: E.F. (External Fault) ON	
	bit 1	1: Reset command	

Figura 1.9: Tabla direcciones 1

Operation	Command
Forward Run	Set Word 48193 = 18
Reverse Run	Set Word 48193 = 34
Stop Run	Set Word 48193 = 1

Tabla 1.3: Motor Control Commands

Address Description	Value (dec)
Motor Control Address	40001 + 8192 = 48193
Frequency Address	40001 + 8193 = 48194

Tabla 1.4: Address Calculations

VFD-MS300, por tanto, se puede consultar el manual en caso de ser necesario.

Para la programación en TIA portal es importante configurar los bloques como se explica a continuación:

En el bloque de la figura 1.11 se establece la frecuencia de reloj a la cual trabajara el

Content	Address	Function	
		bit 2 1: B.B. ON	
		bit 4-3 Reserved	
		bit 5 1: Enable fire mode	
		bit 15-6 Reserved	
Status monitor read only	2100H	High byte: Warning code / Low Byte: Error code	
	2101H	bit 1-0	AC motor drive operation status 00B: The drive stops 01B: The drive is decelerating 10B: The drive is in standby status 11B: The drive is operating
		bit 2	1: JOG command
		bit 4-3	Operation direction 00B: FWD running 01B: From REV running to FWD running 10B: From FWD running to REV running 11B: REV running
		bit 8	1: Master frequency controlled by the communication interface
		bit 9	1: Master frequency controlled by the analog / external terminal signal
		bit 10	1: Operation command controlled by the communication interface
		bit 11	1: Parameter locked
		bit 12	1: Enable to copy parameters from keypad
		bit 15-13	Reserved
		2102H	Frequency command (XXX.XX Hz)
		2103H	Output frequency (XXX.XX Hz)
		2104H	Display the drive's output current (XX.XX A). When the current is higher than 655.35, it automatically shifts one decimal place as (XXX.X A). Refer to the high byte of 211F for information on the decimal places.
		2105H	DC bus voltage (XXX.X V)
		2106H	Output voltage (XXX.X V)
2107H	Current step for the multi-step speed operation		
2108H	Reserved		
2109H	Counter value		
210AH	Output power factor angle (XXX.X)		
210BH	Output torque (XXX.X %)		
210CH	Actual motor speed (XXXXX rpm)		
210DH	Reserved		
210EH	Reserved		
210FH	Power output (X.XXX kW)		
2116H	Multi-function display (Pr.00-04)		
211BH	Maximum Operation Frequency (Pr.01-00) or Maximum User-defined Value (Pr.00-26) When Pr.00-26 is 0, this value is equal to Pr.01-00 setting When Pr.00-26 is not 0, and the command source is keypad, this value = Pr.00-24 * Pr.00-26 / Pr.01-00. When Pr.00-26 is not 0, and the command source is 485, this value = Pr.09-10 * Pr.00-26 / Pr.01-00.		
	211FH	High byte: the decimal place of current value (display)	
2200H	Display the drive's output current (XX.XX A). When the current is higher than 655.35, it automatically shifts one decimal place as (XXX.X A). Refer to the high byte of 211F for information on the decimal places.		
2201H	Counter value		
2202H	Actual output frequency (XXXXX Hz)		

Figura 1.10: Tabla direcciones 2

Parameter	Value (dec)
Output Frequency	40001 + 2103 (hex) = 40001 + 8451 = 48452
Output Current	48453
DC-BUS Voltage	48454
Output Voltage	48455

Tabla 1.5: Parámetros eléctricos y sus direcciones

programa.

En el bloque de la figura 1.12 se inicializa un **first scan** para inicializar las variables que van a ser utilizadas. Seguido a esto se le asigna a port la dirección del módulo que vamos a utilizar. A continuación, se establece la velocidad de transmisión de datos en baudios. En este caso 9600 baudios. Finalmente se pone el bit de paridad 2 como fue configurado anteriormente.

En el bloque de la figura 1.13, se procesa el impulso generado en la primera línea

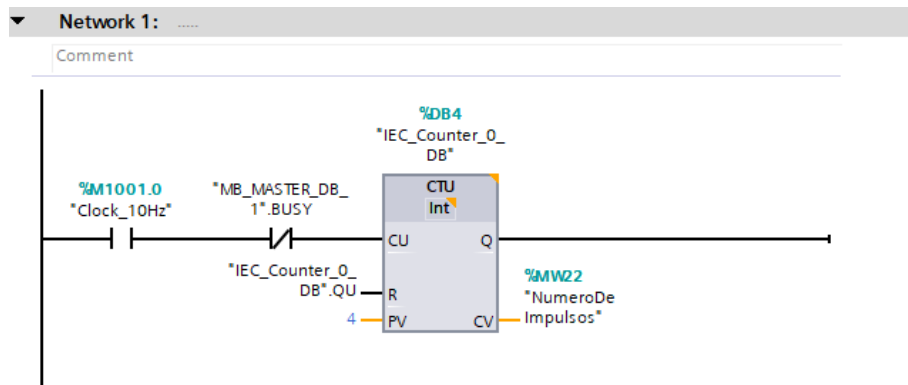


Figura 1.11: Distribución De Lectura y Escritura

+ Communication Load Block

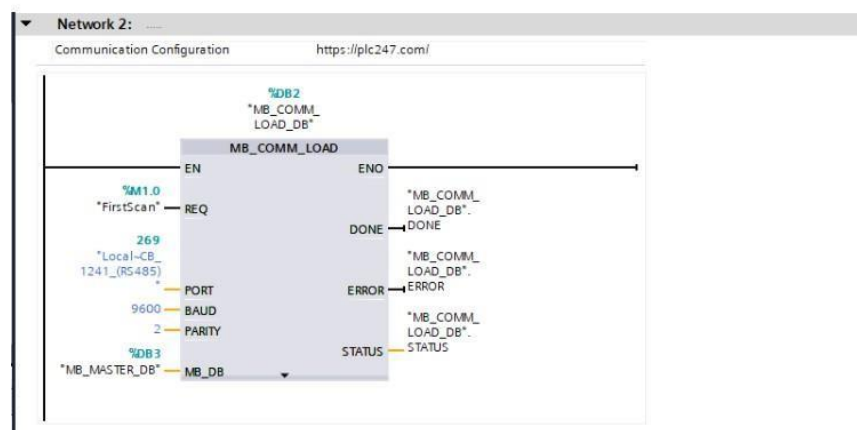


Figura 1.12: Bloque de carga de la comunicación

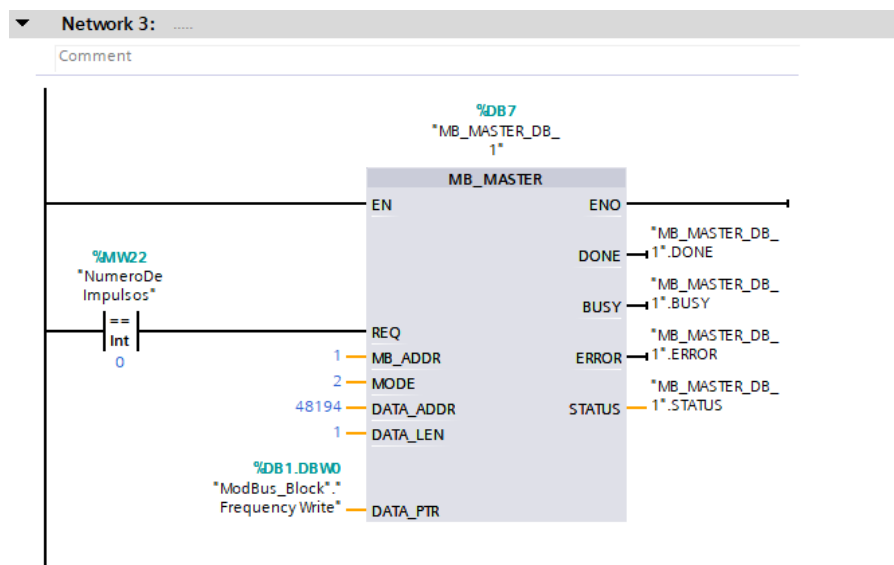


Figura 1.13: Bloque para establecer frecuencia

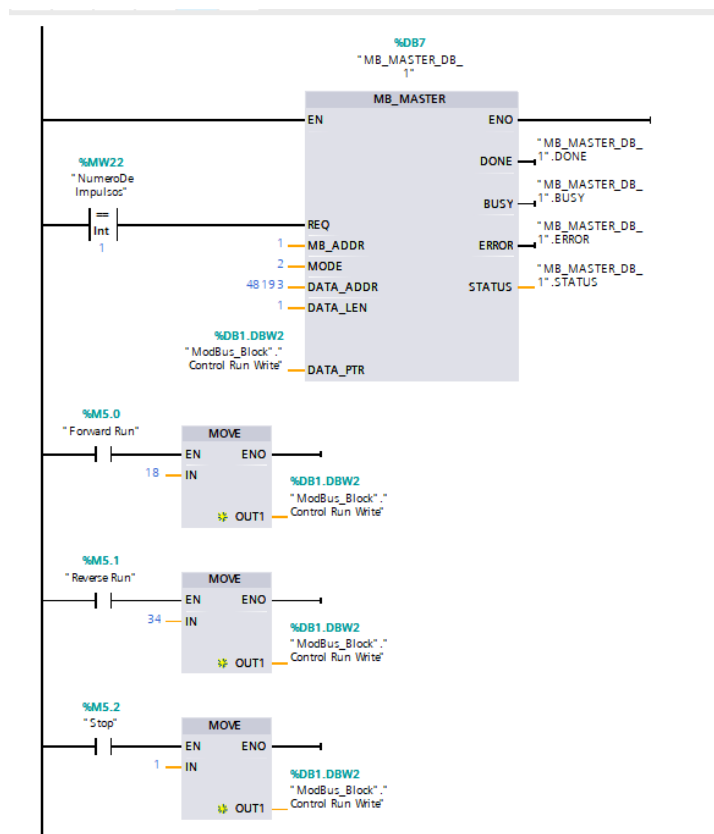


Figura 1.14: Comando para el control del motor

de código. Cuando el valor del impulso es 0, el módulo maestro procede a leer la frecuencia enviada al variador de frecuencia y posteriormente la transcribe en el mismo.

Es importante destacar que, como se indica en **Data_addr**, la dirección utilizada es la 48194. Esta especificación nos permite escribir directamente en la dirección de control de frecuencia del variador. Este detalle se determinó gracias a un análisis previo relacionado con el manejo de direcciones para el control eficiente del dispositivo.

Cuando se detecta un impulso igual a 1, el bloque MB Master ejecutará la escritura del dato de control de motor en el variador de frecuencia. La selección del dato específico a escribir depende de cuál memoria está activa en ese momento. Por ejemplo, si la memoria M5.0 (como se ilustra en la Figura 1.14) está activa, el bloque MOVE enviará el dato 18 para el control del motor. Cada uno de estos datos ha sido calculado con antelación y se detalla en la Tabla 1.3.

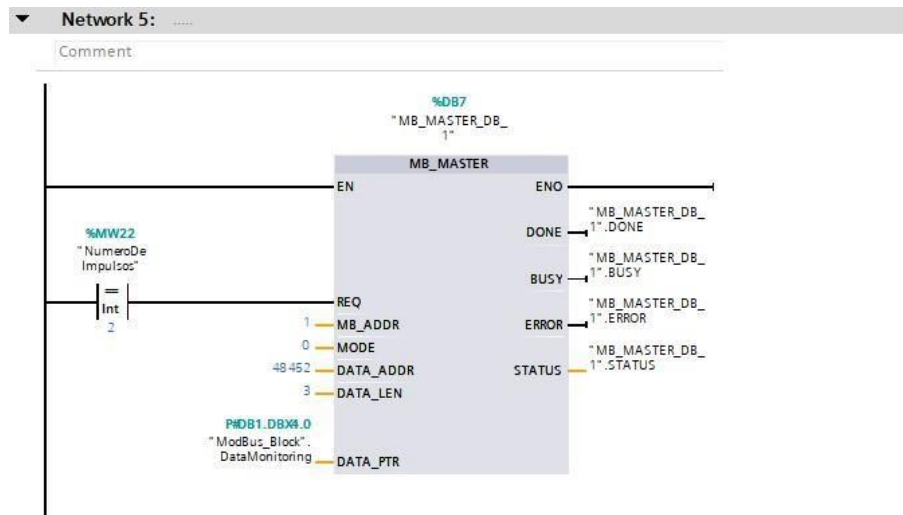


Figura 1.15: Comando para la lectura de datos

DataMonitoring		"Data_Response"	4.0		
Output Frequency	UInt	4.0	0		
Output Current	UInt	6.0	0		
DC-BUS Voltage	UInt	8.0	0		
Output Voltage	UInt	10.0	0		

Figura 1.16: Datos Obtenidos Del Variador

En esta parte del programa, cuando se detecta un impulso con valor de 2, el módulo MB Master procederá a recibir los datos monitorizados por el variador de frecuencia. Estos datos serán almacenados en un arreglo denominado Data Monitoring. Referirse a la figura 1.16 para observar el funcionamiento del arreglo de datos.

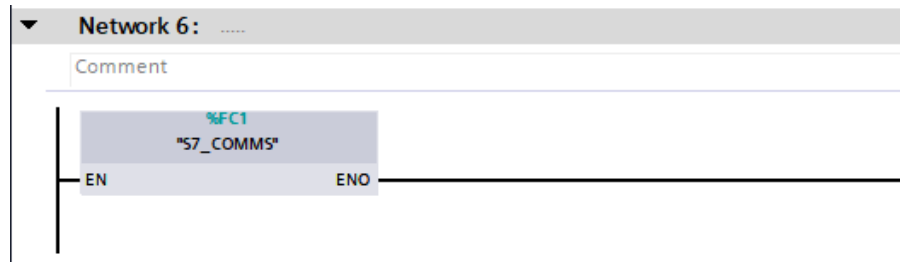


Figura 1.17: Bloque de envío de datos

El bloque de la figura 1.17 se dedicará a la transmisión de datos hacia Ubidots. Los detalles específicos acerca de su funcionamiento se abordarán con mayor profundidad en el capítulo siguiente.

Capítulo 2

Pasarela inteligente

En la actualidad, el Internet ha adquirido una gran importancia al convertirse en una fuente inagotable de información. En los albores de los años 90, el término "Internet" se empleaba principalmente para describir la tecnología que permitía interconectar computadoras de todo el mundo a través de enlaces cableados o inalámbricos. Desde entonces, Internet ha sido aprovechado eficientemente para compartir archivos, navegar por la web, acceder a redes sociales, entre otras funciones. Sin embargo, el desarrollo y la implementación de las últimas tecnologías inteligentes demandan una conexión ubicua a la nube [14].

Como consecuencia de esto, surge la necesidad de tecnologías sofisticadas para habilitar una nueva forma de comunicación conocida como máquina a máquina (M2M) [3]. Esta forma de comunicación ofrece una variedad de beneficios, entre ellos, aumenta la operatividad, mejora la calidad, facilita la toma de decisiones y tiene un impacto positivo en el medio ambiente al permitir el monitoreo y control de negocios, empresas o emprendimientos desde cualquier parte del mundo.

Las empresas que buscan automatizar y supervisar sus procesos a menudo optan por adquirir controladores adecuados para estas tareas. Entre los controladores más destacados se encuentran los PLCs de la marca Siemens. Estos controladores compactos facilitan la ejecución de tareas productivas simples que requieren precisión. Siemens ofrece una amplia gama de PLCs, que se pueden dividir en varias gamas según su capacidad y aplicación. La gama baja incluye el S7-200, diseñado para aplicaciones compactas y simples. La gama media abarca los modelos S7-1200 y

S7-300, adecuados para maquinaria y procesos medianos. Por último, la gama alta comprende los modelos S7-1500 y S7-400, utilizados en procesos complejos y grandes instalaciones.

Para conectar los controladores de la marca Siemens, como los PLCs, a menudo se requiere una pasarela inteligente. Siemens ofrece pasarelas propias para este propósito

2.1. Pasarela inteligente de Siemens

Una de las pasarelas más utilizadas ofrecidas por Siemens es la SIMATIC IoT2040, la cual representa una actualización de hardware con respecto a la IoT2020. Esta última es una placa compacta que sigue el estilo de dispositivos como Raspberry Pi o Arduino, pero con la distinción de estar industrializada y adaptada específicamente a las necesidades de los equipos de Siemens [15]. Como una evolución a la pasarela IoT2020, en los últimos años ha surgido una nueva versión denominada IoT2040. Las diferencias más destacadas entre la pasarela IoT2020 y la pasarela IoT2040, se encuentran las especificadas en la tabla 2.1.

Característica	IoT2020	IoT2040
Procesador	Intel Quark® x1000	Intel Quark® x1020
RAM	512 MB	1 GB
Interfaces Ethernet	1	2
Interfaces RS232/485	Sin puertos extras	2 x RS232/485
RTC (Reloj en tiempo real)	Sin RTC	Con RTC y batería

Tabla 2.1: Comparación entre IoT2020 y IoT2040

Para programar esta pasarela, se puede utilizar el IDE de Eclipse Arduino o Node-RED. Si el objetivo no es enviar grandes cantidades de datos a la nube, Node-RED sería la opción más adecuada debido a su facilidad de uso. Dado que se trata simplemente de una pasarela, no se necesita un programa muy complejo, sino más bien una solución que extraiga los datos de los PLCs y los envíe donde sea necesario [16].

2.2. Raspberry pi como pasarela inteligente

El Raspberry Pi es una pequeña computadora de bajo costo y tamaño compacto, similar al de una tarjeta de crédito. Puede conectarse a un monitor o a una televisión y utilizarse con un mouse y un teclado estándar [17]. Este dispositivo, que funciona con un sistema operativo Linux, permite a cualquier persona explorar el mundo de la computación y aprender a programar en lenguajes como Scratch y Python. Es capaz de realizar la mayoría de las tareas típicas de una computadora de escritorio, desde navegar por internet hasta reproducir videojuegos [18]. Además, el Raspberry Pi tiene la capacidad de interactuar con el exterior, lo que lo hace adecuado para una amplia gama de proyectos digitales.

Dado que existen versiones de más de dos docenas de estos mini computadoras, seleccionar los modelos más comunes y útiles puede resultar un desafío. Para simplificar esta elección, se presenta en la tabla 2.2 los modelos más comunes y utilizados, junto con sus características respectivas.

MODELO	CPU	RAM	CONECTIVIDAD INALÁMBRICA	PUERTOS E/S
Raspberry Pi 4B	1.5 GHz, 4-core Broadcom BCM2711	2/4/8GB	802.11ac / Bluetooth 5.0	2x USB 3.0, 2x USB 2.0, 1x Gigabit Ethernet, 2x micro HDMI
Raspberry Pi 3 B+	1.4 GHz, 4-core Broadcom BCM2837B0	1GB	802.11ac, Bluetooth 4.2, Ethernet	4x USB 2.0, HDMI, 3.5mm audio
Raspberry Pi Zero W	1 GHz, 1-core Broadcom BCM2835	512MB	802.11n / Bluetooth 4.1	1x micro USB, 1x mini HDMI
Raspberry Pi Zero WH	1 GHz, 1-core Broadcom BCM2835	512MB	802.11n / Bluetooth 4.1	1x micro USB, 1x mini HDMI
Raspberry Pi Zero	1 GHz, 1-core Broadcom BCM2835	512MB	-	1x micro USB, 1x mini HDMI

Tabla 2.2: Comparación de modelos de Raspberry Pi

Dadas las prestaciones y características que ofrece, para este proyecto se optó por utilizar la Raspberry Pi 4 como pasarela inteligente. Si bien el uso de pasarelas

inteligentes de marcas como Siemens es una opción lógica y garantiza un mejor rendimiento en entornos industriales, no siempre es la única alternativa viable. Para algunas aplicaciones menos complejas, hay empresas que no pueden costear el precio de estos módulos más costosos. En tales casos, una Raspberry Pi 4 puede ser un reemplazo adecuado y más económico.

2.2.1. Node-RED en Raspberry pi

Node-RED es una herramienta de desarrollo de código abierto basada en el flujo, diseñada para la integración entre dispositivos de hardware de IoT y API (interfaces de programación de aplicaciones), así como para la interacción con servicios en línea desarrollados por IBM Emerging Technology [19].

Esta herramienta, además de ser gratuita, ofrece una accesibilidad universal al público. Se encuentra desarrollada en JavaScript sobre la plataforma Node.js, proporcionando un editor de flujo visual integrado en el navegador. Los nodos, representados por iconos, reflejan claramente su función, facilitando así su identificación. Node-RED opera de dos maneras: permite la manipulación de nodos mediante la interfaz de arrastrar y soltar para conectarlos, o la importación directa de código JavaScript para una mayor flexibilidad y personalización [14].

El nodo, en el contexto de Node-RED, despliega una variedad de funciones esenciales. Estas incluyen desde la monitorización del flujo, a través de nodos de salida de depuración, hasta la capacidad de interactuar con los pines GPIO de las Raspberry Pi, gracias al nodo específico para estas plataformas, independientemente del modelo utilizado. Los flujos creados se almacenan en formato JSON (JavaScript Object Notation). Node-RED también permite a los desarrolladores conectar nodos de entrada, salida y preprocesamiento, lo que facilita la creación de flujos para tareas como el procesamiento de datos, el control de dispositivos o el envío de alertas [20]. Todo esto se visualiza de mejor manera en la Figura 2.1

- En **Node-RED**, se encuentra tres componentes básicos:

1. Paneles de nodos
2. Panel de flujo

3. Panel de información y depuración

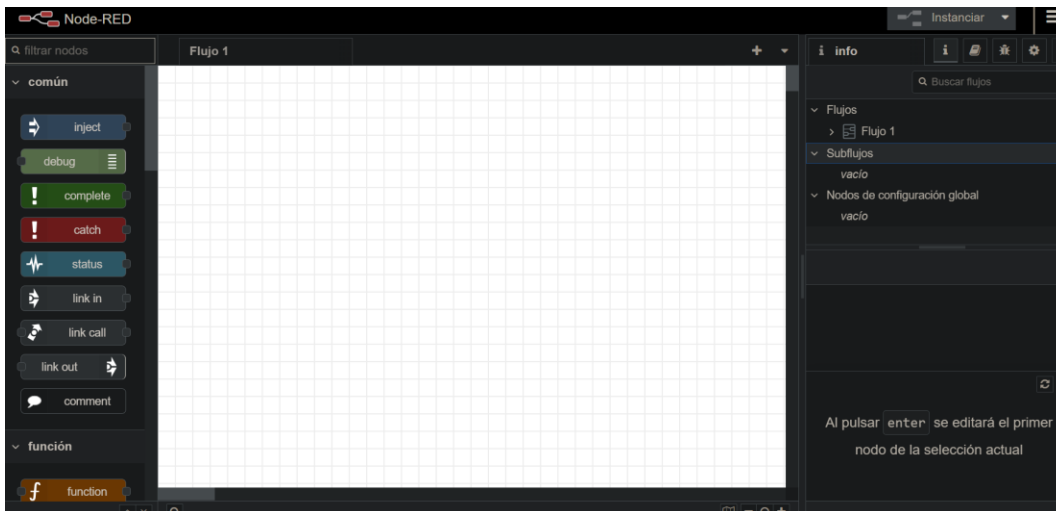


Figura 2.1: Editor de Node-RED

Como se puede notar, esta herramienta es altamente flexible y potente, lo que la convierte en una opción ideal para la creación de prototipos. Node-RED nos permite desarrollar rápidamente aplicaciones, especialmente aquellas que se activan ante eventos, como las aplicaciones de IoT. Su esencia radica en permitir que ingenieros y técnicos creen y configuren aplicaciones de manera sencilla y en tiempo real en dispositivos finales.

Instalación de Node-RED en Raspberry Pi

Como se ha mencionado anteriormente, Node-RED es una herramienta potente y útil para una variedad de casos de uso. Por lo tanto, se utilizará en el Raspberry Pi 4, que fue seleccionado para nuestro proyecto. Antes de proceder con la instalación de Node-RED, es necesario tener previamente instalado el sistema operativo en el Raspberry Pi. En nuestro caso, utilizaremos Raspbian, que es el sistema operativo propio de Raspberry Pi y cuenta con un proceso de instalación sencillo. A continuación, se detallan los pasos para la instalación de Node-RED en el Raspberry Pi.

■ **Parámetros adicionales a instalar**

El siguiente script debe ser ejecutado para asegurar que npm pueda recuperar y compilar los módulos binarios que necesiten ser instalados:

```
sudo apt install build-essential git curl
```

■ **Instalación y Actualización de Node-RED**

El script mostrado a continuación sirve para poder instalar Node.js, npm y Node-RED en la Raspberry Pi. Además de esto, el script también proporcionará una actualización a la versión más reciente disponible en caso de que se tenga previamente instalado Node-RED.

```
bash<(curl-sLhttps://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)
```

■ **Ejecutar localmente**

Una vez instalado todo lo mencionado anteriormente, se debe iniciar Node-RED con un argumento adicional debido a la limitación de memoria en Raspberry Pi. Para hacerlo, ejecuta el siguiente comando:

```
node-red -max-old-space-size=256
```

■ **Inicio automático al arrancar**

Una vez completados los pasos anteriores, es necesario configurar el inicio automático de Node-RED. Esto asegura que cada vez que se encienda el Raspberry Pi, Node-RED estará disponible automáticamente. Esto es crucial, ya que Node-RED será nuestra pasarela principal y debe iniciarse de manera consistente.

```
sudo systemctl enable nodered.service
```

En caso de que se quiera desactivar esta función, ejecute lo siguiente:


```
sudo systemctl disable nodered.service
```

■ Comandos extras para ejecutar como servicios

Además de los comandos básicos para instalar y configurar Node-RED, también hay algunos comandos útiles para administrar el servicio:

```
node-red-start
```

Inicia el servicio de Node-RED y muestra su salida de registro.

```
node-red-stop
```

Detiene el servicio de Node-RED.

```
node-red-restart
```

Detiene y reinicia el servicio de Node-RED.

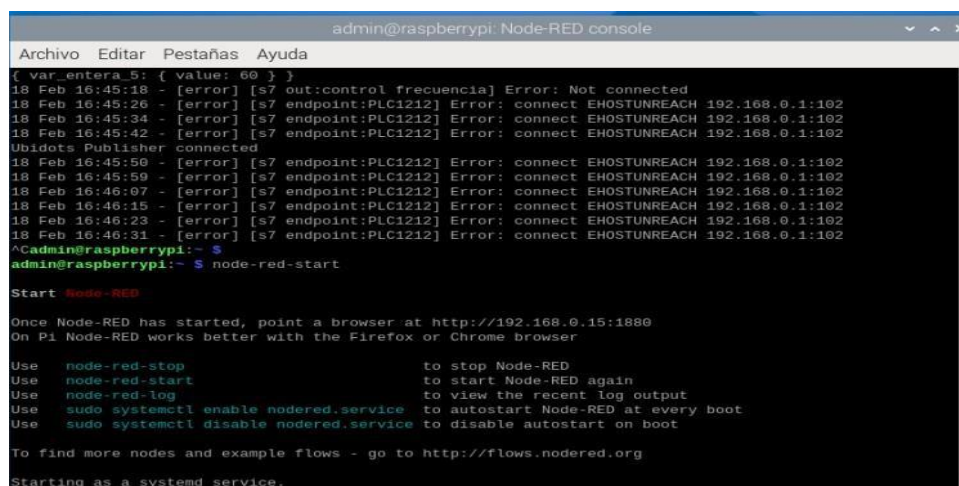
```
node-red-log
```

Muestra la salida del registro del servicio.

También se puede iniciar el servicio de Node-RED desde el escritorio realizando los siguientes pasos:

```
Menu ->Programming ->Node-Red
```

Una vez completados los pasos anteriores, el resultado será el mostrado en la **figura 2.2**



```
admin@raspberrypi: Node-RED console
Archivo Editar Pestañas Ayuda
{ var entera_5: { value: 00 } }
18 Feb 16:45:18 - [error] [s7 out:control frecuencia] Error: Not connected
18 Feb 16:45:26 - [error] [s7 endpoint:PLC1212] Error: connect EHOSTUNREACH 192.168.0.1:102
18 Feb 16:45:34 - [error] [s7 endpoint:PLC1212] Error: connect EHOSTUNREACH 192.168.0.1:102
18 Feb 16:45:42 - [error] [s7 endpoint:PLC1212] Error: connect EHOSTUNREACH 192.168.0.1:102
Ubidots Publisher connected
18 Feb 16:45:50 - [error] [s7 endpoint:PLC1212] Error: connect EHOSTUNREACH 192.168.0.1:102
18 Feb 16:45:59 - [error] [s7 endpoint:PLC1212] Error: connect EHOSTUNREACH 192.168.0.1:102
18 Feb 16:46:07 - [error] [s7 endpoint:PLC1212] Error: connect EHOSTUNREACH 192.168.0.1:102
18 Feb 16:46:15 - [error] [s7 endpoint:PLC1212] Error: connect EHOSTUNREACH 192.168.0.1:102
18 Feb 16:46:23 - [error] [s7 endpoint:PLC1212] Error: connect EHOSTUNREACH 192.168.0.1:102
18 Feb 16:46:31 - [error] [s7 endpoint:PLC1212] Error: connect EHOSTUNREACH 192.168.0.1:102
^Cadmin@raspberrypi:~$
admin@raspberrypi:~$ node-red-start
Start Node-RED
Once Node-RED has started, point a browser at http://192.168.0.15:1880
On Pi Node-RED works better with the Firefox or Chrome browser
Use node-red-stop to stop Node-RED
Use node-red-start to start Node-RED again
Use node-red-log to view the recent log output
Use sudo systemctl enable nodered.service to autostart Node-RED at every boot
Use sudo systemctl disable nodered.service to disable autostart on boot
To find more nodes and example flows - go to http://flows.nodered.org
Starting as a systemd service.
```

Figura 2.2: Boton nodos

2.3. Plataforma IoT

Las plataformas IoT son fundamentales para interconectar dispositivos, sensores y aplicaciones con otros elementos como bases de datos [21]. Estas plataformas ofrecen diversas herramientas que incluyen análisis y visualización de datos, así como características de seguridad como la autenticación de dos factores y el cifrado. En términos generales, constituyen soluciones de hardware y software que simplifican y unifican las acciones de transferencia, almacenamiento y conservación de datos.

En términos conceptuales, una plataforma IoT se concibe como un conjunto integral de tecnologías que proporciona los cimientos necesarios para el desarrollo de soluciones personalizadas. Esta infraestructura adaptable permite la configuración de características específicas en respuesta a los requisitos particulares de cada proyecto [22].

Además de ser conocidas como plataformas IoT, también se les denomina middleware IoT debido a su función mediadora entre el hardware y el software de gestión, almacenamiento y visualización. El trabajo de estas plataformas se puede dividir en varias etapas: la primera consiste en la recopilación de datos mediante sensores; la segunda se encarga de almacenar y medir, y finalmente, la última etapa implica el análisis, que puede realizarse localmente en los dispositivos o en la nube [23].

Entre las tareas clave que debe desarrollar una plataforma IoT podemos encontrar las siguientes:

- Adquisición de datos del entorno físico a través de sensores especializados.
- Análisis de datos tanto a nivel local como en entornos de computación en la nube.
- Establecimiento de conexiones con servicios en la nube para la transmisión y recepción de comandos.
- Almacenamiento de datos de manera segura y eficiente.

- Interpretación y análisis avanzado de los datos alojados en la nube, proporcionando una visión detallada de los eventos y patrones relevantes.
- Orquestación de acciones específicas en dispositivos físicos (cosas") en base a algoritmos y lógica de procesamiento.
- Presentación de información y resultados de análisis de manera clara y comprensible para los usuarios finales.

2.3.1. Elementos de una infraestructura IoT

El Internet de las Cosas (IoT) se extiende a diversos contextos, desde dispositivos portátiles hasta infraestructuras urbanas e entornos industriales, también conocidos como Industria 4.0.

Conectividad IoT

La conexión para Internet de las Cosas es el elemento fundamental que permite la comunicación entre los dispositivos en el terreno y los servidores en la nube. Los dispositivos en el terreno, también llamados nodos IoT, están encargados de recopilar los datos necesarios para su almacenamiento y análisis en la infraestructura central. Es importante destacar que la conectividad IoT podría considerarse como el pilar central de cualquier aplicación IoT que se desarrolle.

Conectores para estándares de comunicación

Los conectores para los estándares de comunicación son un elemento esencial en la estructura de niveles de aplicación en los dispositivos IoT, también denominados nodos IoT o nodos edge. Estos dispositivos están enlazados directamente a otros equipos en el campo, como sensores, actuadores u otros dispositivos industriales. Estos dispositivos industriales pueden proceder de diferentes fabricantes, cada uno con su propio hardware y firmware, y pueden emplear una amplia variedad de protocolos de comunicación. Estos protocolos pueden variar desde estándares industriales ampliamente reconocidos, como Modbus, incluso incluyendo protocolos

más concretos o de propiedad exclusiva. Esto hace que el papel de los conectores sea fundamental para recopilar datos de manera efectiva de los equipos.

Plataformas IoT

En el panorama actual, el mercado de plataformas IoT ofrece una amplia gama de soluciones disponibles para su implementación en diversos ámbitos. Estas plataformas, diseñadas para facilitar la conectividad y gestión de dispositivos en el Internet de las Cosas (IoT), ofrecen un conjunto diverso de funciones y características, adaptadas a las necesidades específicas de los usuarios.

A continuación, se presenta la tabla comparativa 2.3 que analiza algunas de las plataformas IoT disponibles en el mercado actualmente. Esta comparación se centra en las funcionalidades de cada una de las plataformas.

Plataforma	Fractal	ThingSpeak	Twilio	Ubidots STEM
Cantidad total de características	131	25	79	97
API	✓	✓	✓	✓
Acceso móvil	✓			✓
Funciones activadas por eventos	✓		✓	✓
Medidas preventivas y correctivas (CAPA)				
Actualizaciones instantáneas				✓
Agregación y publicación de datos	✓			✓
Alertas de móvil	✓		✓	
Notificaciones y aumento de la gravedad de los problemas			✓	✓
Notificaciones	✓		✓	✓
Almacén de archivo				

Tabla 2.3: Funcionalidades de las plataformas

2.4. Integración de Raspberry Pi y el PLC

Una vez que el Raspberry Pi y Node-RED están configurados para la transmisión y recepción de datos de manera óptima, el paso crucial que sigue es asegurar la configuración precisa de los datos que serán transmitidos y recibidos a mediante la pasarela inteligente. Esta fase implica una meticulosa configuración de los parámetros de transmisión y recepción en el PLC, esto se realizará utilizando la plataforma de ingeniería TIA Portal V17.

En esta etapa, se ejecutará un proceso meticuloso y detallado, donde se definirán los protocolos de comunicación, se establecerán los formatos de los datos a transmitir y a recibir, y se configurarán los puntos de acceso a la información relevante.

2.4.1. Arquitectura de la comunicación entre el PLC y el Raspberry Pi

La interacción planificada entre el PLC y el Raspberry Pi, representada en la Figura 2.3, garantiza una transferencia precisa de datos en tiempo real. Para lograrlo, se asegura una comunicación fluida entre ambos dispositivos, lo que implica identificar y asignar variables, configurar los parámetros de conexión y validar exhaustivamente la configuración para garantizar su correcto funcionamiento en el entorno operativo deseado.

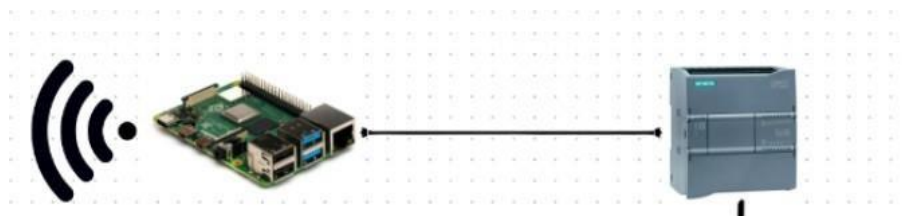


Figura 2.3: Arquitectura de comunicación entre el PLC y el Raspberry Pi

2.4.2. Configuración de la Comunicación Bidireccional en el PLC.

En este apartado se presenta un desglose exhaustivo de los ajustes necesarios, acompañado de explicaciones que cubren cada aspecto crítico para garantizar una comunicación óptima:

Para lograr la comunicación entre el PLC y la nube se va utilizar la herramienta Node-Red la cual cuenta con distintos tipos de nodos los cuales, tienen distintas funcionalidades. En la figura 2.4 se puede ver el código elaborado para el intercambio de datos y sus respectivos nodos utilizados.

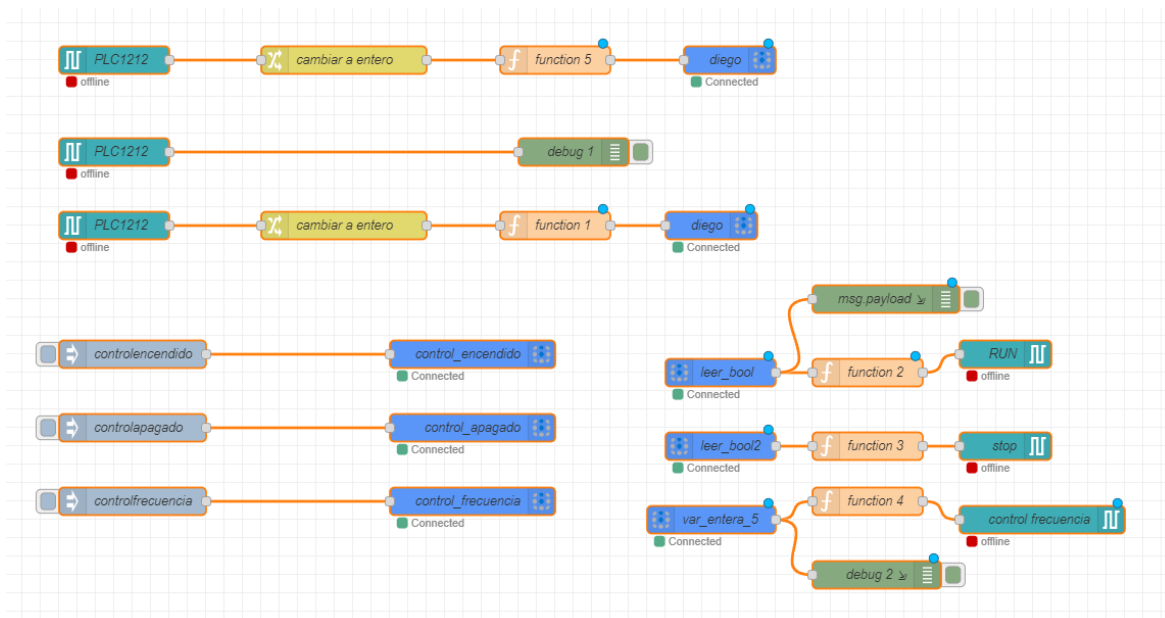


Figura 2.4: Nodos Programados para el intercambio de datos entre PLC y la nube

Explicación de los Nodos

El nodo inject (Figura 2.5) en Node-RED es comúnmente utilizado para iniciar un flujo de mensajes manualmente o automáticamente en un horario o intervalo especificado. Es uno de los nodos más fundamentales en Node-RED y sirve varios propósitos:

Disparador Manual: Puede configurarse para que cuando se haga clic en el botón del nodo (la pequeña caja al lado izquierdo del nodo en el editor de flujos), se envíe un mensaje. Esto es útil durante la fase de desarrollo y prueba para simular la entrada de datos.

Temporalizador: Puede configurarse para enviar mensajes en intervalos regulares (cada segundo, cada hora, diariamente, etc.) o en momentos específicos del día. Esto es útil para tareas de automatización que deben ejecutarse periódicamente.

Carga Útil Personalizada: Se puede configurar el nodo para enviar una carga útil específica, como un número, un string, un booleano, un objeto JSON o incluso una

cadena vacía. Esto puede actuar como un valor constante o como datos de prueba para otros nodos en el flujo.

Inicio Automático: Al desplegar un flujo, el nodo inject puede configurarse para enviar automáticamente un mensaje. Esto es útil para inicializar el estado de un flujo o para activar procesos inmediatamente después del despliegue.



Figura 2.5: Nodo Inject

El nodo Change (Figura 2.6) es una herramienta esencial en Node-RED para manipular datos dentro de un flujo, permitiendo a los usuarios adaptar los mensajes a sus necesidades específicas sin necesidad de escribir código personalizado. Su capacidad para modificar dinámicamente las propiedades del mensaje lo hace indispensable para una amplia gama de aplicaciones en la automatización de procesos, IoT, y más.



Figura 2.6: Nodo Change

El nodo function (Figura 2.7) brinda a los usuarios la capacidad de extender la funcionalidad de sus flujos de trabajo con lógica personalizada. Su flexibilidad y poder, hacen posible la implementación de casi cualquier funcionalidad que se pueda necesitar en los proyectos de automatización con IoT, o cualquier otro escenario donde Node-RED sea aplicable.



Figura 2.7: Nodo Function

El nodo Ubidots Out (Figura 2.8) recibe un mensaje y utiliza sus valores para publicar una solicitud a la API de Ubidots mediante MQTT.



Figura 2.8: Nodo Para enviar datos a Ubidots

El Nodo Ubidots In (Figura 2.9) se suscribe hasta 10 variables o temas de Ubidots y los envía como un mensaje a otros nodos. Utiliza la biblioteca MQTT para establecer la conexión y suscribirse a la Variable o Tema.

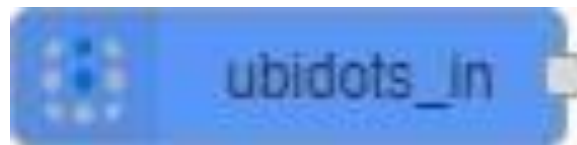


Figura 2.9: Nodo Para leer datos de Ubidots

El Nodo S7 In (Figura 2.10) lee datos de un PLC S7. Este nodo fue creado como parte del proyecto ST-One. Todos los datos se leen cíclicamente del PLC según se configura en el punto final s7, pero existen tres modos de hacerlos disponibles en un flujo:



Figura 2.10: Nodo para extraer datos del PLC

El Nodo S7 Out (Figura 2.11) escribe en un área de memoria del PLC. Escribe datos en el PLC en la variable especificada en la configuración del nodo o en msg.variable.



Figura 2.11: Nodo para escribir en el PLC

Para establecer una comunicación con TIA Portal, empleamos el nodo S7 Comms (Figura 2.12), que se compone de dos elementos fundamentales: uno para la operación PUT y otro para GET. Estos bloques son esenciales para la escritura y lectura de datos, respectivamente. Este enfoque permite una interacción eficiente y efectiva con TIA Portal, facilitando el intercambio de información entre el sistema de control y la interfaz de programación, garantizando así un flujo de trabajo optimizado y una integración sistemática de las operaciones de automatización.

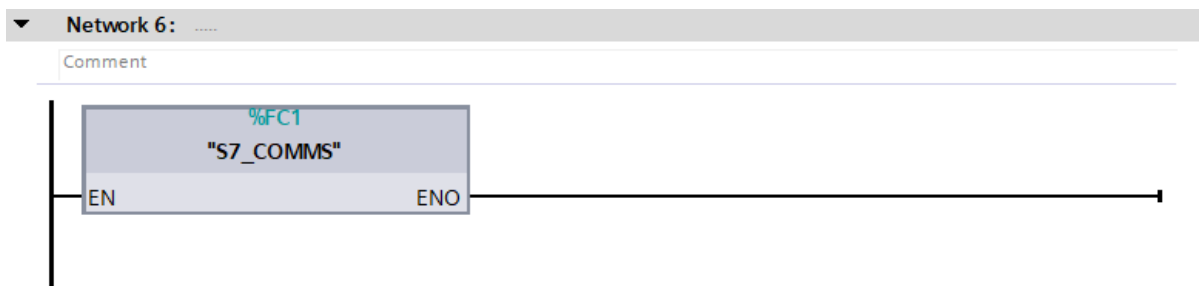


Figura 2.12: Bloque S7 Comms

El Bloque S7 Comms contiene los bloques Put y Get para la transmisión y recepción de datos (Figura 2.13). El funcionamiento de estos bloques consiste en un array de datos los cuales se irán guardando cíclicamente según el PLC vaya adquiriendo los datos. La estructura de este array se muestra en la Figura 2.14

GET_DATA									
Name	Data type	Offset	Start value	Retain	Accessible f...	Write...	Visible in...	Sequence	Comment
1	Symbolic								
2	Array of 16 Int	0.0							
3	RECIBE_datos[0]	Int	0.0						
4	RECIBE_datos[1]	Int	2.0						
5	RECIBE_datos[2]	Int	4.0						
6	RECIBE_datos[3]	Int	6.0						
7	RECIBE_datos[4]	Int	8.0						
8	RECIBE_datos[5]	Int	10.0						
9	RECIBE_datos[6]	Int	12.0						
10	RECIBE_datos[7]	Int	14.0						
11	RECIBE_datos[8]	Int	16.0						
12	RECIBE_datos[9]	Int	18.0						

Figura 2.14: Array De datos

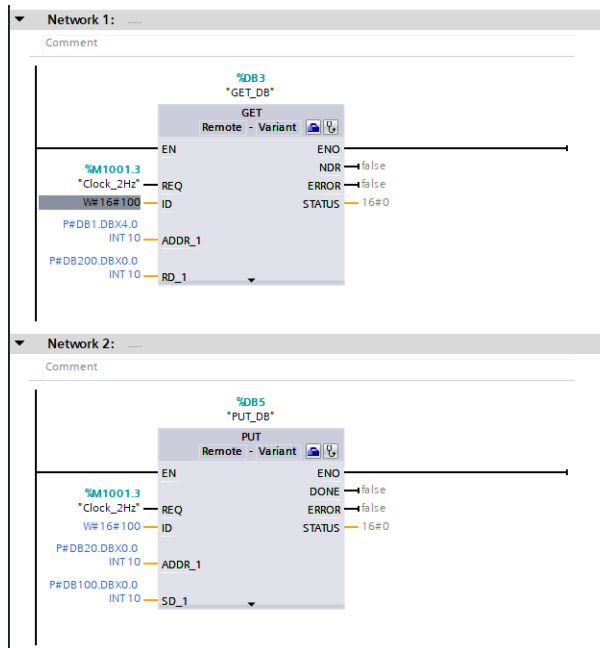


Figura 2.13: Bloques Put y Get del bloque S7-Comms

Capítulo 3

Implementación del sistema de monitoreo y control remoto

En el presente capítulo se lleva a cabo la implementación del sistema destinado al monitoreo y control remoto del Variador de Frecuencia Delta. Este proceso se fundamenta en las configuraciones previamente establecidas en los capítulos 1 y 2, donde se detallaron las especificaciones y pasos necesarios para su realización progresiva. El propósito es explicar de manera rigurosa cómo se puede efectuar el control y monitoreo del variador de frecuencia. En este capítulo, se abordará la creación y configuración del panel de control de la plataforma IoT, seleccionada por sus atributos ventajosos, como es el caso de Ubidots. Además, se examinará cómo se pueden recibir y enviar datos a través de la pasarela inteligente.

3.1. Arquitectura de del sistema de monitoreo y control remoto

La topología para la comunicación que se pretende establecer se ilustra en la figura 3.1. En esta configuración, se emplea una pasarela inteligente para habilitar la conexión del PLC a internet, permitiendo así la transmisión y recepción de instrucciones desde la plataforma IoT. Este enfoque posibilita la verificación de la frecuencia de salida del variador de frecuencia, así como el control remoto del motor

asociado al variador. Con este sistema, se logra la capacidad de gestionar la velocidad del motor y de encenderlo o apagarlo de manera remota.

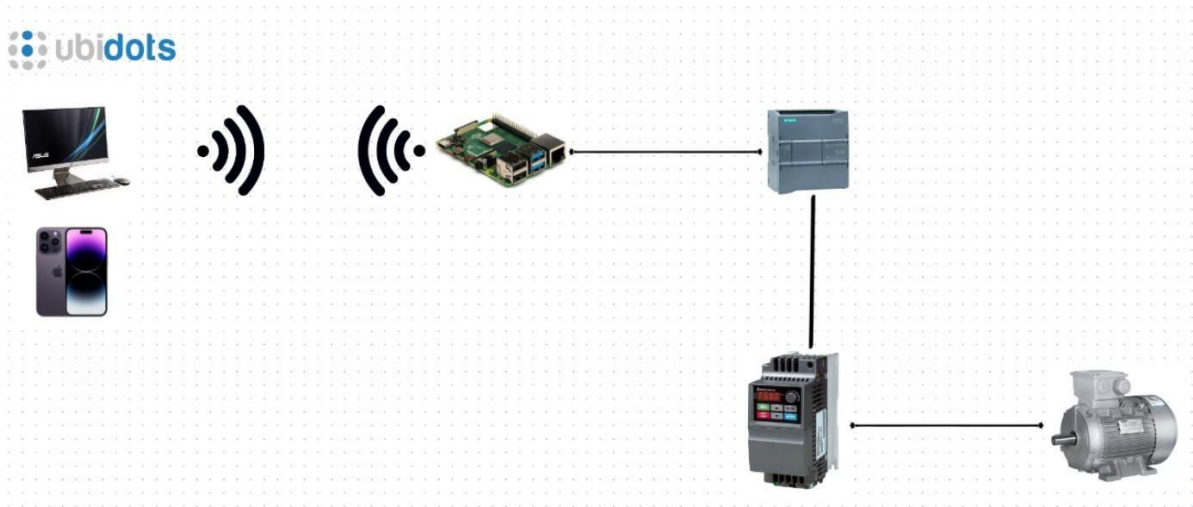


Figura 3.1: Arquitectura de del sistema de monitoreo y control remoto

3.2. Configuración y programación del dashboard de Ubidots

Para iniciar la configuración de la plataforma IoT, el primer paso es crear una cuenta en la misma. En caso de no disponer de una cuenta previa, el acceso a la plataforma en cuestión puede observarse en la Figura 3.2. El enlace para acceder a Ubidots es el siguiente:

<https://stem.ubidots.com/accounts/signin/>

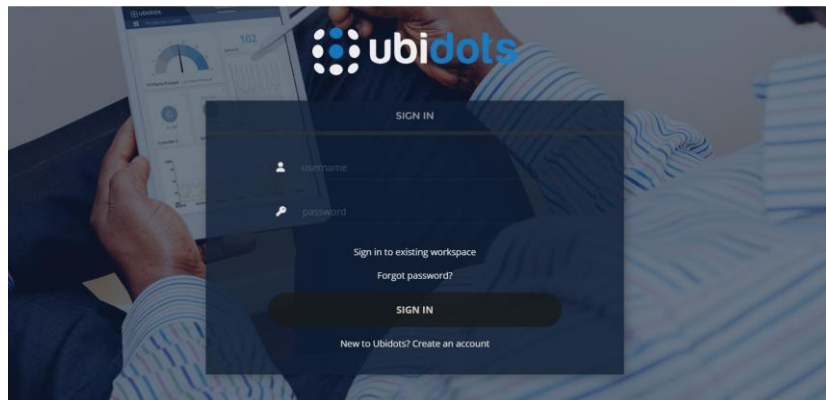


Figura 3.2: Plataforma de inicio de Ubidots

Una vez accedido a la plataforma, el siguiente paso es crear el dispositivo correspondiente. Para ello, seleccionamos la opción **Devices** y hacemos clic en **Create a new device**, tal como se ilustra en la Figura 3.3.

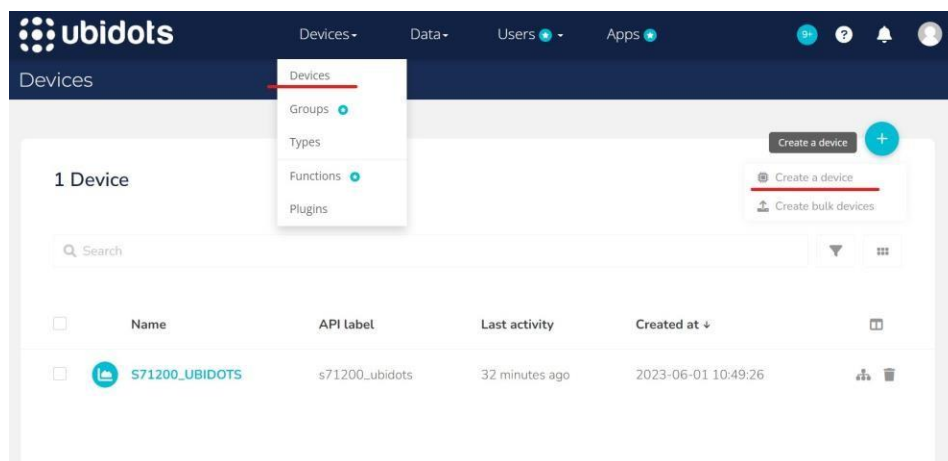


Figura 3.3: Devices Ubidots

Una vez creado un nuevo dispositivo, se obtendrá la información detallada que se muestra en la Figura 3.4. Los elementos más relevantes y que serán utilizados posteriormente están resaltados en rojo en la imagen. Estos elementos a tener en cuenta son: API Label, ID y Token.

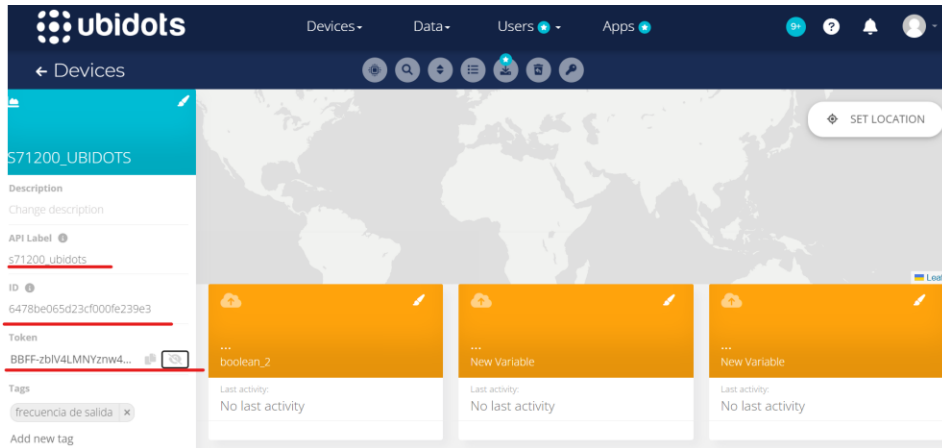


Figura 3.4: Devices Ubidots

Después de esto, se procede a crear el panel de control (**dashboard**) para recibir y enviar los datos necesarios. Para ello, accedemos a la sección **Data** donde se desplegarán diversas opciones, entre las cuales seleccionamos **Dashboard** y se hará clic en ella, tal como se muestra en la Figura 3.5. Una vez hecho esto, tendremos el **Dashboard** en donde se podrá agregar los **widget** necesarios. Para esto se dará click en el signo de **+** que está a la derecha de la ventana.

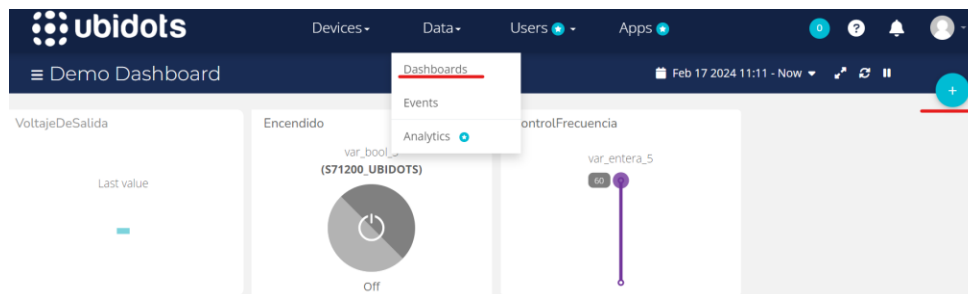


Figura 3.5: Dashboards

Una vez se haga clic en el botón **+**, se presentará una serie de widgets como los de la imagen 3.6, de entre los cuales seleccionaremos aquellos que resulten pertinentes para nuestro proyecto. La selección de los widgets específicos a utilizar se detallará en las siguientes secciones de este capítulo.

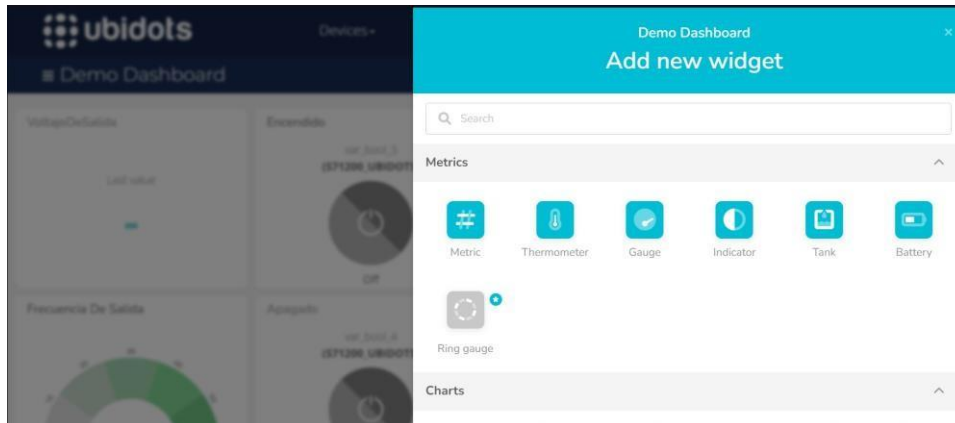


Figura 3.6: Widgets

3.3. Envío y recepción de datos en Node-Red

Una vez completadas las respectivas configuraciones en el Raspberry Pi, como se detalla en la sección 2, se accederá al siguiente entorno de trabajo 3.7. Aquí, las principales bibliotecas a utilizar son **S7-in**, **S7-out** y **S7-control**, tal como se indica en la imagen 3.7.

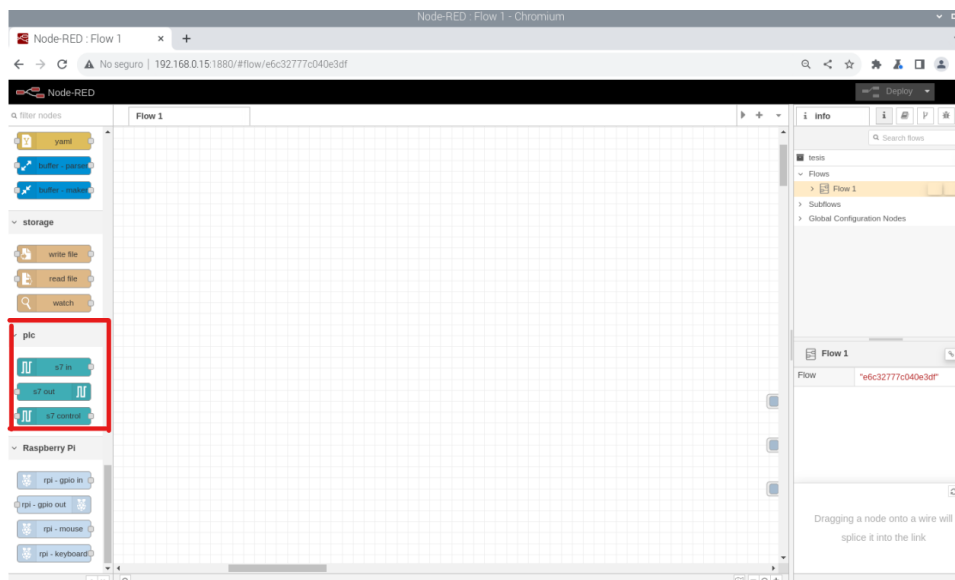


Figura 3.7: S7 Node

Dado que se está utilizando Ubidots, también se necesita de la librería de Ubidots. En caso de no tenerla, se puede descargarla como se muestra en la imagen 3.8. Además, si falta alguna otra librería con la que se desea trabajar, puede adquirirla

seleccionando las tres líneas horizontales ubicadas en la esquina derecha, como se ilustra en la imagen 3.8.

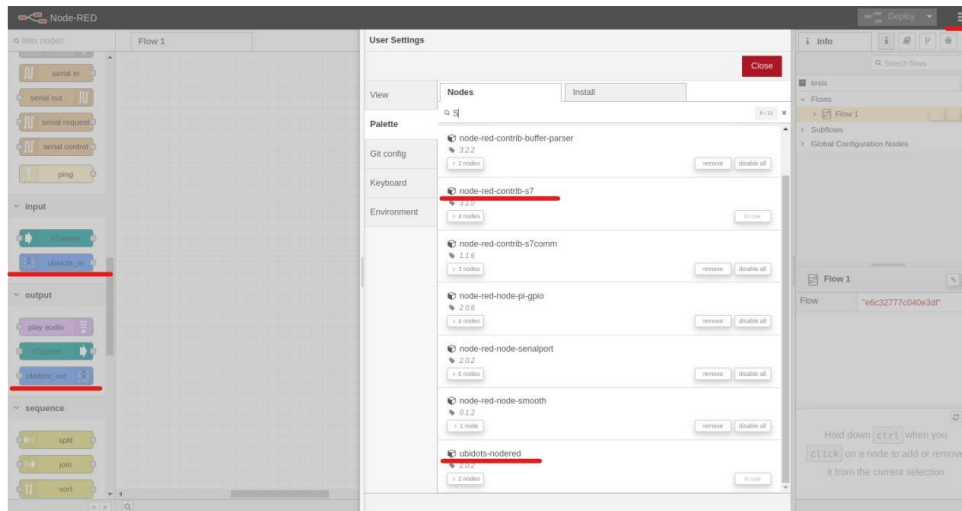


Figura 3.8: Node-Ubidtos

Para iniciar se utilizan los nodos, en este caso el nodo de PLC S7, para poder recibir los datos enviados por el PLC. Una vez seleccionado el nodo, aparecerá la siguiente ventana 3.9 en la que debemos configurar primero la dirección IP del PLC S7-1200 para establecer una conexión (Figuras 3.10 y 3.11).

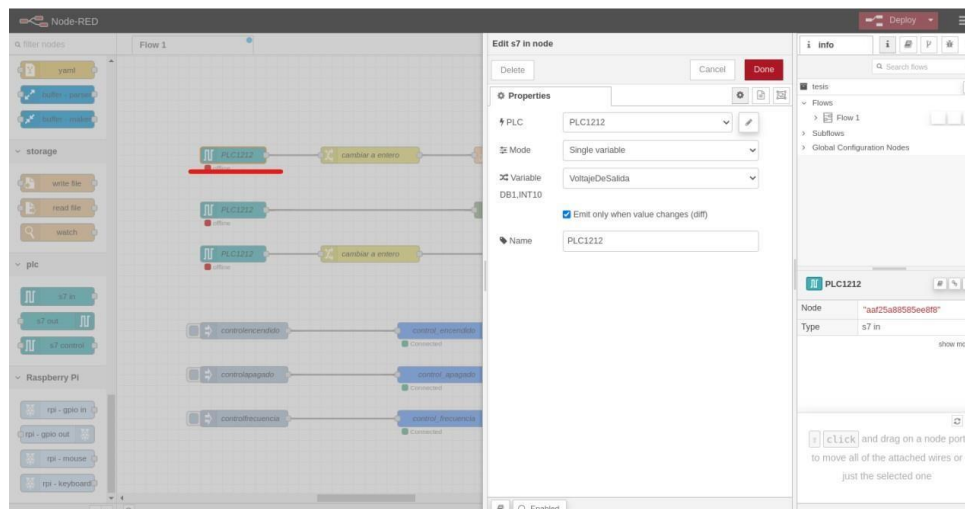


Figura 3.9: PLC Node

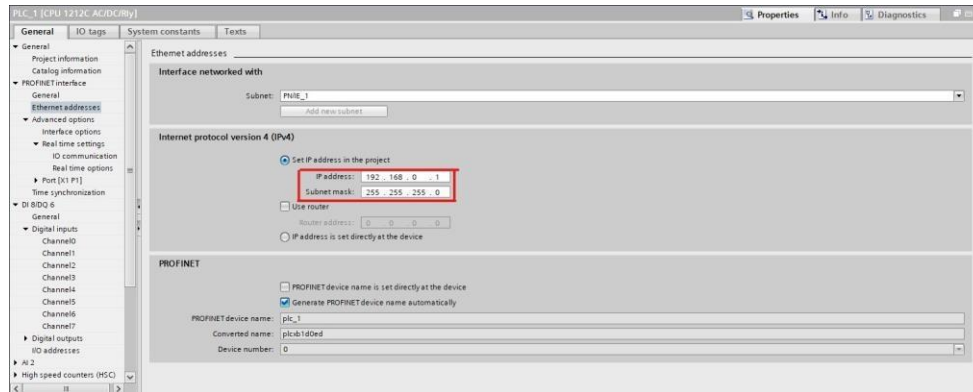


Figura 3.10: Ip PLC

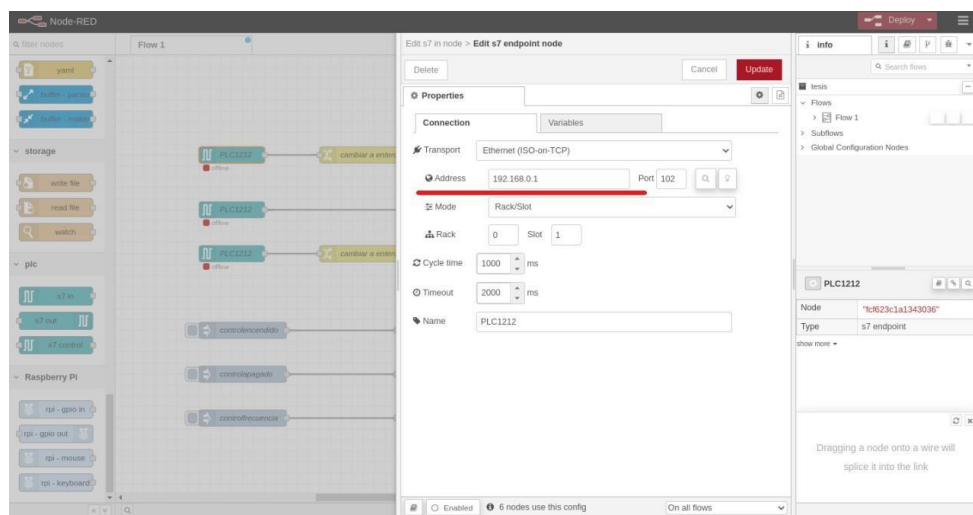


Figura 3.11: Configuración Ip del nodo

Por otro lado, si deseamos leer las variables que tenemos en el programa de TIA Portal (Figura 3.12) para realizar el análisis del variador de frecuencia, se puede utilizar los nodos que se muestran en la Figura 3.13. Una vez tengamos estas variables en Node-RED, el siguiente paso es convertirlas al lenguaje de JavaScript mediante las funciones que se muestran en la Figura 3.14 para que los nodos de Ubidots puedan interpretarlas.

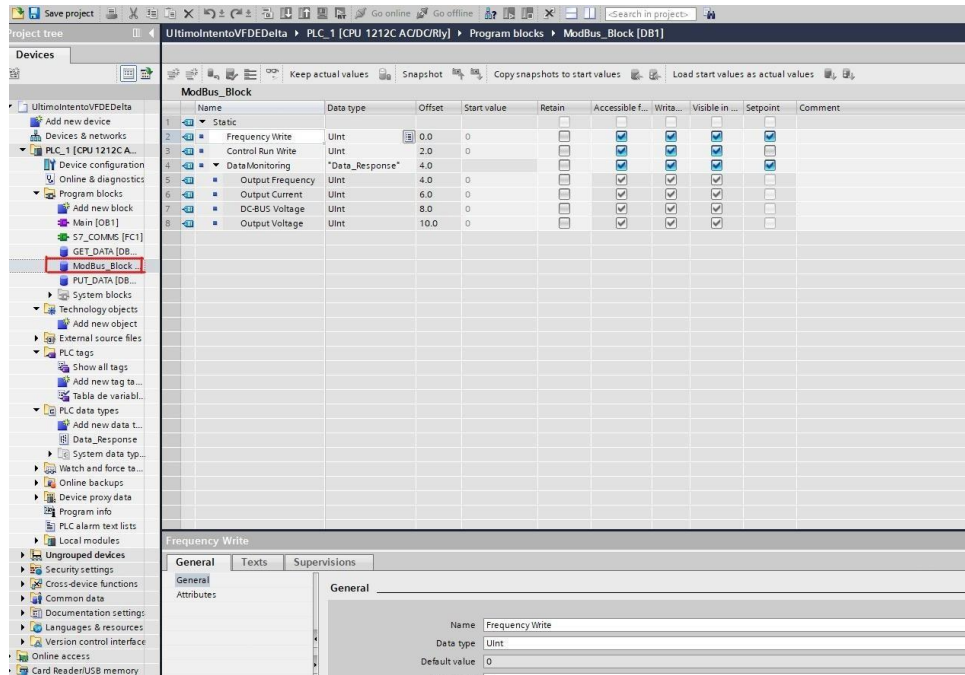


Figura 3.12: Datos PLC S7

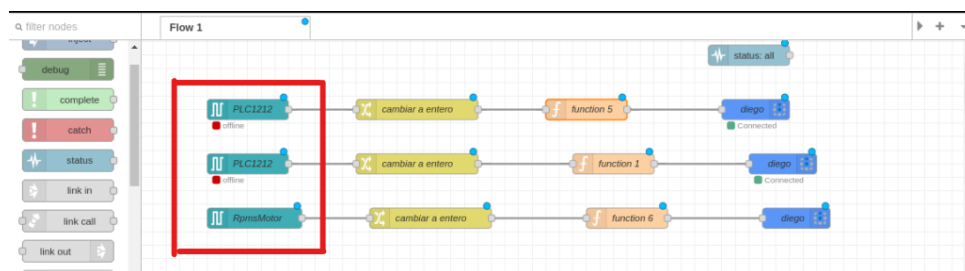


Figura 3.13: Nodos PLC S7

Una vez que los datos han sido convertidos, el siguiente paso es enviarlos a la plataforma de Ubidots utilizando los nodos específicos de esta plataforma (Figura 3.14).

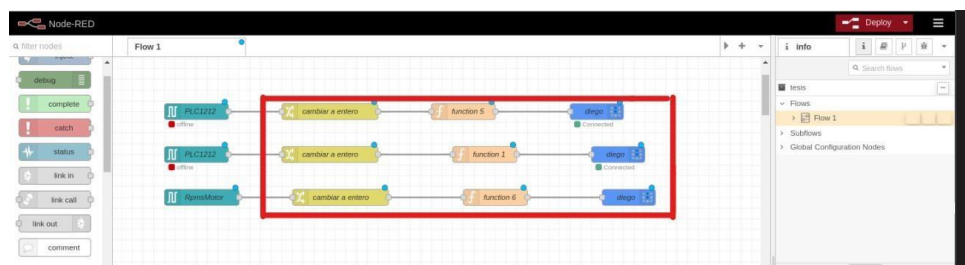


Figura 3.14: Conversion de datos

Después de este proceso, para visualizar las variables en Ubidots, necesitamos

crear las variables correspondientes en la plataforma. Una vez creadas estas variables, podremos visualizarlas tal como se muestra en la Figura 3.15.

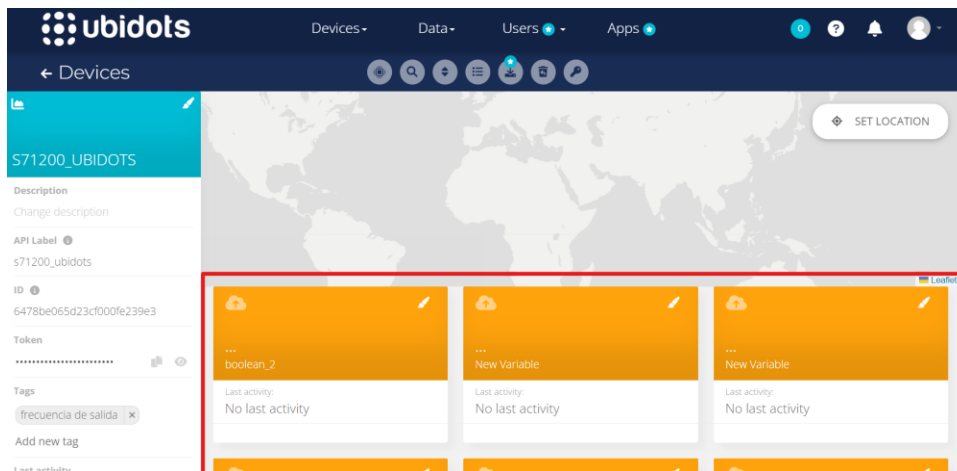


Figura 3.15: Variables en Ubidots

Para poder tener un control desde Ubidots y así poder accionar algo en el PLC, se crean las variables correspondientes en Node-RED. Las variables que han sido creadas se muestran en la Figura 3.16:

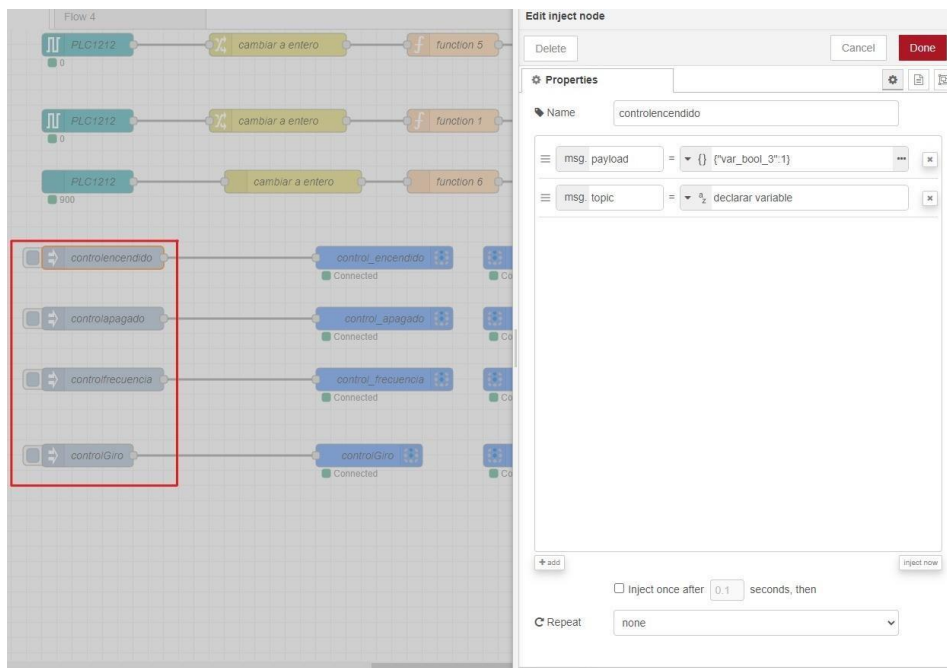


Figura 3.16: Variables para recibir datos desde Ubidots

Después de este proceso, para que se pueda comunicar con Ubidots, simplemente se debe utilizar los nodos de Ubidots y realizar la configuración

correspondiente, como se muestra en la Figura 3.17. Además, es necesario utilizar el token de Ubidots, el cual se puede obtener desde la plataforma, como se muestra en la Figura 3.4.

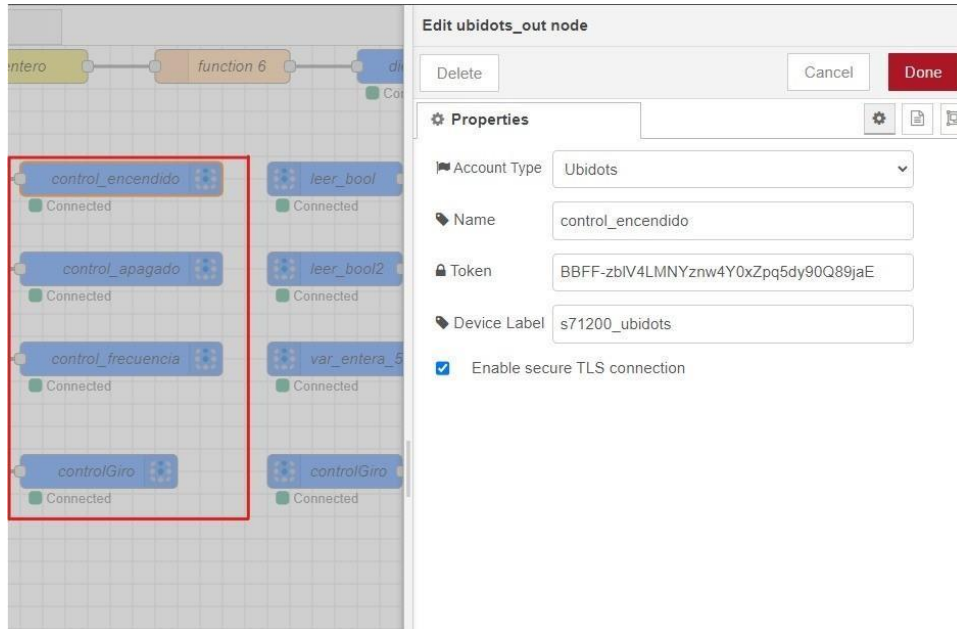


Figura 3.17: Nodos Ubidots

Cuando se accionen los botones asociados a las variables 3.18, estas últimas se generarán de manera automática en nuestra plataforma de Ubidots 3.19.

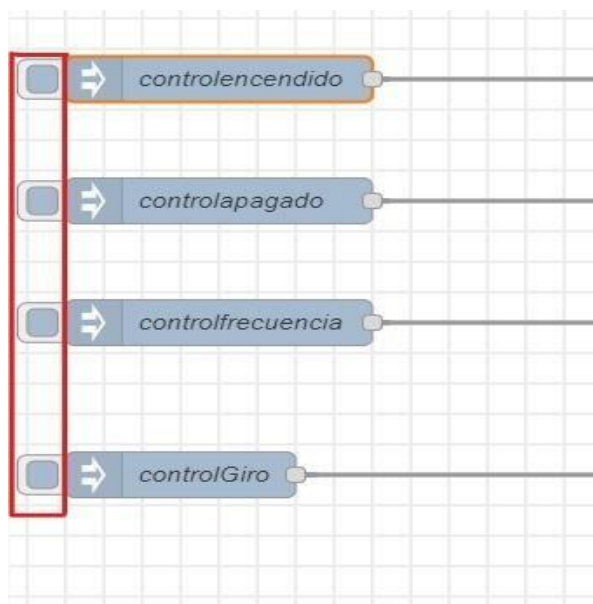


Figura 3.18: Boton nodos

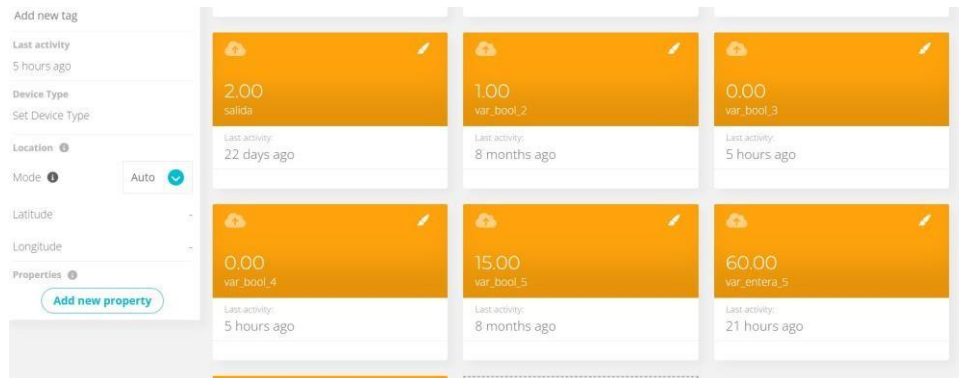


Figura 3.19: Variables en Ubidots

Para utilizar las variables recién creadas, se debe emplear el nodo **ubidots_in** (Figura 3.20), el cual facilitará la importación de dichas variables. Es crucial asegurarse de que coincidan el etiquetado de la variable, el dispositivo y el token con los de Ubidots, para garantizar un intercambio de datos sin errores.

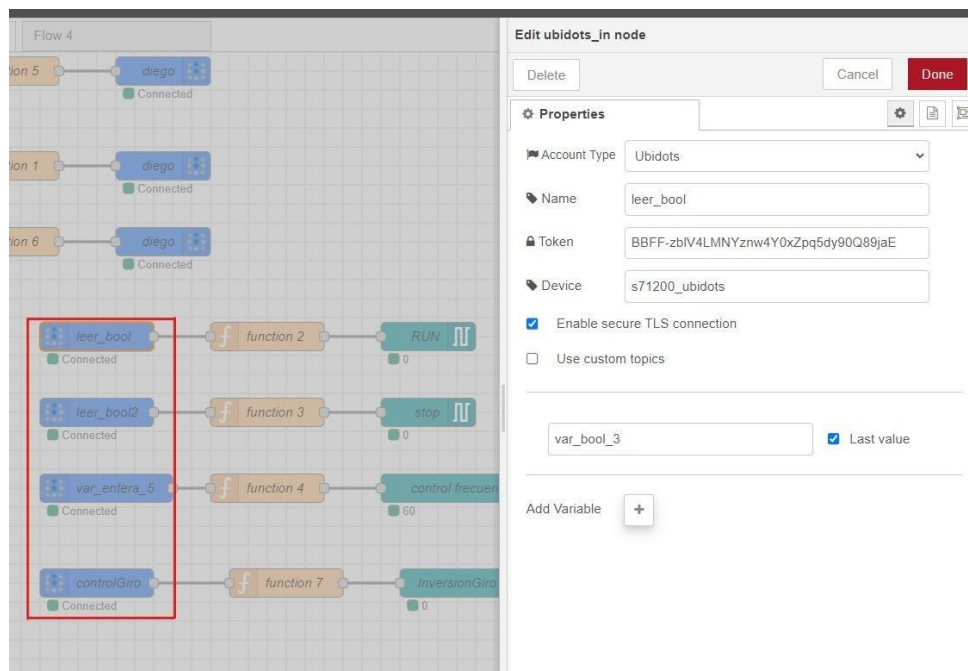


Figura 3.20: Nodo Ubidots In

Configurados los nodos de Ubidots, se pasa a configurar los de Snap 7, los cuales se encargan de enviar los valores al PLC en el TIA Portal (Figura 3.21).

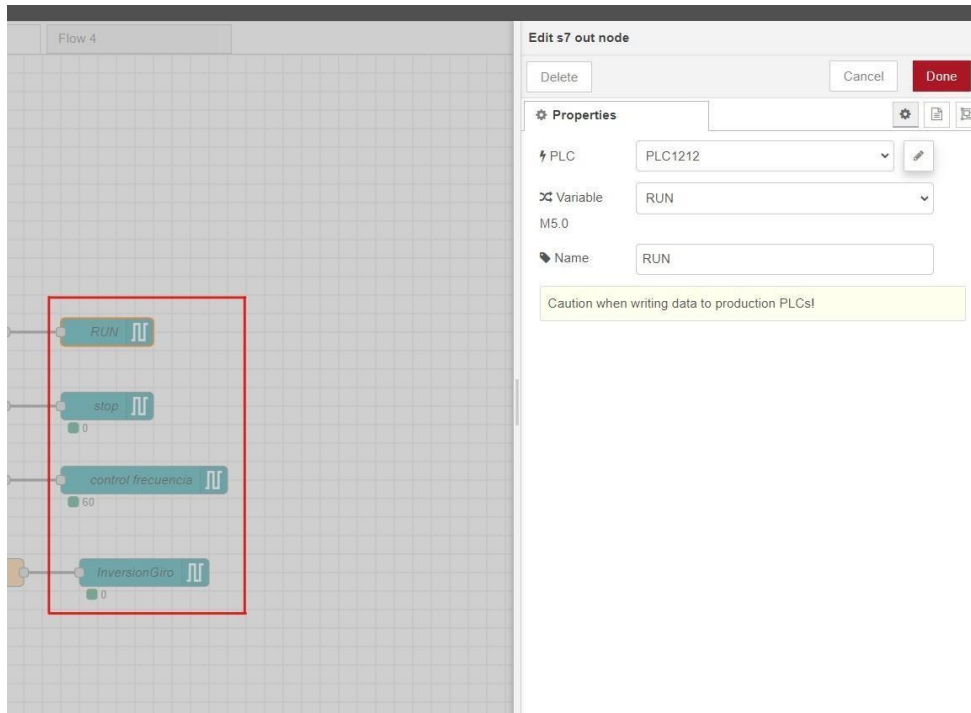


Figura 3.21: Nodos PLC

Lo único que queda es que la información que envía el nodo de Ubidots pueda ser interpretada por el nodo de Snap 7 del PLC, para que así se puedan enviar los datos de manera eficiente y sin pérdida de información. Para lograr esto, se realiza una conversión utilizando el nodo de función (Figura 3.22), donde la variable leída de Ubidots en formato JSON se convierte a un número, que es el valor que interpretará el nodo de Snap 7.

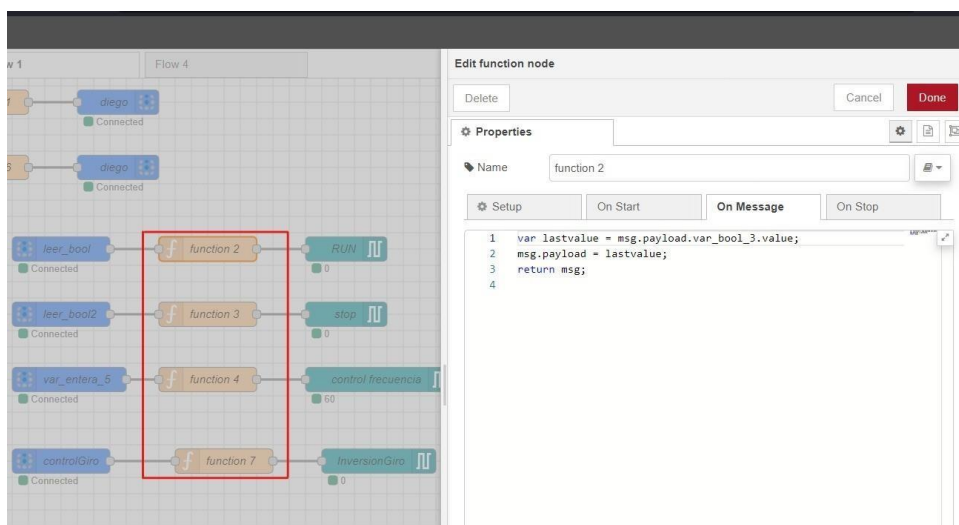


Figura 3.22: Nodos Función

Entonces, una vez completado todo este proceso, se tendrá establecida la comunicación entre Ubidots y el programa en TIA Portal, lo que implica que el PLC estará conectado a Ubidots. Ahora, se dirige al dashboard, donde se utiliza algunos widgets para controlar las variables creadas anteriormente. Con esto, se logra controlar el PLC de forma remota. El aspecto final del dashboard es como se muestra en la figura 3.23.

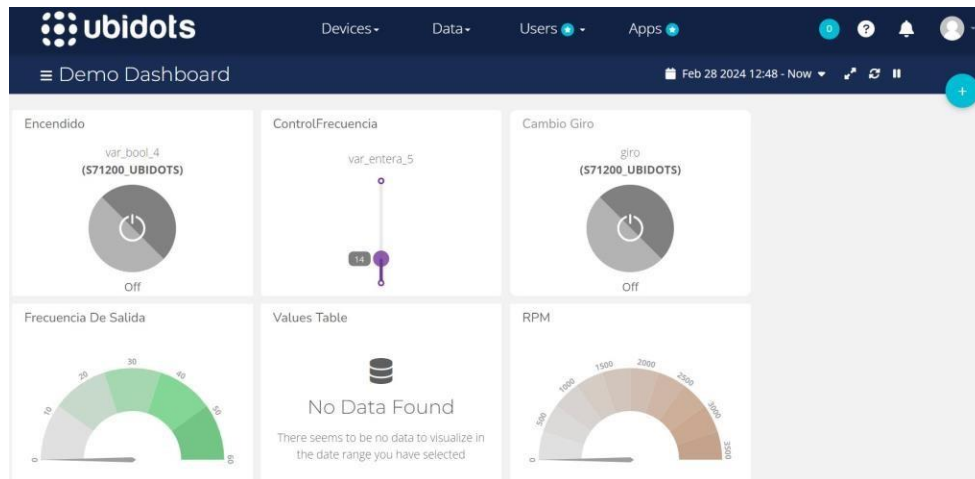


Figura 3.23: Dashboard Ubidots

3.4. Pruebas y resultados

Finalmente, después de realizar todas las configuraciones pertinentes y organizar todos los bloques para que tuvieran la conexión adecuada, procedimos a realizar las respectivas pruebas para verificar el funcionamiento del proyecto. En primer lugar, llevamos a cabo las pruebas para asegurarnos de que el botón de encendido y apagado estuviera enviando los datos hacia Node-RED. El resultado de estas pruebas se muestra en la Figura 3.24.

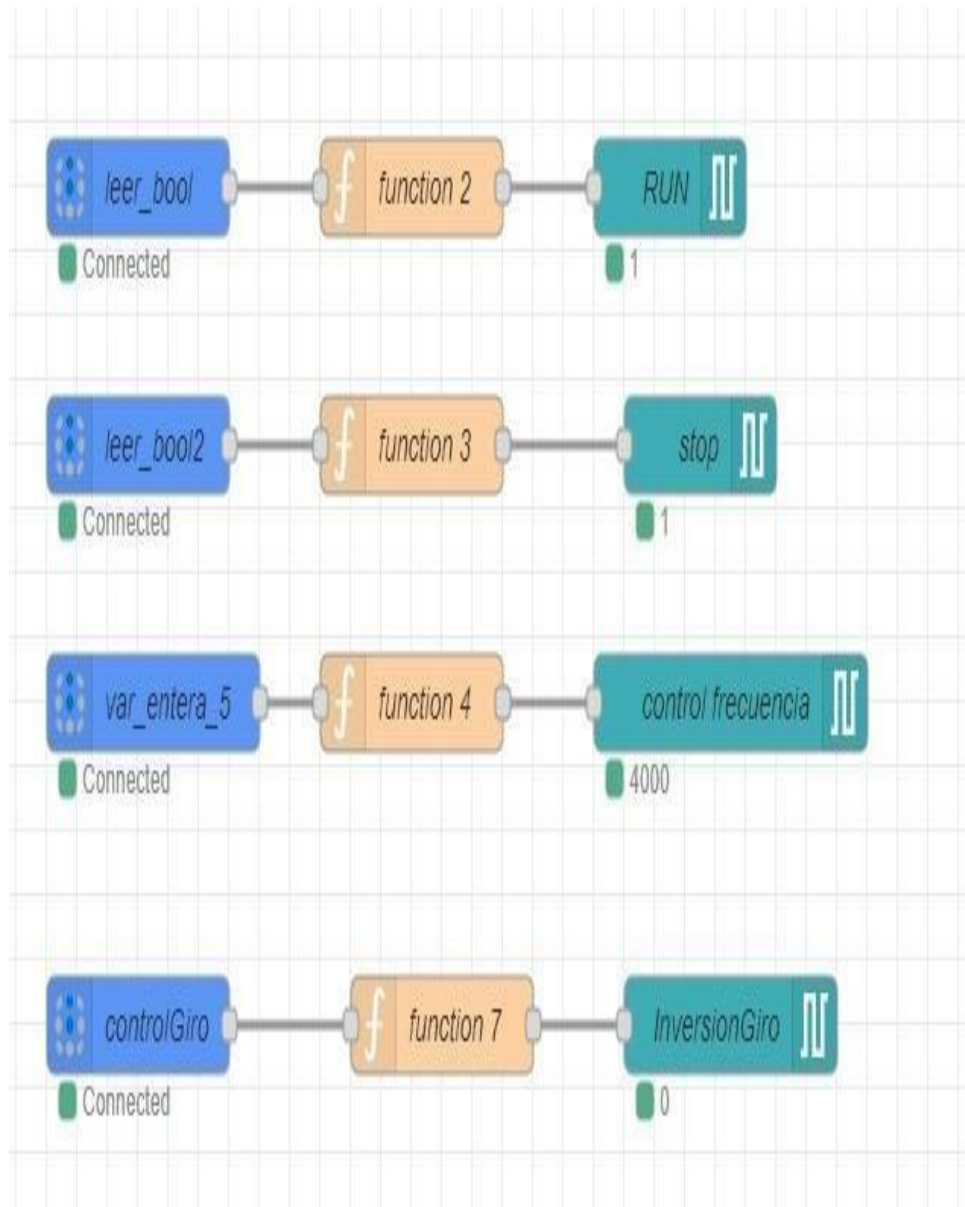


Figura 3.24: Datos Node-RED

Como se puede observar, se están recibiendo los datos pertinentes en Node-RED. Posteriormente, se verifica esta funcionalidad de manera práctica para determinar si el motor puede encenderse o apagarse según sea necesario. Además, se puede verificar claramente en la pantalla del variador de frecuencia si esta en modo RUN o STOP 3.25. Además constatando también con el proceso de Ubidots como se puede ver en la imagen 3.26.



Figura 3.25: RUN VFD



Figura 3.26: RUN en Ubidots

Una vez confirmada la capacidad de encendido del motor a través de la página web y verificada mediante el variador de frecuencia, se procedió a realizar la prueba correspondiente para confirmar su capacidad de apagado, como se puede observar en la imagen 3.27. Además, se llevó a cabo la verificación respectiva utilizando el panel de control de Ubidots, tal como se muestra en la imagen 3.28.



Figura 3.27: STOP VFD



Figura 3.28: STOP en Ubidots

Por otro lado, el motor utilizado es el que se muestra en la figura 3.29.



Figura 3.29: Motor puesto a prueba

Después de verificar la capacidad tanto de encendido como de apagado del motor, se puso a prueba su capacidad de cambio de dirección. Durante esta prueba, también observamos cómo los parámetros del variador de frecuencia cambiaban según lo ajustado en la plataforma de IoT. Las imágenes respectivas del variador de frecuencia se muestran en la 3.30 y en la plataforma IoT en la 3.31.



Figura 3.30: Ubidots giro horario



Figura 3.31: VFD giro horario

Luego se probó si el cambio de dirección responde como se espera, por lo que se realizó la acción correspondiente en el botón de cambio de dirección en Ubidots, como se muestra en la figura 3.32. Se verificó que el cambio se realizara correctamente en el variador de frecuencia, como se puede observar en la figura 3.33.

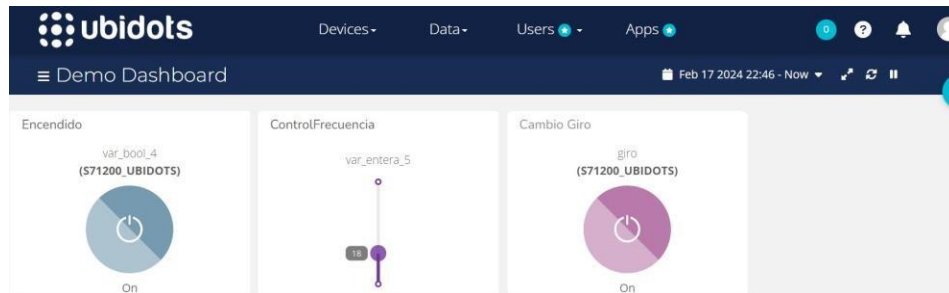


Figura 3.32: Ubidots giro Antihorario



Figura 3.33: VFD giro Antihorario

Finalmente, para el control del motor, verificamos que se pudiera variar la velocidad del mismo mediante la variación de frecuencia realizada en la plataforma. Además, al igual que en las partes anteriores, pudimos confirmar que la frecuencia se estaba variando correctamente tanto en la pantalla del variador de frecuencia como en la plataforma de IoT estos resultados se reflejan en las imágenes 3.34 y 3.35.

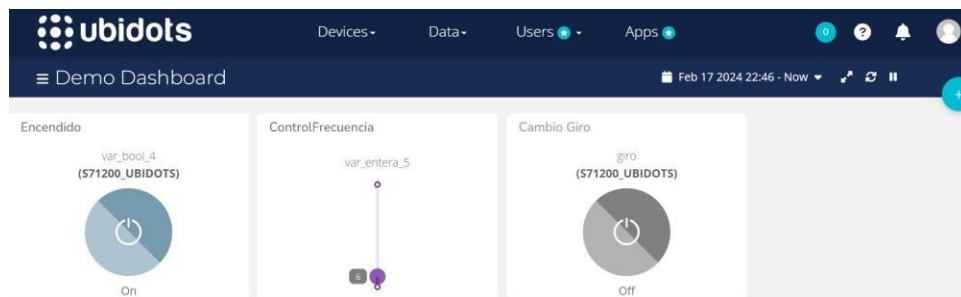


Figura 3.34: Ubidots Frecuencia 6Hz



Figura 3.35: VDF Frecuencia 6Hz

Para verificar la capacidad de variación a diferentes valores de frecuencia, se probaron distintos valores. Las figuras 3.36 y 3.37 muestran la variación a 24 Hz.



Figura 3.36: Ubidots Frecuencia 24Hz



Figura 3.37: VFD Frecuencia 24Hz

Finalmente, se realizó la prueba con la máxima frecuencia de operación, que es de 60 Hz. Dicha comprobación está documentada en las figuras 3.38 y 3.39.

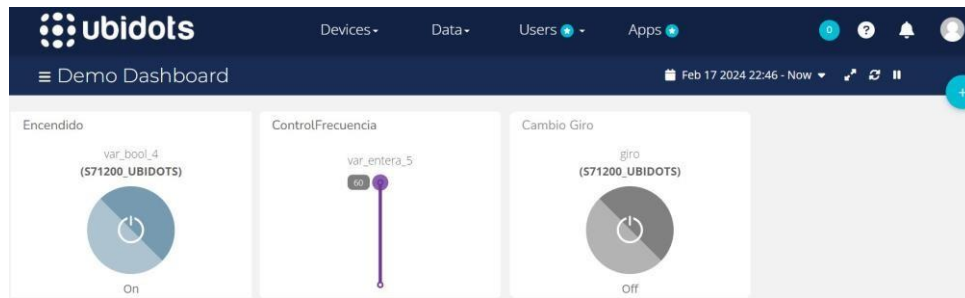


Figura 3.38: Ubidots Frecuencia 60Hz



Figura 3.39: VFD Frecuencia 60Hz

Una vez comprobado que se puede controlar tanto el variador de frecuencia como el motor conectado a él, que es el objetivo final de nuestro proyecto, pasamos a la comprobación del siguiente paso. Este consiste en verificar cuál es la frecuencia de salida que nos está proporcionando el variador de frecuencia. Dichos valores podrán monitorearse como se ve en la 3.40.

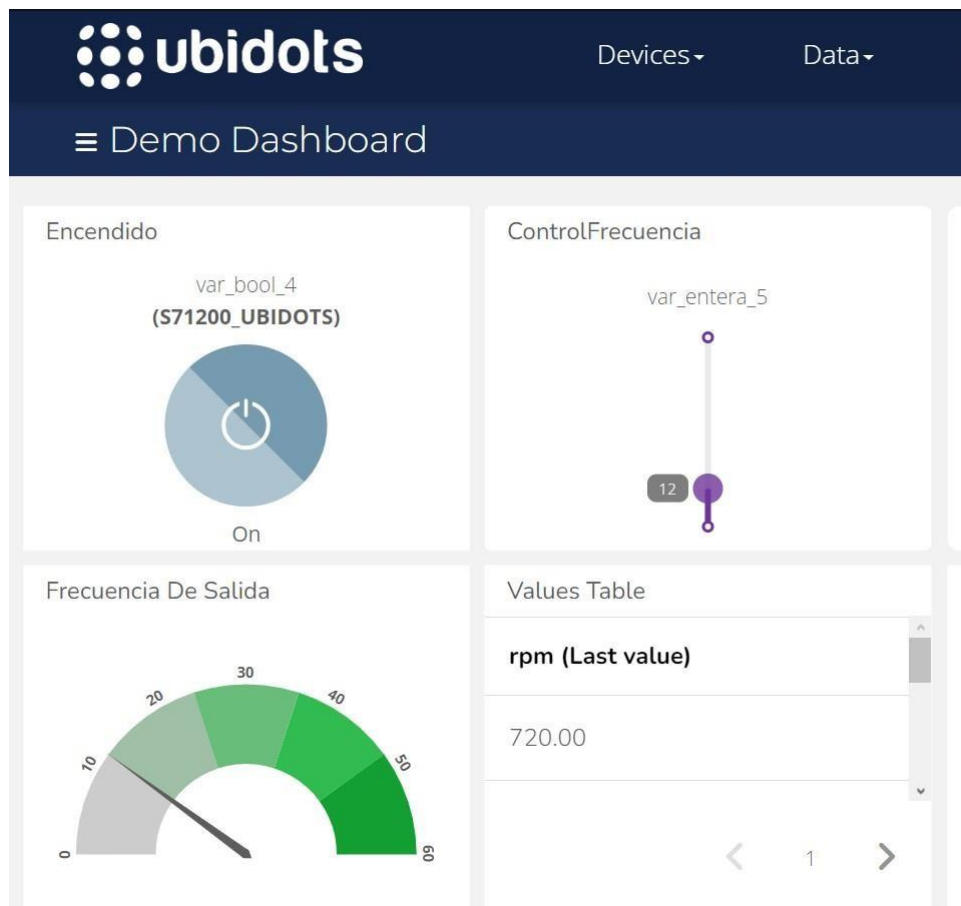


Figura 3.40: Frecuencia de salida

De igual manera, se llevó a cabo una verificación de las RPM a las que está girando el motor, y ese valor se mostrará en el indicador que se presenta en las Figuras 3.41, 3.42 y 3.43. En estas imágenes se constata cómo se está realizando la respectiva monitorización de la velocidad a la que está girando el motor.

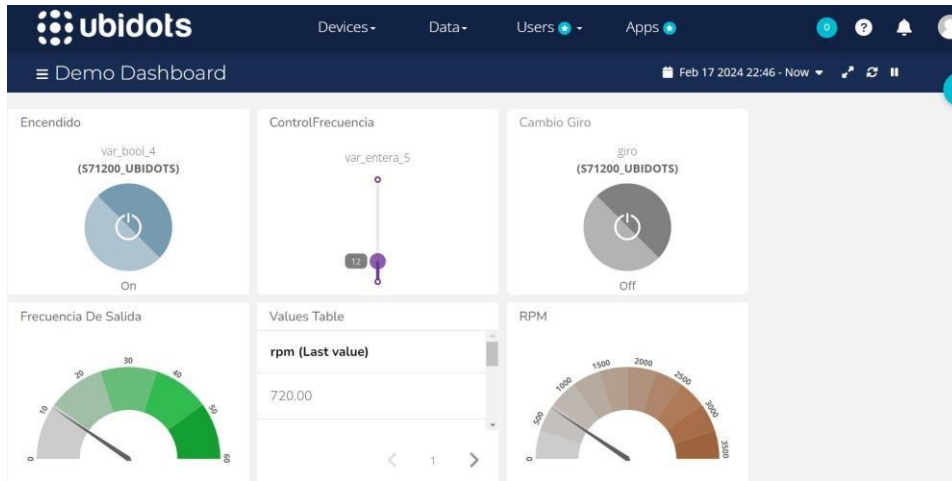


Figura 3.41: Lectura 720 RPM

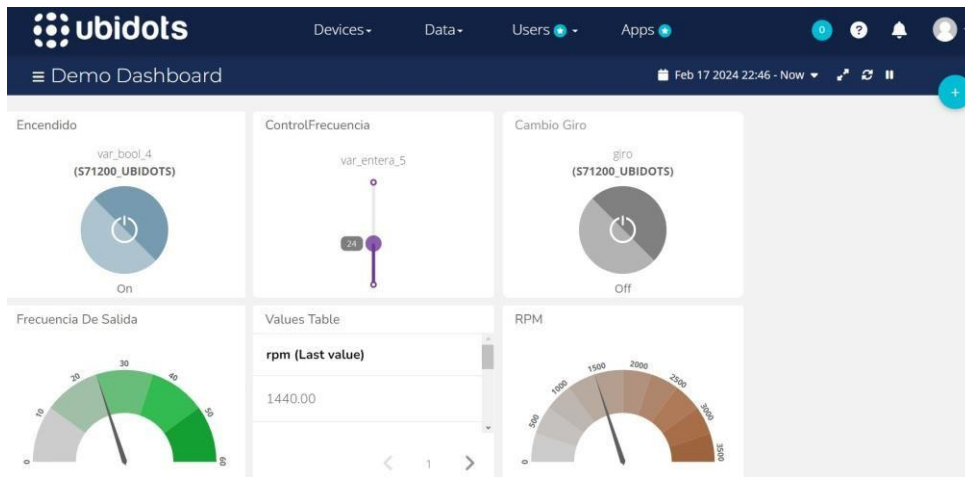


Figura 3.42: Lectura 1440 RPM

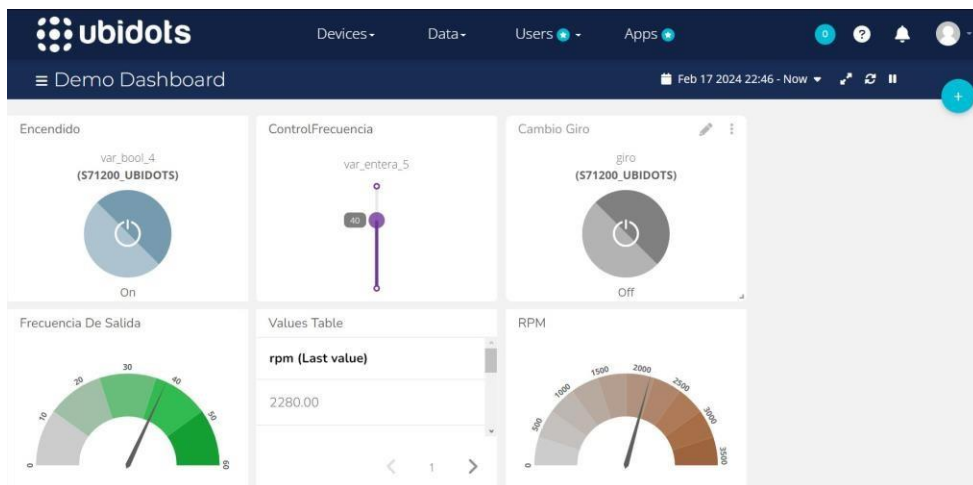


Figura 3.43: Lectura 2280 RPM

Capítulo 4

Conclusiones y Trabajos Futuros

En este trabajo, se ha explorado la implementación y el control efectivo de un variador de frecuencia Delta MS300 utilizando comunicación Modbus RS485, así como la integración de una pasarela inteligente basada en Raspberry Pi para ampliar las capacidades de control y monitoreo del sistema. A través de la plataforma en la nube de Ubidots, se ha demostrado la capacidad de controlar la frecuencia de operación del variador, el sentido de giro del motor, y analizar las RPM del mismo, lo que representa un avance significativo en el ámbito del control de motores eléctricos.

Uno de los aspectos más destacados de este trabajo es la versatilidad y la escalabilidad que proporciona la integración de la Raspberry Pi como pasarela inteligente. Esto permite una mayor flexibilidad en el control y monitoreo del variador de frecuencia, así como la posibilidad de implementar soluciones de control más avanzadas en el futuro. Además, la utilización de la plataforma en la nube de Ubidots ha facilitado el acceso remoto y la visualización de los datos del sistema, lo que resulta invaluable para el mantenimiento predictivo y la optimización del rendimiento del motor.

Si bien este estudio ha logrado importantes avances en el control y monitoreo de motores eléctricos a través de la combinación de tecnologías como Modbus RS485, Raspberry Pi y la nube de Ubidots, existen áreas para futuras investigaciones y mejoras. En este contexto se plantea la utilización de otras plataformas similares a Ubidots y el análisis de desempeño en cuanto a capacidades, tiempos de latencia, costos, etc. Por otra parte se plantea la necesidad de explorar el análisis y tratamiento

de los datos una vez se encuentran en la nube, por ejemplo utilizando técnicas estadísticas para imprimir históricos o tendencias y poder visualizar o predecir un mal funcionamiento de la máquina.

Glosario

CA Corriente Alterna – Alternating Current.

CC Corriente Continua – Direct Current.

CSI Inversor de Corriente – Current Source Inverter.

GPIO Entrada/Salida de Propósito General – General Purpose Input/Output.

GTO Tiristor Controlado por Gatera – Gate Turn-Off Thyristor.

IGBT Transistor Bipolar de Puerta Aislada – Insulated Gate Bipolar Transistor.

IoT Internet de las Cosas – Internet of Things.

IP Protocolo de Internet – Internet Protocol.

LCI Convertidor de Corriente Continua a Corriente Alterna – Load Commutated Inverter.

M2M Máquina a Máquina – Machine to Machine.

PAC Controlador Automático Programable – Programmable Automation Controller.

PLC Controlador Lógico Programable – Programmable Logic Controller.

RTU Unidad Terminal Remota – Remote Terminal Unit.

TIA Automatización Totalmente Integrada – Totally Integrated Automation.

VDF Variador de Frecuencia – Variable Frequency Drive.

VSI Inversor de Voltaje – Voltage Source Inverter.

Referencias

- [1] J. Sossenheimer, O. Vetter, T. Stahl, A. Weyand y M. Weigold, «Hybrid virtual metering points – a low-cost, near real-time energy and resource flow monitoring approach for production machines without PLC data connection,» *Procedia CIRP*, vol. 98, págs. 452-457, 2021, The 28th CIRP Conference on Life Cycle Engineering, March 10 – 12, 2021, Jaipur, India, ISSN: 2212-8271. DOI: <https://doi.org/10.1016/j.procir.2021.01.133>. dirección: <https://www.sciencedirect.com/science/article/pii/S2212827121001633>.
- [2] C. Sharma y N. K. Gondhi, «Pila de protocolos de comunicación para sistemas IoT restringidos,» en *2018 3ª Conferencia Internacional sobre Internet de las Cosas: Innovación y Usos Inteligentes (IoT-SIU)*, 2018, págs. 1-6. DOI: 10.1109/IoT-SIU.2018.8519904.
- [3] J. Sossenheimer, O. Vetter, T. Stahl, A. Weyand y M. Weigold, «Hybrid virtual metering points – a low-cost, near real-time energy and resource flow monitoring approach for production machines without PLC data connection,» *Procedia CIRP*, vol. 98, págs. 452-457, 2021, The 28th CIRP Conference on Life Cycle Engineering, March 10 – 12, 2021, Jaipur, India, ISSN: 2212-8271. DOI: <https://doi.org/10.1016/j.procir.2021.01.133>. dirección: <https://www.sciencedirect.com/science/article/pii/S2212827121001633>.
- [4] *The Modbus Organization*, <https://modbus.org/>, (accessed Feb. 12, 2024), 2020.
- [5] S. Tamboli, M. Rawale, R. Thoraiet y S. Agashe, «Implementación de comunicación Modbus RTU y Modbus TCP utilizando el PLC Siemens S7-1200 para el proceso por lotes,» en *Conferencia Internacional 2015 sobre Tecnologías Inteligentes y Gestión para la Computación, la Comunicación, los Controles, la Energía y los Materiales (ICSTM)*, 2015, págs. 258-263. DOI: 10.1109/ICSTM.2015.7225424.

- [6] Y. Zhu, F. Zhuo y L. Xiong, «Communication platform for energy management system in a master-slave control structure microgrid,» en *Proceedings of The 7th International Power Electronics and Motion Control Conference*, vol. 1, 2012, págs. 141-145. DOI: 10. 1109/ IP EMC. 2012. 6258875.
- [7] M. D. Hossain, H. Ochiai, T. Arisawa e Y. Kadobayashi, «Smart Meter Modbus RS-485 Spoofing Attack Detection by LSTM Deep Learning Approach,» en *2022 9th Swiss Conference on Data Science (SDS)*, 2022, págs. 47-52. DOI: 10. 1109/SDS54800. 2022. 00015.
- [8] M. Lekić y G. Gardašević, *Sistema Único de Autenticación – Universidad Politécnica Salesiana*, <http://bibliotecas.ups.edu.ec:2095/document/8345544/authors#authors>, Último acceso: 18 de febrero de 2024, 2024. dirección: <http://bibliotecas.ups.edu.ec:2095/document/8345544/authors#authors> (visitado 18-02-2024).
- [9] M. M. Swamy, J.-K. Kang y K. Shirabe, «Power Loss, System Efficiency, and Leakage Current Comparison Between Si IGBT VFD and SiC FET VFD With Various Filtering Options,» *IEEE Transactions on Industry Applications*, vol. 51, n.º 5, págs. 3858-3866, sep. de 2015, ISSN: 1939-9367. DOI: 10. 1109/TIA. 2015. 2420616.
- [10] Ultatek, *Cómo seleccionar un variador de frecuencia en función de la carga*, <https://ultatek.com/como-seleccionar-un-variador-de-frecuencia-basado-en-las-caracteristicas-de-carga/>, Accedido: 18 de febrero de 2024, 2022.
- [11] M. H. Khan, M. R. Uddin, Z. Tasneem y K. M. Salim, «Power Factor Analysis of an Energy Efficient Single Phase VFD for Industrial Sewing Machines,» en *2019 5th International Conference on Advances in Electrical Engineering (ICAEE)*, sep. de 2019, págs. 800-803. DOI: 10. 1109/ICAEE48663. 2019. 8975645.
- [12] V. Bolgova, A. Leonov y D. Charkov, «Influence of VFD parameters on voltage stresses in low voltage windings,» en *2016 57th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTU CON)*, 2016, págs. 1-5. DOI: 10 . 1109/RTU CON. 2016. 7763153.
- [13] M. M. Swamy, J.-K. Kang y K. Shirabe, «Power Loss, System Efficiency, and Leakage Current Comparison Between Si IGBT VFD and SiC FET VFD With Various Filtering Options,» *IEEE Transactions on Industry Applications*, vol. 51, n.º 5, págs. 3858-3866, 2015. DOI: 10. 1109/TIA. 2015. 2420616.

- [14] M. Lekić y G. Gardašević, «IoT sensor integration to Node-RED platform,» en *2018 17th International Symposium INFOTEH-JAHORINA (INFOTEH)*, 2018, págs. 1-5. DOI: 10.1109/INFOTEH.2018.8345544.
- [15] N. S. Ismail, N. A. Rashid, N. A. Zakaria, Z. I. Khan y A. Mahmud, «Low Cost Extended Wireless Network Using Raspberry Pi 3B+,» en *2020 IEEE Symposium on Industrial Electronics Applications (ISIEA)*, 2020, págs. 1-4. DOI: 10.1109/ISIEA49364.2020.9188215.
- [16] I. Gútiez. «SIMATIC IoT2040 - La Pasarela Inteligente de Siemens.» Último acceso: 18 de febrero de 2024. (oct. de 2018), dirección: <https://programacionsiemens.com/simatic-iot2040-la-pasarela-inteligente-de-siemens/> (visitado 18-02-2024).
- [17] E. Rodríguez, *Qué modelo de Raspberry Pi comprar: un repaso a las principales placas y los proyectos más habituales...* <https://www.xataka.com/seleccion/que-modelo-raspberry-pi-comprar-repaso-a-principales-placas-proyectos-habituales-para-dar-mejor>, Accedido: 18 de febrero de 2024, 2020.
- [18] E. R. de Luis. «Qué Modelo de Raspberry Pi Comprar: Un Repaso a Las Principales Placas Y Los Proyectos Más Habituales Para Dar Con La Mejor.» Último acceso: Fecha de Consulta Aquí. (2020), dirección: <https://www.xataka.com/seleccion/que-modelo-raspberry-pi-comprar-repaso-a-principales-placas-proyectos-habituales-para-dar-mejor> (visitado 02-08-2020).
- [19] A. Zare y M. T. Iqbal, «Low-Cost ESP32, Raspberry Pi, Node-Red, and MQTT Protocol Based SCADA System,» en *2020 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, 2020, págs. 1-5. DOI: 10.1109/IEMTRONICS51293.2020.9216412.
- [20] R. K. Kodali y A. Anjum, «IoT Based HOME AUTOMATION Using Node-RED,» en *2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT)*, 2018, págs. 386-390. DOI: 10.1109/ICGCIoT.2018.8753085.
- [21] A. A. Ismail, H. S. Hamza y A. M. Kotb, «Performance evaluation of open source IoT platforms,» en *2018 IEEE global conference on internet of things (GCIoT)*, IEEE, 2018, págs. 1-5.
- [22] Barbara, *Plataformas IoT: Diferentes tipos de plataformas y cómo elegir la que necesitas*, <https://www.barbara.tech/es/blog/tipos-de-plataforma-iot>, Accedido: 18 de febrero de 2024, 2021.

- [23] alfaiot-webmaster, *¿Qué es una plataforma IOT?* <https://alfaiot.com/iot/que-es-una-plataforma-iot/>, Accedido: 18 de febrero de 2024, 2023.