Recibido: 3/12/2023
Aceptado: 11/12/2023

# Conexión Arduino Modbus

## Arduino Modbus Connection

Christian Eduardo Hincapie Blandón[1]
Angelo Morales Castaño[2]
José David López Alzate[3]
Luis Edier Gañan Gañan[4]

Cómo citar:
Hincapie, C., Morales, A., Lopez , J., y Gañan, L. (2023). Arduino Modbus Connection: A Case Study. Revista Sennova: Revista Del Sistema De Ciencia, Tecnología E Innovación, 7(1), 87-107. https://doi.org/10.23850/23899573.6062.

1 Tecnólogo en automatización de sistemas mecatrónicos, centro de automatización industrial, thepalmdeira@gmail.com, Manizales, Caldas.
2 Tecnólogo en automatización de sistemas mecatrónicos, centro de automatización industrial, anghelotiy@gmail.com, Manizales, Caldas.
3 Instructor automatización, Centro de automatización industrial, jdlopez@sena.edu.co, Manizales, Caldas.
4 Instructor automatización, centro de automatización industrial, lganan@sena.edu.co, Manizales, Caldas.

## RESUMEN

Este artículo proporciona una visión general para la configuración y operación básicas de la comunicación Modbus serial entre un Arduino Uno, el cual actúa como un dispositivo esclavo que recibe y ejecuta comandos, y CODESYS 3.5, cuya función es de un dispositivo maestro queinicia y controla la comunicación. Esta configuración requiere ciertos parámetros que se deben tener en cuenta para establecer una conexión óptima desde la parte física con la asignación del puerto serial para el Arduino, así como la sincronización en la velocidad de transmisión, bits de datos, paridad y otros parámetros que coincidan con lo especificado en el tipo de interfaz RS-232, RS-485 del CODESYS 3.5. Lo anterior con el propósito de leer o escribir datos en el Arduino Uno mediante el uso de funciones como “Read Holding Registers” para obtener información desde el dispositivo esclavo, y “Write Single Register” para enviar la información, pruebas que se hacen de forma preliminar con el uso del software Modbus Poll para garantizar latransmisión.

Esta actividad permitió alcanzar uno de los objetivos claves de la temática de redes decomunicación industrial del programa Tecnólogo en Automatización de Sistemas Mecatrónicos,como es comprender la estructura de un sistema de comunicación Modbus Serial a partir deelementos simples como Arduino Uno y el software CODESYS 3.5, ambos sistemas de código abierto, mediante un formación teórico-práctica que dio como resultado un aprendizaje basado en el proyecto descrito en este artículo.


**Palabras clave:** *Comunicación, Redes, RS232, RS485, Automatización, Serial, Interfaz, Datos.*

## Abstract

This article provides an overview for the basic configuration and operation of the Modbus serial communication between an Arduino Uno, which acts as a slave device that receives and executes commands, and CODESYS 3.5 whose function is that of a master device that initiates and controlsthe communication. This configuration requires certain parameters that must be taken into accountto establish an optimal connection from the physical part with the assignment of the serial port for the Arduino, as well as the synchronization in the transmission speed, data bits, parity and other parameters that match what is specified in the type of RS-232, RS-485 interface of CODESYS 3.5. The above with the purpose of reading or writing data in the Arduino Uno by using functions such as “Read Holding Registers” to obtain information from the slave device, and “Write Single Register” to send the information, tests which are done preliminarily with the use of the “Modbus Poll” software to ensure the transmission.

This activity allowed to achieve one of the key objectives of the thematic of industrial communication networks of the Mechatronic Systems Automation Technologist program, as is tounderstand the structure of a Modbus Serial communication system from simple elements such asan Arduino one and CODESYS 3.5 software, both open source systems, this was possible througha theoretical and practical training that resulted in a project-based learning which is described in this article.


**keywords:** *Communication, Networking, RS232, RS485, Automation, Serial, Interface, Data.*

## 1. Introducción

De acuerdo con los constantes avances tecnológicos y la necesidad del sector productivo de optimizar sus procesos en tiempos y calidad del producto final, aparecen los sistemas automatizados teniendo como base una serie de dispositivos que controlan y monitorean los diversos sistemas compuestos en su mayoría de veces por ordenadores, autómatas programables, robots y tecnologías de la información para manejar diferentes procesos productivos y maquinarias en la industria, teniendo un impacto positivo al reemplazar operaciones mecánicas de ensamblaje que son peligrosas, por operaciones automatizadas caracterizadas por conceptos deindustria 4.0.

Dentro de la evolución continua de estos sistemas automatizados surge la necesidad de intercambiar información entre los dispositivos de campo como los autómatas programables, siendo base de una estructura piramidal de las comunicaciones industriales, eficiente y confiable en entornos automatizados. En este contexto, el protocolo Modbus serial emerge como un estándar ampliamente utilizado para facilitar la comunicación entre dispositivos, permitiendo la supervisión y control efectivos en sistemas de automatización.

Por lo anterior, este artículo se centra en la implementación de la comunicación Modbus serial entre un microcontrolador Arduino Uno conocido por su versatilidad y facilidad de programación y que actúa como el nodo esclavo para ejecutar tareas específicas dentro del sistema, y el entorno de desarrollo CODESYS 3.5 que es empleado en la programación de equipos para la automatización industrial, en este caso desempeña el papel de nodo maestro, dirigiendo la comunicación y coordinando las operaciones del sistema. Combinando estos dos sistemas se logra el desarrollo de aplicaciones claves en los procesos de automatización industrial.

## 2. Materiales y métodos

Este trabajo se desarrolló como parte del resultado de aprendizaje: integrar redes de comunicación industrial en sistemas de manufactura de acuerdo a procedimientos técnicos, de la temática Redes de comunicaciones y SCADA, correspondiente al VI trimestre del programa de formación Tecnólogo en Automatización de Sistemas Mecatrónicos, donde a través de un aprendizaje basado en proyectos se planteó la actividad “Conexión Arduino Modbus” para comprender de una mejor manera cómo funciona el intercambio de información

entre dispositivos mediante elprotocolo Modbus serial. Los materiales utilizados para el desarrollo de dicha actividad deaprendizaje son:

1. **Arduino Uno**: tarjeta de desarrollo de bajo costo**,** fácil de conseguir, fácil de programar y de código abierto, la cual actuará como el dispositivo esclavo en la comunicación serial. Esta tarjeta se programa bajo un entorno de desarrollo integrado (IDE) (Figura 1), usandoel lenguaje de programación C++.


**Figura 1. IDE para programación de la placa ArduinoFuente: https://arduino.cl/programacion/**

2. **Computador con Software CODESYS 3.5: (**Figura 2) Es un entorno de desarrollo libre utilizado como herramienta de ingeniería de proyectos para aplicaciones de automatización que está bajo la norma de los lenguajes de programación *IEC 61131-3*, fácil de instalar y de usar. Se ejecutará en un computador portátil o de escritorio que actuará como el dispositivo maestro en la comunicación.


**Figura 2. Entorno de programación CODESYS 3.5 Fuente: https://www.codesys.com/**

3. **Software Modbus Poll: (**Figura 3) Este software se emplea para validar la programación Modbus descargada en el Arduino, así como el direccionamiento de los registros empleados en este dispositivo. Con este programa, puede monitorear múltiples esclavos Modbus o áreas de datos al mismo tiempo.


**Figura 3.** Software de prueba comunicación Modbus Fuente: https://www.modbustools.com/modbus_poll.html

4. **Cable de Conexión Serial:** (Figura 4) Se utiliza un cable serial USB tipo B para establecer la conexión física entre el Arduino Uno y el computador donde se instaló el software CODESYS 3.5.

**Figura 4.** Cable de interfaz PC placa Arduino Fuente: https://www.sigmaelectronica.net/producto/

5. **Tarjeta Protoboard: (**Figura 5) Elemento empleado para la conexión rápida de elementos electrónicos e interfaz de prueba, se utiliza para validar en funcionamiento delcircuito, en este caso, los elementos del proyecto como LED, pulsadores, resistencias e interfaz de entrada y salida del Arduino Uno.

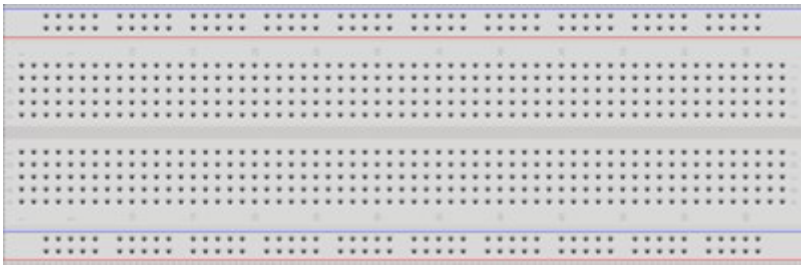


**Figura 5. Placa de conexión de elementos electrónicos ProtoboardFuente: https://www.sigmaelectronica.net/producto/**

Los métodos empleados para validar y probar el estado del proyecto fueron los siguientes:

1. **Configuración del Puerto Serial en Arduino Uno:** después de abrir el entorno dedesarrollo Arduino IDE, se configura el puerto serial, el cual incluye la velocidad de transmisión baud rate de **9600**, bits de datos **8**, paridad **None** y bits de parada **1**, estaconfiguración debe coincidir con la configuración que se utilizará en CODESYS 3.5.
2. **Configuración de CODESYS 3.5:** se inicia por crea un nuevo proyecto y configura la comunicación Modbus serial, es decir, se definen los parámetros del puerto serial como lavelocidad de transmisión en **9600 bps**, bits de

datos en **8**, paridad **None** y bits de parada **1**, asegurándose de que coincidan con la configuración del Arduino Uno.

3. **Programación en Arduino Uno:** se desarrolla el programa para el Arduino Uno para el control de tres LED, lectura de la señal digital de dos pulsadores normalmente abiertos, y la lectura de una señal analógica de un potenciómetro. Antes de proceder a transferir el programa al Arduino Uno, se configuran las direcciones de los registros Modbus que recibirán y transmitirán la información al CODESYS 3.5.
4. **Programación en CODESYS 3.5:** se desarrolla una lógica de control simple para interactuar con el Arduino Uno a través de las funciones Modbus. Se asignan las direcciones Modbus a las variables que se utilizarán para la lectura y escritura de datos, de igual manera se diseña un interfaz de visualización para garantizar el envío yla recepción de los datos, además de un diseño amigable con el usuario.
5. **Pruebas y Depuración:** una vez descargado el programa en ambos dispositivos, se realizan pruebas para verificar la comunicación entre el Arduino Uno y CODESYS 3.5, seemplean herramientas de monitoreo Modbus para verificar el intercambio de datos y asegurar una comunicación estable y precisa al direccionarse a los registros correctos.

Para desarrollar una comunicación entre los dos sistemas Arduino-CODESYS 3.5 a través del protocolo Modbus, se realizó un montaje simple para validar la comunicación y comprender las características, parámetros, configuración, y el intercambio de datos entre el sistema maestro CODESYS 3.5 y el dispositivo esclavo Arduino Uno. Figura 6.

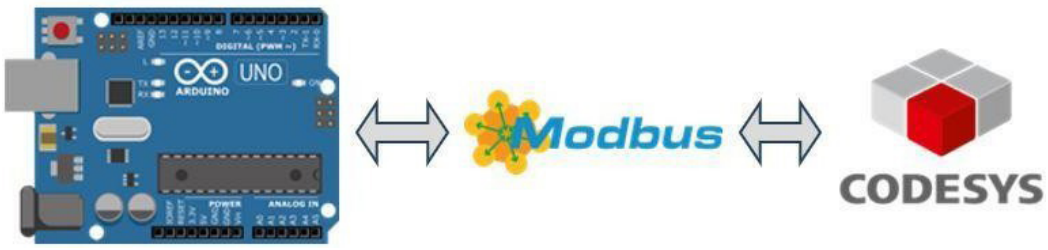



**Figura 6. Dispositivo físico y software para comunicaciónFuente: Recurso de Internet.**

La programación de la tarjeta Arduino Uno inició con la descarga de la librería *"Modbus-ArduinoExample - Lamp (Modbus Serial) Copyright by André Sarmento*

*Barbosa*", esta librería permite configurar las entradas y salidas tanto digitales como analógicas de la placa Arduino permitiendoque esta tarjeta pueda actuar como si fuera un Controlador Lógico Programable (PLC).

A continuación, se describen las líneas de código empleadas en la programación de latarjeta Arduino Uno.

- Se incluye la librería de Modbus.h y ModbusSerial.h como requisito para la comunicaciónModbus:

```
#include “Modbus.h”
#include “ModbusSerial.h”
```

- Se procede a crear y asociar variables a los registros Modbus, estableciendo una estructura de datos clara. Este paso fue esencial para la comunicación efectiva entre el Arduino y otros dispositivos. Aquí están las variables y sus respectivos registros:

```
const int LAMP1_COIL = 100;
const int LAMP2_COIL = 101;
const int LAMP3_COIL = 102;
const int INP1_REG =200;
const int INP2_REG =201;
const int SENSOR_IREG = 300;
const int SALI_HREG = 400;
```

- Se definen las variables para los pines de entrada/salida del Arduino Uno, lo cual fue necesario para interactuar con el entorno físico. Estas variables se asignaron a los pines correspondientes de la placa:

```
const int VERDE = 5;
const int ROJO = 4;
const int AMARILLO = 3;
const int START = 8;
const int STOP = 9;
const int AnalogPin = A0;
const int SALIDA = 11; //PWM
```

- Se declara el objeto Modbus Serial, en este caso se nombró ‘mb’. Este objeto es crucial para facilitar la comunicación Modbus a lo largo del programa.

```
ModbusSerial mb;
```

- Se declara una variable numérica llamada ‘ts’, la cual tiene una longitud de 32 bits sin parte decimal. Esta variable se utilizaría para almacenar el valor de la función ‘millis()’ y así medir el tiempo transcurrido en milisegundos desde el inicio del programa:

```
long ts;
```

- En el bloque ‘void setup()’ se realizan las configuraciones iniciales. Se configura el Modbus serial especificando el puerto, la velocidad de transmisión y el formato del byte:

```
void setup() {
mb.config(&Serial, 9600, SERIAL_8N1);
```

- Se establece una identificación para el dispositivo esclavo, eligiendo el valor 10 para este ejemplo:

```
mb.setSlaveId(10);
```

- Se definen los pines digitales de entrada y salida de la tarjeta Arduino, esenciales para la interacción física con el entorno:

```
pinMode(VERDE, OUTPUT);
pinMode(ROJO, OUTPUT);
pinMode(AMARILLO, OUTPUT);
pinMode(START, INPUT);
pinMode(STOP, INPUT);
```

- Se adicionan los registros Modbus, claves en la transferencia de información para las variables digitales como salidas, entradas y variables analógicas:

```
mb.addCoil(LAMP1_COIL);
mb.addCoil(LAMP2_COIL);
mb.addCoil(LAMP3_COIL);
```

```
mb.addIsts(INP1_REG);
mb.addIsts(INP2_REG);
mb.addIreg(SENSOR_IREG);
mb.addHreg(SALI_HREG);
```

- Se asigna el valor actual de 'millis()' en la variable 'ts', que serviría para medir el tiempo transcurrido:

```
ts = millis();
```

- En el bloque 'void loop()', el cual representa el bucle principal del programa, se adiciona el objeto Modbus para ejecutar las tareas fundamentales del intercambio de datos, siendo una parte esencial del programa:

```
void loop() {
     mb.task();
```

- En este punto se asocian las variables del programa con los registros Modbus correspondientes solo para las entradas y salidas digitales:

```
mb.Ists(INP1_REG, digitalRead(START));
mb.Ists(INP2_REG, digitalRead(STOP));
digitalWrite(VERDE, mb.Coil(LAMP1_COIL));
digitalWrite(ROJO, mb.Coil(LAMP2_COIL));
digitalWrite(AMARILLO, mb.Coil(LAMP3_COIL));
```

- De igual manera se asocia una variable analógica de salida con el registro Modbus correspondiente, con un retardo de 15 ms:

```
analogWrite(SALIDA, mb.Hreg(SALI_HREG));
    delay(15);
```

- Se realiza la configuración para la lectura del valor analógico de entrada y se carga ese valor de lectura en el registro Modbus 'SENSOR_IREG', con un retraso en la lectura de 2000 ms:

```
if (millis() > ts + 2000) {
   ts = millis();
   mb.Ireg(SENSOR_IREG, analogRead(AnalogPin));
```

El montaje físico de los elementos se realizó sobre una Protoboard. Se incorporaron tres indicadores luminosos LED, tres resistencias 220Ω, un potenciómetro, dos pulsadores NA y unvoltímetro para la lectura de la señal Arduino Uno. El objetivo principal se centra en un intercambio bidireccional de datos digitales y analógicos desde CODESYS 3.5 y la placa Arduino por medio de Modbus serial.

El diagrama del montaje de los elementos en la placa se puede observar en la Figura 7, para lo cualse empleó una página en línea de libre acceso para diseño de circuitos electrónicos llamado TinkerCAD.

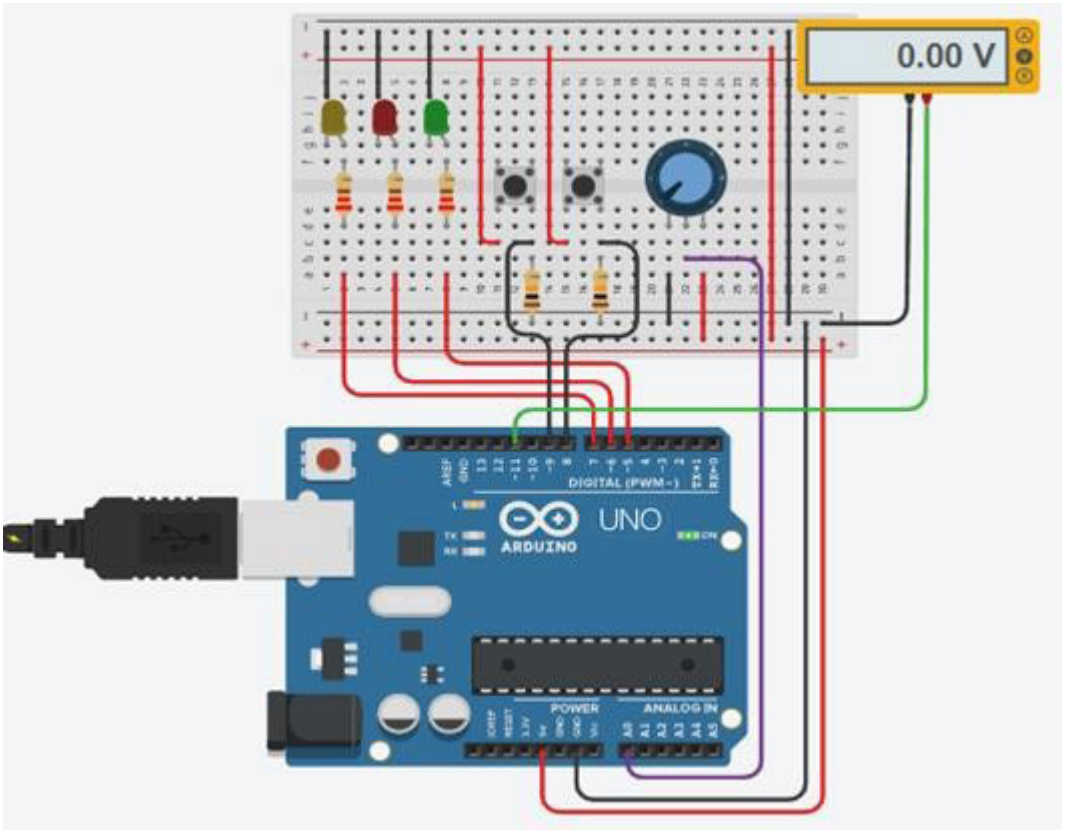



**Figura 7. Diseño o diagrama de conexión de los elementos.Fuente: https://www.tinkercad.com/**

Es necesario realizar una prueba de comunicación antes del enlace final con CODESYS 3.5, paraesta prueba se utiliza el Software Libre Modbus Poll donde se configuran los mismos parámetros que se configuraron para la placa Arduino Uno como se observa en la Figura 8.

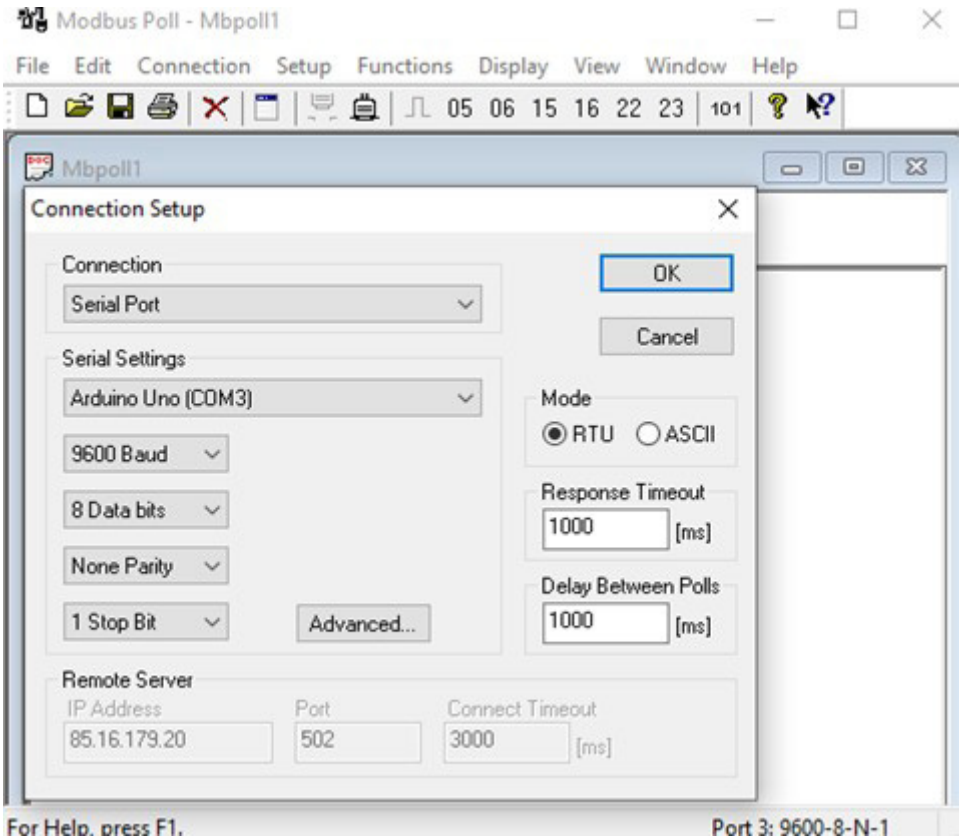



**Figura 8. Diseño o diagrama de conexión de los elementosFuente: Los Autores.**

Se realiza la prueba para la activación de las tres salidas digitales, enlazadas con los registros100, 101, 102, como se observa en la Figura 9.

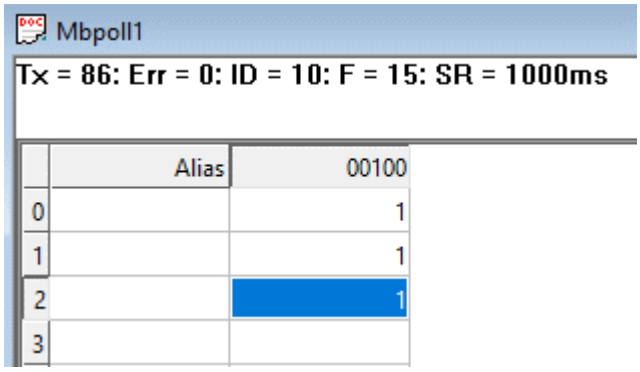



**Figura 9. Envío de datos para activación de salidas en el Arduino.Fuente: Los Autores.**

Se realiza la misma prueba para la lectura de las dos entradas digitales, enlazadas con losregistros 200, 201, como se observa en la Figura 10.

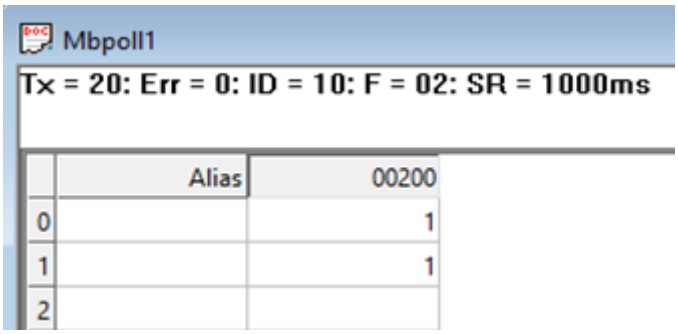



**Figura 10. Lectura de los pulsadores conectados al Arduino.Fuente: Los Autores.**

Es momento para probar la entrada analógica correspondiente al registro 300 de la lectura del potenciómetro. Entrega un valor desde 0 a 1023, como se observa en la Figura 11.

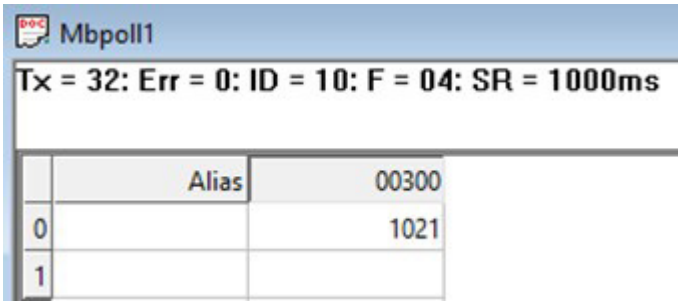



**Figura 11. Lectura de señal analógica desde el Arduino.Fuente: Los Autores.**

La última prueba desde el Modbus Poll corresponde a la escritura de una salida del Arduino por PWM a través del registro 400, cuyo valor de escritura va desde 0 para 0 Voltios, hasta 255 correspondiente a un valor de 5 Voltios.

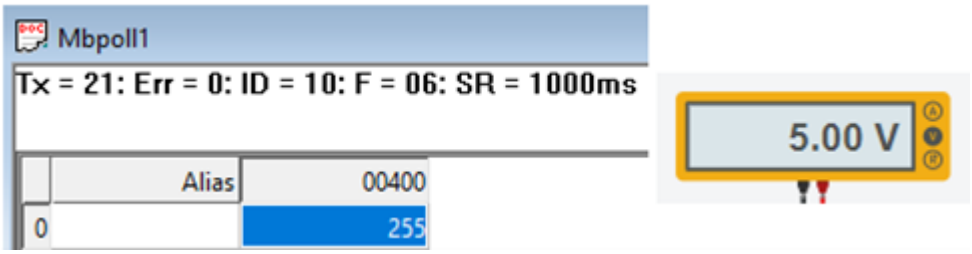
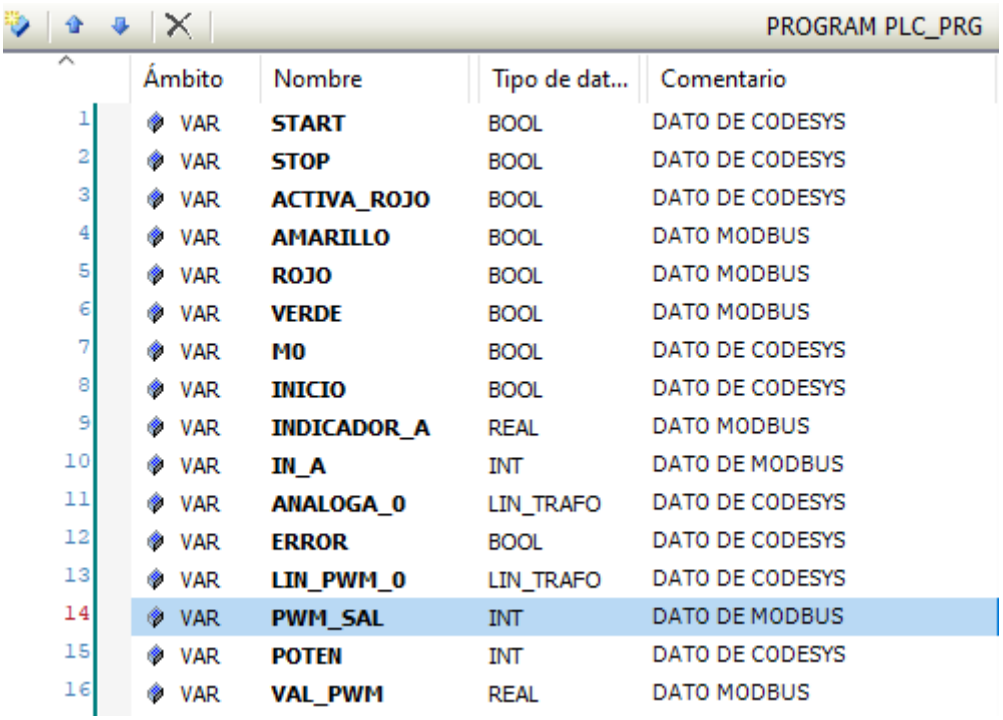



**Figura 12. Escritura de un valor PWM Arduino.Fuente: Los Autores.**

Después de realizar las pruebas de comunicación Modbus y garantizar una ejecución correcta enel Arduino Uno, se procede a abrir el Software CODESYS 3.5 y crear un nuevo proyecto. Unavez creado el proyecto se agregan las variables y se clasifican de acuerdo a si son datos internos propios de CODESYS para la lógica de programación o datos Modbus que se intercambiarán con el Arduino, como se observa en la Figura 13.




| | Ámbito | Nombre | Tipo de dat... | Comentario |
|---|---|---|---|---|
| 1 | VAR | START | BOOL | DATO DE CODESYS |
| 2 | VAR | STOP | BOOL | DATO DE CODESYS |
| 3 | VAR | ACTIVA_ROJO | BOOL | DATO DE CODESYS |
| 4 | VAR | AMARILLO | BOOL | DATO MODBUS |
| 5 | VAR | ROJO | BOOL | DATO MODBUS |
| 6 | VAR | VERDE | BOOL | DATO MODBUS |
| 7 | VAR | M0 | BOOL | DATO DE CODESYS |
| 8 | VAR | INICIO | BOOL | DATO DE CODESYS |
| 9 | VAR | INDICADOR_A | REAL | DATO MODBUS |
| 10 | VAR | IN_A | INT | DATO DE MODBUS |
| 11 | VAR | ANALOGA_0 | LIN_TRAFO | DATO DE CODESYS |
| 12 | VAR | ERROR | BOOL | DATO DE CODESYS |
| 13 | VAR | LIN_PWM_0 | LIN_TRAFO | DATO DE CODESYS |
| 14 | VAR | PWM_SAL | INT | DATO DE MODBUS |
| 15 | VAR | POTEN | INT | DATO DE CODESYS |
| 16 | VAR | VAL_PWM | REAL | DATO MODBUS |

**Figura 13. Creación de variables en Codesys 3.5.Fuente: Los Autores.**

En el siguiente paso, se agregan en el árbol de proyecto de CODESYS los dispositivos Modbus como se observa en la Figura 14, los cuales permiten configurar los parámetros deacuerdo a los de la tarjeta Arduino Uno.


Figura 14. Adición de los dispositivos Modbus.Fuente: Los Autores.

Después se deben crear los canales que permitirán el intercambio de datos al asignar lasdirecciones Modbus que se configuraron previamente en el Arduino Uno. Figura 15.

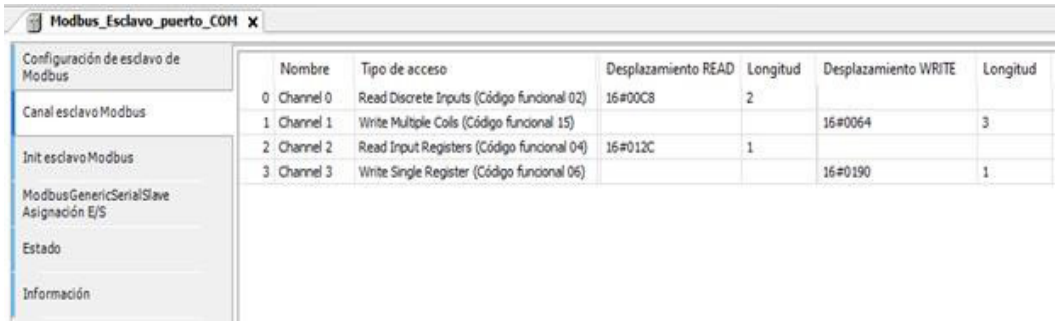



**Figura 15. Adición de los canales de acuerdo con los registros Modbus.Fuente: Los Autores.**

Una vez creados los canales de comunicación, se procede a asignar las direcciones de entradas y salidas, tanto digitales como analógicas, que intervienen en el intercambio de datos entre ambos dispositivos. Figura 16.

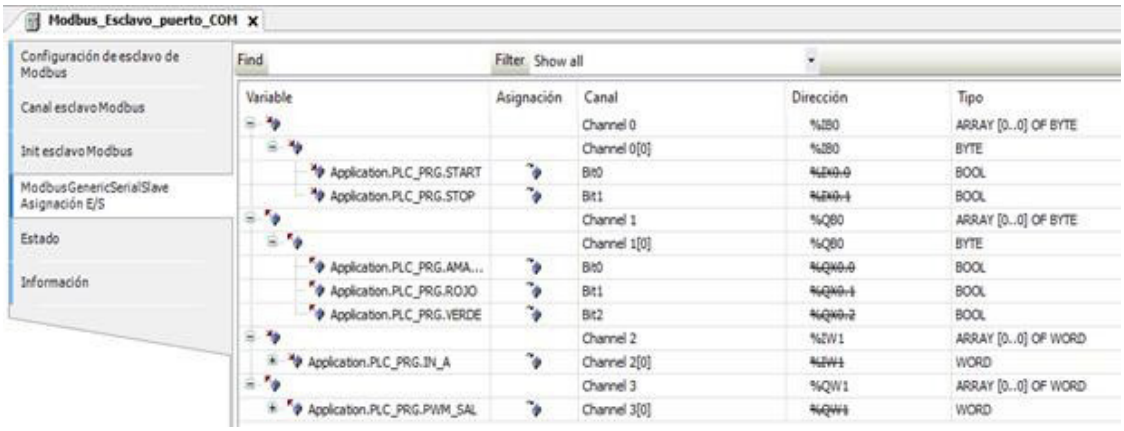



**Figura 16. Asignación de direcciones Modbus.Fuente: Los Autores.**

Para realizar la programación en lenguaje de contactos Ladder, se inicia por un programa sencillo,donde al activar el pulsador de START en la protoboard se enciende el LED amarillo, quedando energizado mediante una marca interna denominada M0. Al presionar el pulsador de STOP, la marca se desactiva y el LED amarillo se apaga; se utiliza un contacto abierto de una marca interna llamada ACTIVA_ROJO para encender el LED rojo en la protoboard, de igual

manera se crea una marca INICIO para activar el LED verde directamente. Esta parte del programa corresponde a la Figura 17.

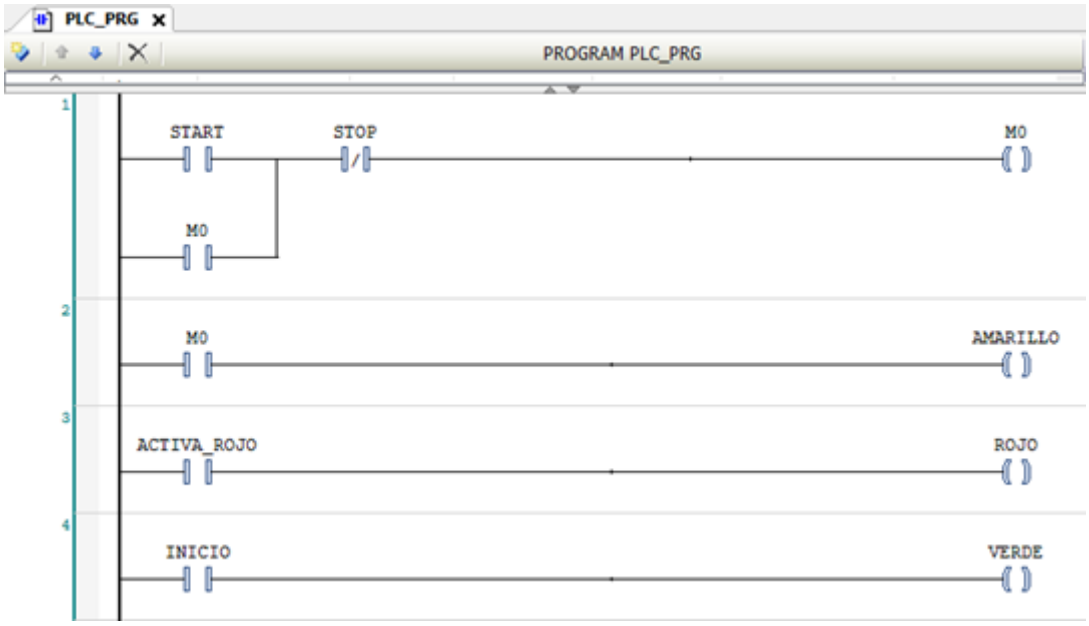



**Figura 17. Programa en Ladder para entradas y salidas digitales.Fuente: Los Autores.**

Sigue el programa para la entrada y salida analógica. Como lo muestra la Figura 18, ambas señalesse escalan de 0 a 100 para ser representadas gráficamente en una interfaz de visualización.

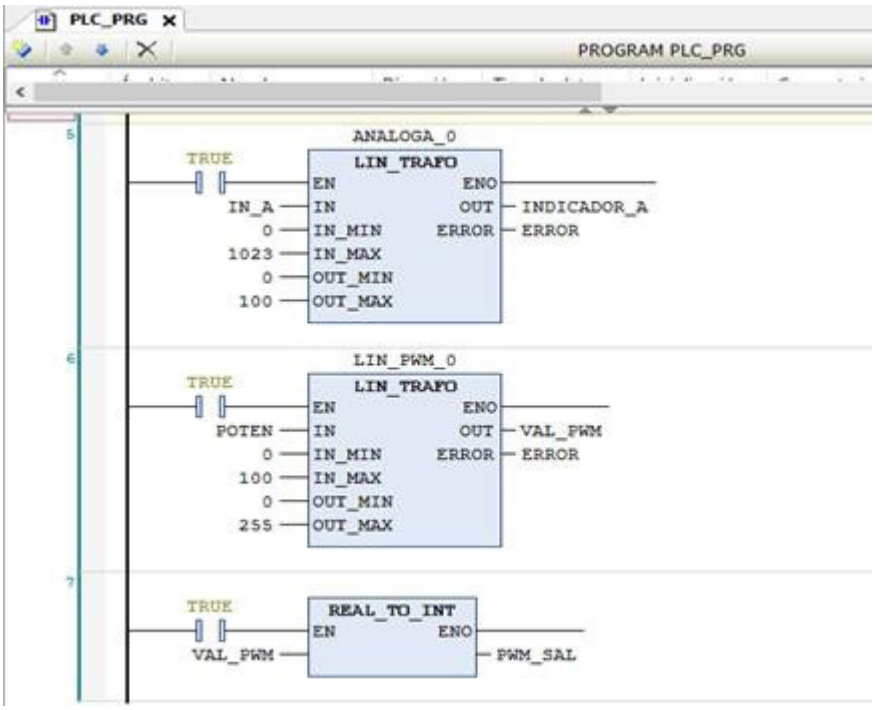



**Figura 18. Programa en Ladder para entradas y salidas analógicas.Fuente: Los Autores.**

## 3. Resultados y discusión

Como resultado de esta actividad de formación, se realizó una interfaz de visualización que permite recibir y enviar los datos de una forma amigable y fácil de entender para el usuario final.Figura 19.

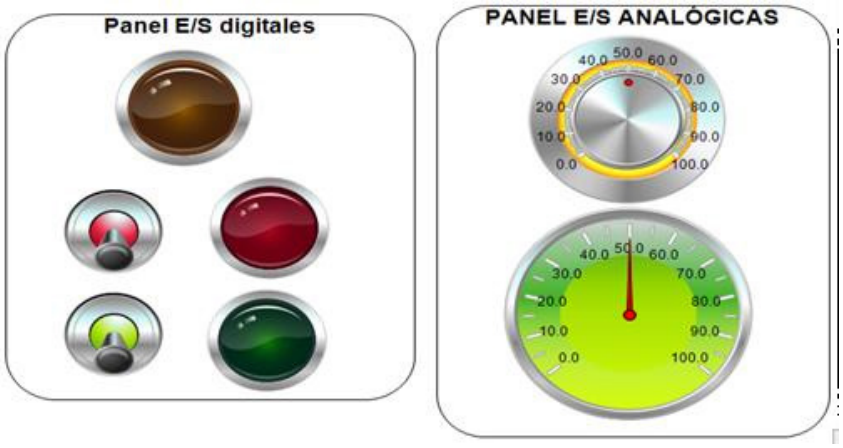



**Figura 19. Interfaz gráfica de visualización en CODESYS.Fuente: Los Autores.**

Otro resultado importante es el montaje de prueba físico de los elementos, como lo muestra laFigura 20, y permitió validar el intercambio de información, así como la velocidad de transmisión, parametrización y estabilidad de las señales.

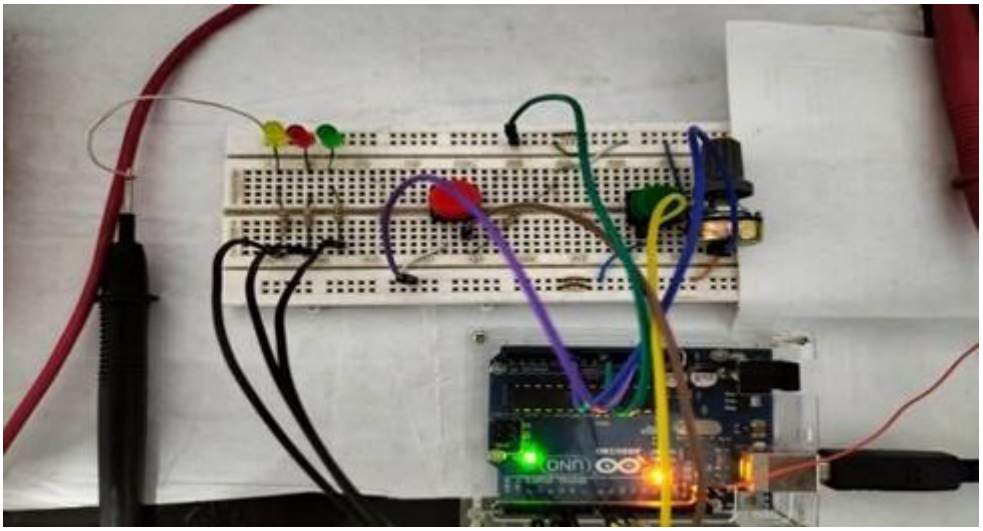


**Figura 20. Interfaz gráfica de visualización en CODESYS.Fuente: Los Autores.**

En el inicio de la programación y para poder entender la estructura de un protocolo decomunicación industrial, se hace necesario realizar actividades prácticas de este tipo, pues permite entender la lógica del programa mediante un entorno fácil de comprender. Figura 21.

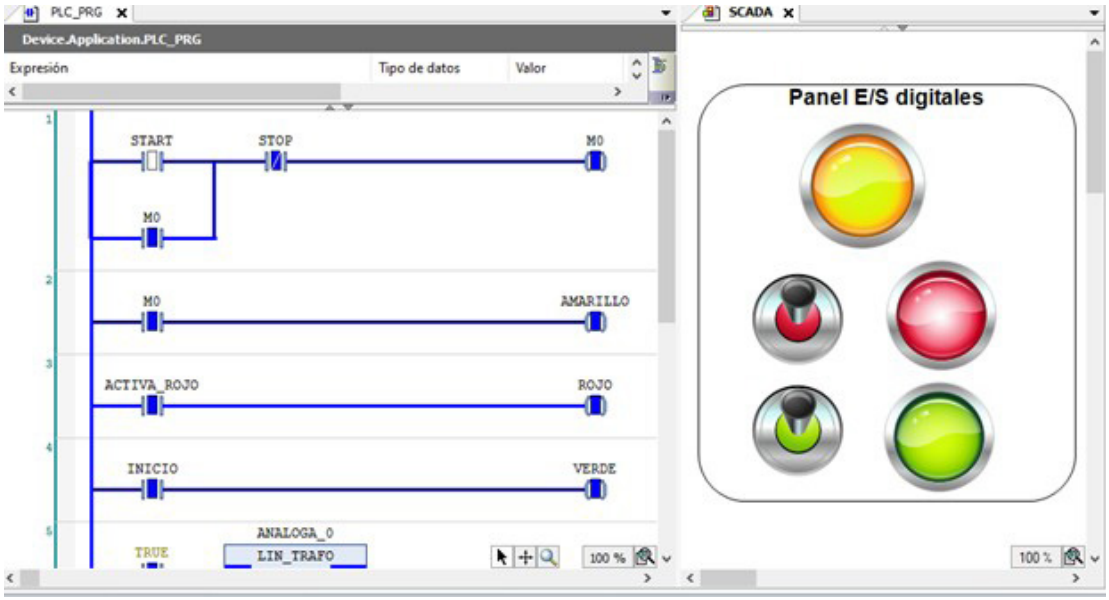



**Figura 21.** Monitoreo en línea de la programación en CODESYS.Fuente: Los Autores.

Para este caso, validar el programa realizado como lo muestra la Figura 22, garantiza el resultado óptimo de la comunicación Modbus, además de abrir un horizonte de oportunidades y de mejoras a diversos procesos encontrados a nivel industrial.

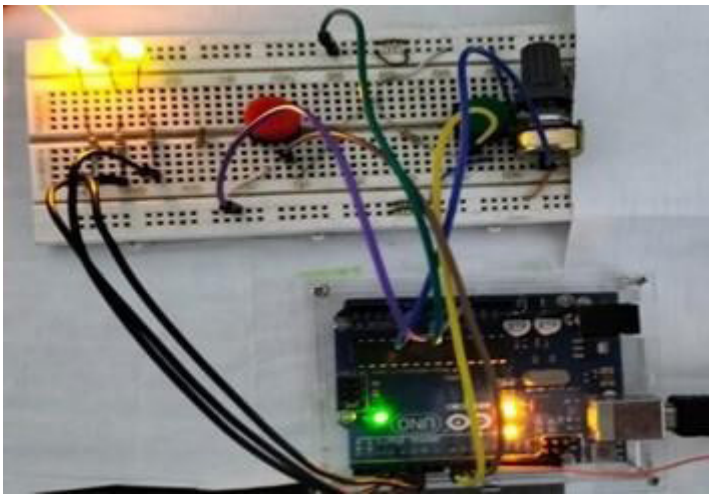


**Figura 22. Prueba de la comunicación Modbus.Fuente: Los Autores.**

El resultado más importante es comprender que no hay límites para adquirir conocimientos, puesdesde el dispositivo más sencillo hasta el más complejo al momento de trabajar con estos si se cuenta buenos conceptos a partir de una buena fundamentación teórico-práctica, permitirá que losegresados del Servicio nacional de aprendizaje SENA se destaquen en cualquier ámbito laboral.

## 4. Conclusiones

El protocolo Modbus demuestra la viabilidad de integrar dos tecnologías muy usadas en la automatización industrial. Esto abre oportunidades para aplicaciones más avanzadas ensistemas de control complejos.

La implementación de esta comunicación se llevó a cabo de manera relativamente sencilla,aunque con algunos retrasos, lo que sugiere que con conocimientos básicos en automatización se puede llevar a cabo un proyecto de interconexión con estas dosplataformas que aparentemente son sustancialmente diferentes.

El control de algún tipo de proceso en tiempo real se hace evidente en este tipo de proyecto donde a través elementos de entradas y salidas tanto digitales como analógicas, se puede notar la eficiencia de un sistema de acuerdo a sus tiempos de respuesta, pues es una exigencia en el sector industrial donde la respuesta debe ser inmediata ya que es esencial para el rendimiento en un sistema.

Aunque se logró una comunicación exitosa, se identificaron desafíos como la seguridad de la red y la optimización del rendimiento de acuerdo con exigencias de la industria 4.0 que deben abordarse en futuras investigaciones.

Con aplicaciones de software libre se pueden hacer proyectos interesantes que busquen dar soluciones concretas a pequeñas empresas que requieran comunicaciones industrialessimples y económicas en sus procesos.

## 5. Referencias

*Arduino Modbus. (s/f).*. (18 de 09 de 2023). Obtenido de Arduino.Cc:https://www.arduino.cc/reference/en/libraries/arduinomodbus/

*Campus Tecnológico. (s/f)*. (18 de 09 de 2023). Obtenido de Campus Tecnológico:https://campustecnologicovirtual.es/

*CODESYS Modbus TCP/RTU. (s/f).* (18 de 09 de 2023). Obtenido de Codesys.com: https://www.codesys.com/products/codesys-fieldbus/modbus-tcpr-tu.html

Fuller, J. (14 de 04 de 2023). *Modbus - Arduino tutorial.* Obtenido de Circuits DIY: https://www.circuits-diy.com/modbus-arduino-tutorial

Garcia, D. (. (18 de 09 de 2023). *Guía teórico práctica de Modbus en Arduino*. Obtenido de Infoplc.net: https://www.infoplc.net/documentacion/7-comunicaciones- industriales/2700-modbus-arduino

Lara, E. (21 de 11 de 2016). *PROTOCOLO MODBUS: COMUNICACIÓN ARDUINO vs ANDROID.* . Obtenido de Blog EscolaEspai.: https://www.espai.es/blog/2016/11/protocolo-modbus/

Satoshi. (21 de 11 de 2020). *De Arduino a Codesys con Modbus*. Obtenido de Opiron: https://www.opiron.com/de-arduino-a-codesys-con-modbus/