

PRÁCTICA

Tema: Comunicación de una red Modbus RTU

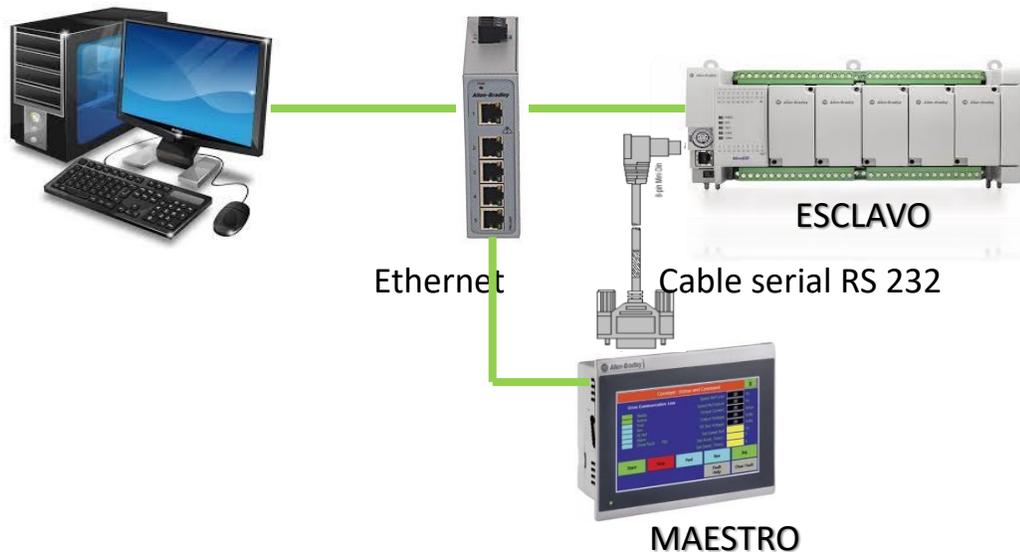
1 Objetivos

- Comunicar un Panel View 800 y un PLC Micro 850 mediante el protocolo Modbus RTU en una red esclavo/maestro.
- Configurar los puertos seriales del controlador y de la Interfaz Hombre-Máquina para la comunicación del protocolo Modbus RTU.
- Realizar el mapeo de las variables del controlador utilizando los códigos de función del protocolo Modbus.

2 Equipos y herramientas:

- Micro 850 - 2080-LC50-48QWB.
- Panel View 800 2711R-T7T.
- Cable Serial: 1761-CBL-PM02.
- Connected Components Workbench.
- RSLinx Classic.
- Cables Ethernet.
- Switch.

3 Conexiones



4 Marco Teórico

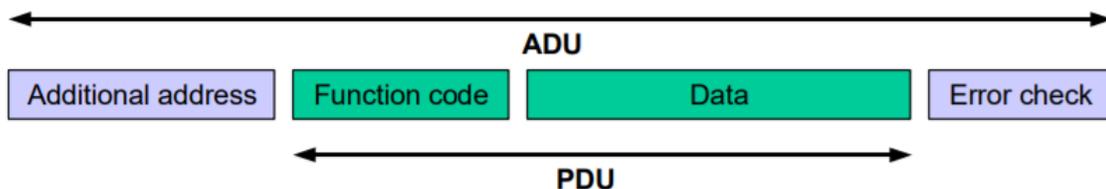
Modbus es un protocolo de comunicación serial que fue desarrollado y publicado por Modicon en 1979. Modbus es un protocolo de enlace, debido a esto puede implementarse con diversos tipos de redes físicas donde normalmente cada fabricante suele suministrar un software de aplicación propio, en la actualidad el protocolo es de acceso libre, lo cual lo ha convertido en un protocolo ampliamente utilizado, con las siguientes características:

- Basado en la arquitectura master/slave para RTU, y client/server para TCP/IP.
- Modos de transmisión serial: ASCII (10 bits), y RTU (11 bits).
- Capas de la red Modbus:

Enlace: se basa en el principio de comunicación master/slave, el cual permite que en cualquier momento se puede calcular el tiempo de transferencia de las solicitudes y las respuestas de cada dispositivo.

Física: puede transmitir a través del RS 232 (semidúplex) y RS 485 (full dúplex), la comunicación es asíncrona y las velocidades de transmisión pueden oscilar entre 75 baudios a 19200 baudios, máxima distancia hasta 1200 m sin repetidores.

La codificación de datos del protocolo Modbus dentro de la Trama puede hacerse en modo ASCII o puramente binario, según el estándar RTU (Remote Transmission Unit). En cualquiera de los dos casos, cada mensaje tiene una trama que contiene cuatro campos principales, según se muestra a continuación.



**Additional Address:* Representa la dirección del esclavo Modbus.

**Function Code:* Se envía el código de función para establecer que se hará con el dato.

**Data:* Es el dato que se envía hacia el esclavo.



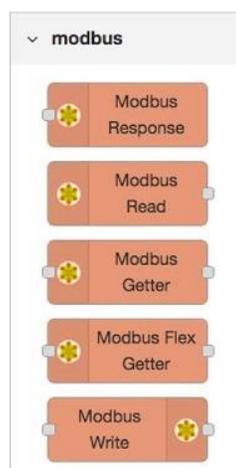
**Error Check:* En esta parte de la trama se obtiene un reconocimiento, para saber si el mensaje fue recibido.

A continuación, se detalla la tabla de códigos de funciones del protocolo Modbus:

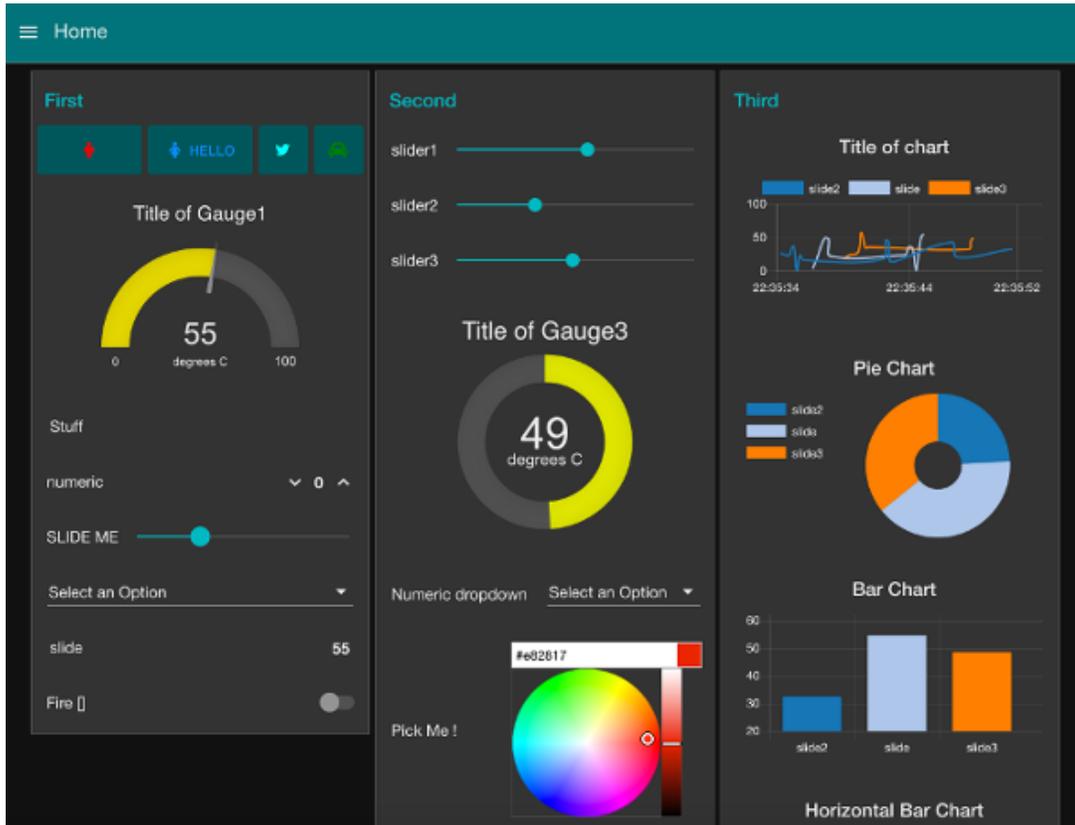
Primary tables	Object type	Type of	Comments
Discretes Input	Single bit	Read-Only	This type of data can be provided by an I/O system.
Coils	Single bit	Read-Write	This type of data can be alterable by an application program.
Input Registers	16-bit word	Read-Only	This type of data can be provided by an I/O system
Holding Registers	16-bit word	Read-Write	This type of data can be alterable by an application program.

Node-RED es una plataforma de código abierto desarrollada para la creación de aplicaciones IoT (Internet de las cosas) de forma visual. Utiliza un enfoque basado en flujos de trabajo (flows) para conectar dispositivos y servicios, permitiendo la integración de sistemas de manera sencilla y eficiente. Opera sobre Node.js, un entorno de ejecución de JavaScript del lado del servidor. La arquitectura de Node-RED se compone de nodos que representan funcionalidades específicas y se conectan mediante flujos. Los nodos pueden ser desde simples funciones hasta integraciones complejas con servicios externos.

Node-RED incluye una librería dedicada para la comunicación Modbus, que simplifica la integración de dispositivos compatibles con este protocolo. La librería ofrece nodos Modbus tanto para maestros (clientes) como para esclavos (servidores), permitiendo la lectura y escritura de registros en dispositivos Modbus TCP y RTU.



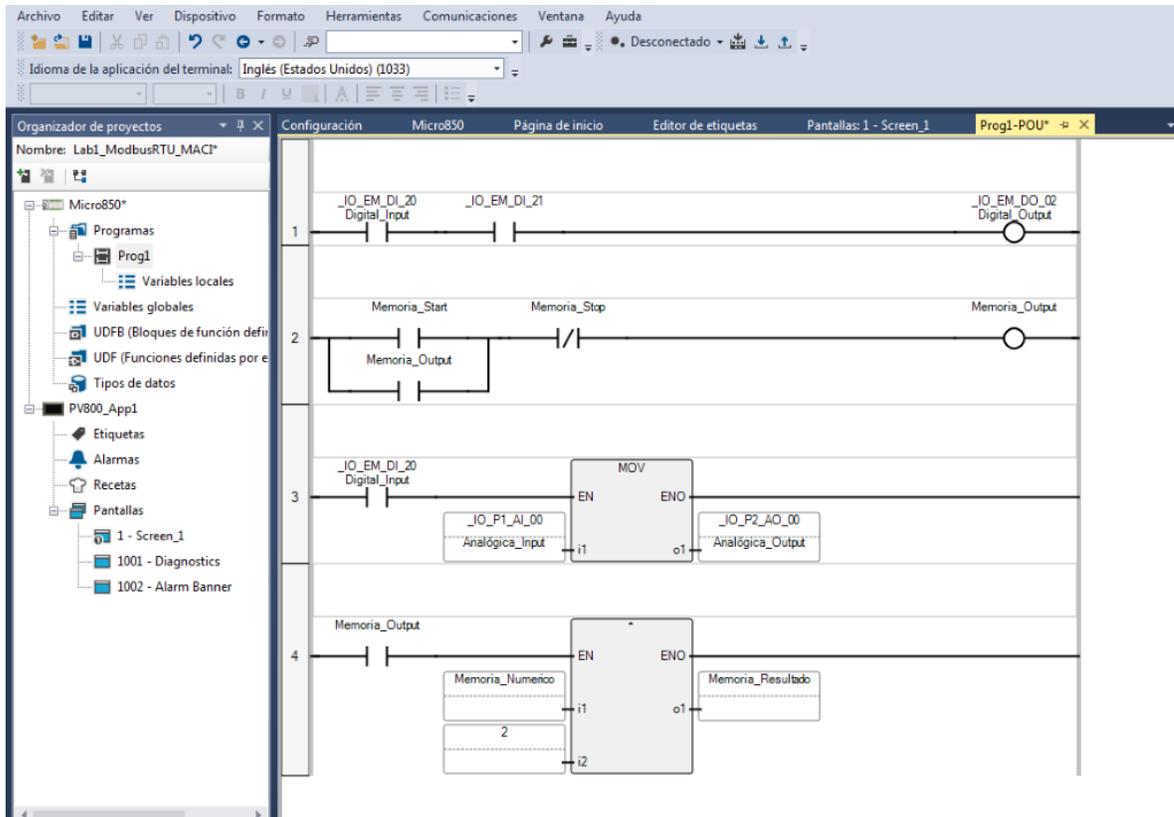
Node-RED Dashboard es una extensión o librería que permite la creación fácil y rápida de interfaces de usuario para las aplicaciones desarrolladas en Node-RED. Proporciona nodos específicos que facilitan la visualización de datos y la interacción con flujos de trabajo de Node-RED a través de paneles de control personalizables.



5. Procedimiento

Configuración del puerto serial del Micro 850 y el esclavo Modbus

1. Abrir el archivo "Practica_modbus", ubicada en la [carpeta compartida](#) del laboratorio. En este proyecto encontraremos dos dispositivos agregados que corresponden al Micro 850 y el Panel View, además, contiene un pequeño lenguaje de programación.



El lenguaje de programación está conformado por siete variables:

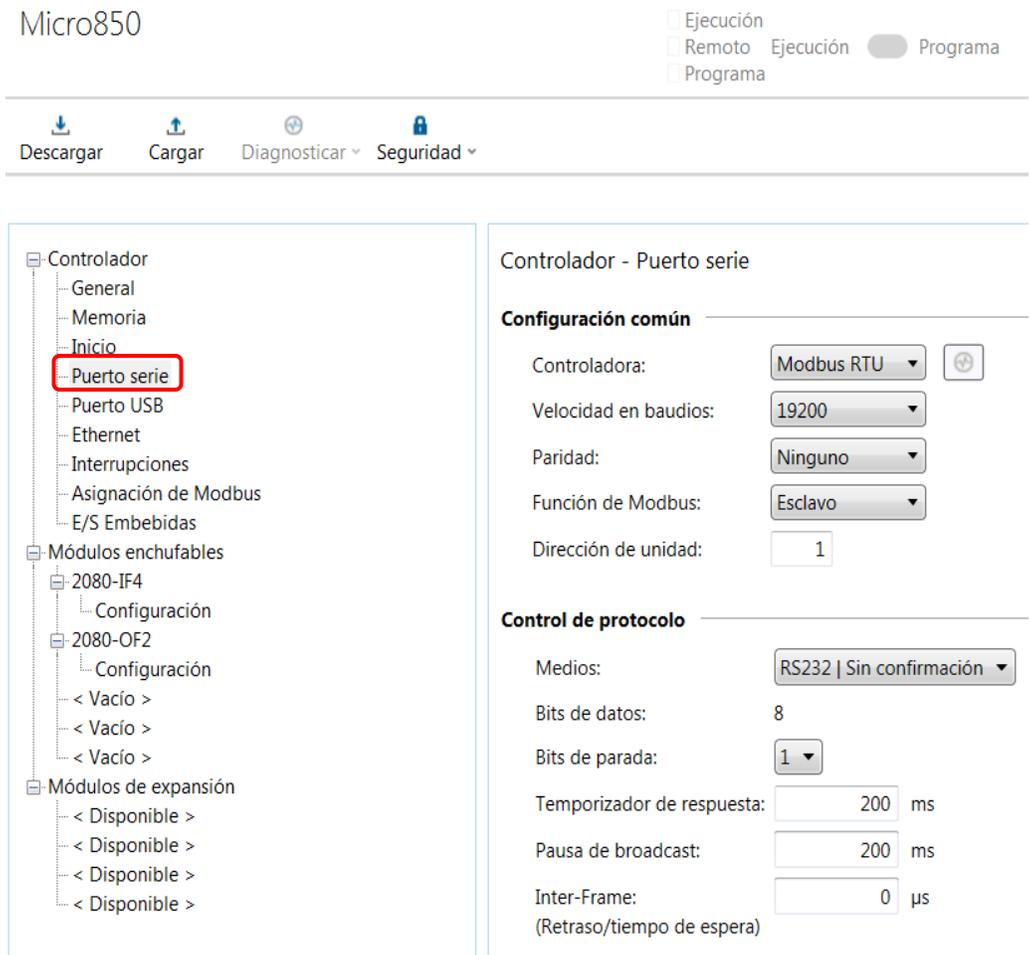
Nombre	Tipo de variable.
Digital Input, <code>_IO_EM_DI_20</code>	Entrada digital física del PLC, switch 20 del tablero.
Digital Output, <code>_IO_EM_DO_02</code>	Salida digital física del PLC, luz piloto verde del tablero.
<code>Memoria_Start</code>	Variable interna tipo booleana del PLC.
<code>Memoria_Stop</code>	Variable interna tipo booleana del PLC.

_IO_P1_AI_00	Entrada analógica física del PLC, potenciómetro del tablero.
_IO_P2_A0_00	Salida analógica física del PLC.

2. Seleccionar el controlador en la ventana del organizador de proyecto.



3. Seleccionar "Puerto serie", y configurar de acuerdo con lo que muestra en la Ilustración.



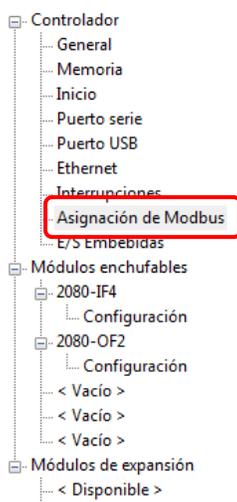
Mapeo Modbus de variables de Micro 850

En esta sección se asociará un código de función Modbus a las variables del controlador, estas variables serán de distintos tipos de datos.

1. Luego, se debe mapear las variables que se van a transmitir del Micro 850 al Panel View, en el menú Modbus Mapping colocar la dirección de las variables de acuerdo con la siguiente tabla.

Address	Range	Data Type	Access
Output Coils	000001-065536	Boolean	Read/Write
Input Coils	100001-165536	Boolean	Read Only
Input Registers	300001-365536	Word (16-bit)	Read Only
Holding Registers	400001-465536	Word (16-bit)	Read/Write

Se muestran las variables asociadas del controlador, las cuales se les ha asignado un código de función, asignar estas variables al controlador.



Controlador - Asignación de Modbus

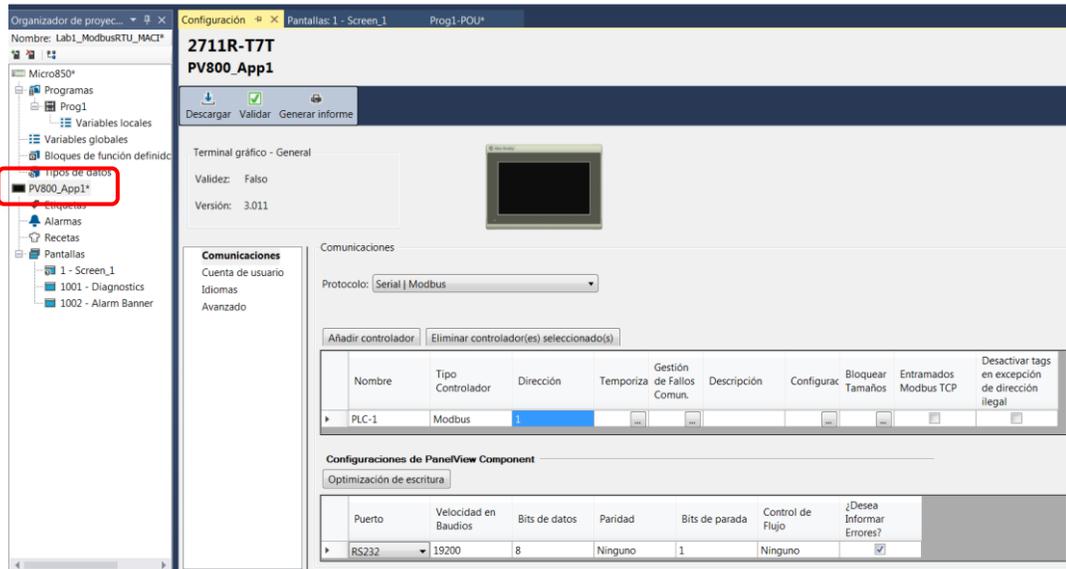
Agregar Duplicar Importar Exportar

Nombre de la variable	Tipo de datos	Dirección	Direcciones usadas
Memoria_Start	BOOL	000001	000001 X
Memoria_Output	BOOL	000002	000002
Memoria_Stop	BOOL	000003	000003
_IO_EM_DI_20	BOOL	100001	100001
_IO_EM_DO_02	BOOL	100002	100002
_IO_P1_AI_00	UINT	300001	300001
_IO_P2_AO_00	UINT	400001	400001
Memoria_Numerico	INT	400017	400017
Memoria_Resultado	INT	400033	400033



Configuración del puerto serial del Panel View

1. Seleccionar el Panel View en la ventana del organizador de proyecto, luego configurar de acuerdo a lo que muestra en la Ilustración.



2. Seleccionar “Etiquetas” en la ventana del organizador de proyecto, para vincular las variables del controlador, se deben asignar los mismos códigos de función Modbus que se vincularon en el Micro 850. Asignar las siguientes etiquetas.

Propiedades Configuración Página de inicio Editor de etiquetas Pantalla

Externo Memoria Sistema Conexiones globales

Añadir Eliminar Deshacer Rehacer

	Nombre de etiqueta	Tipo de datos	Dirección	Controlador	Descripción
	Memoria_Start	Boolean	000001	PLC-1	
	Memoria_Output	Boolean	000002	PLC-1	
	Memoria_Stop	Boolean	000003	PLC-1	
	_IO_EM_DI_20	Boolean	100001	PLC-1	
	_IO_EM_DO_02	Boolean	100002	PLC-1	
	_IO_P1_AI_00	16 bit integer	300001	PLC-1	
	_IO_P2_AO_200	16 bit integer	400001	PLC-1	
	Memoria_Num...	16 bit integer	400017	PLC-1	
	Memoria_Resul...	16 bit integer	400033	PLC-1	

Ilustración 1: Creación de etiquetas en el Panel View



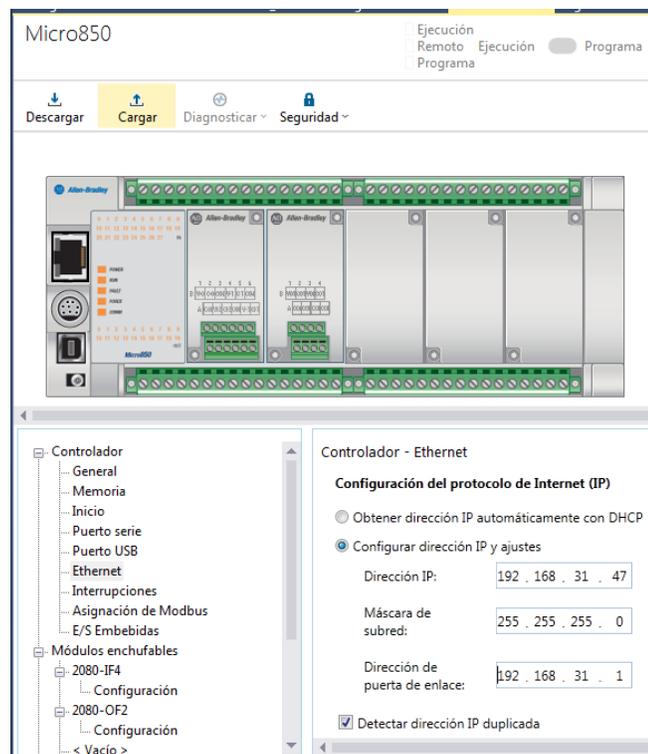
Descarga del proyecto

En esta sección mostraremos como cargar un programa al controlador, y a su vez como cargar el programa realizado al PanelView, cabe mencionar que este proceso se realiza cuando se haya realizado todas las configuraciones antes realizadas, en caso de que exista alguna modificación ya sea del controlador o de la pantalla, se debe cargar el programa al dispositivo respectivamente del cual se realizó el cambio.

1. Seleccionar el controlador en la ventana del organizador de proyecto.



2. Seleccionar "Ethernet", para asignarle una dirección IP al controlador, esta dirección IP se encuentra escrita en cada tablero, la máscara de subred y dirección de puerta de enlace, colocar la misma mostrada en la ilustración.

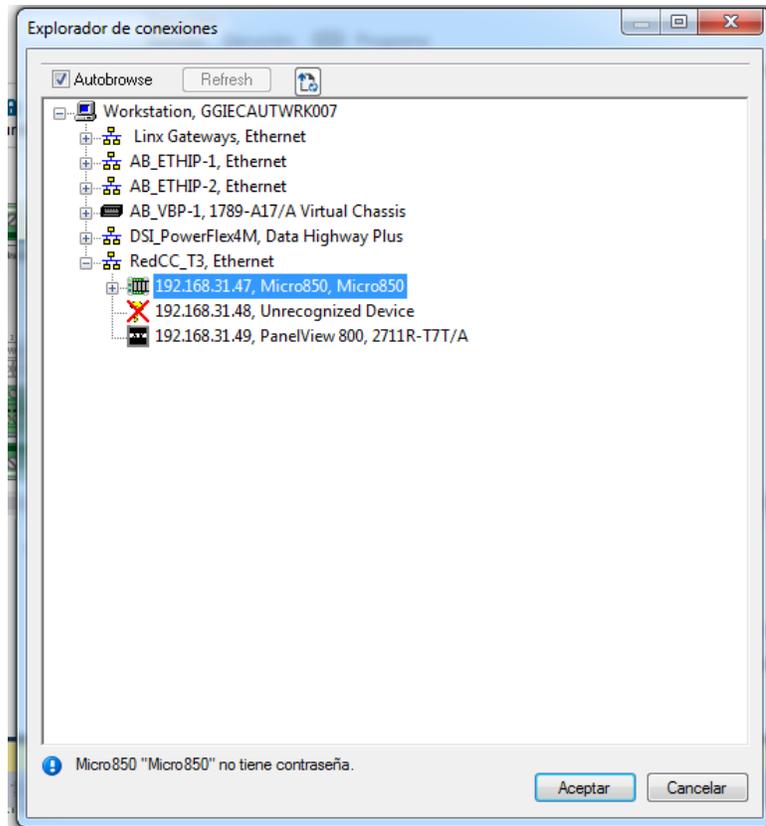


3. Seleccionar al controlador "Micro850" en la ventana "Organizador de proyectos", luego dar clic en el ícono que se indica en la Ilustración 13, el cual nos permitirá descargar el programa al controlador.



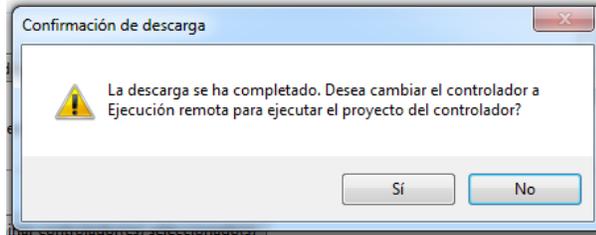


4. Luego, nos aparecerá una ventana similar a RSLinx Classic, donde seleccionaremos la RedCC_T#, desplegaremos su lista, y seleccionaremos al controlador, y damos clic en "Aceptar".

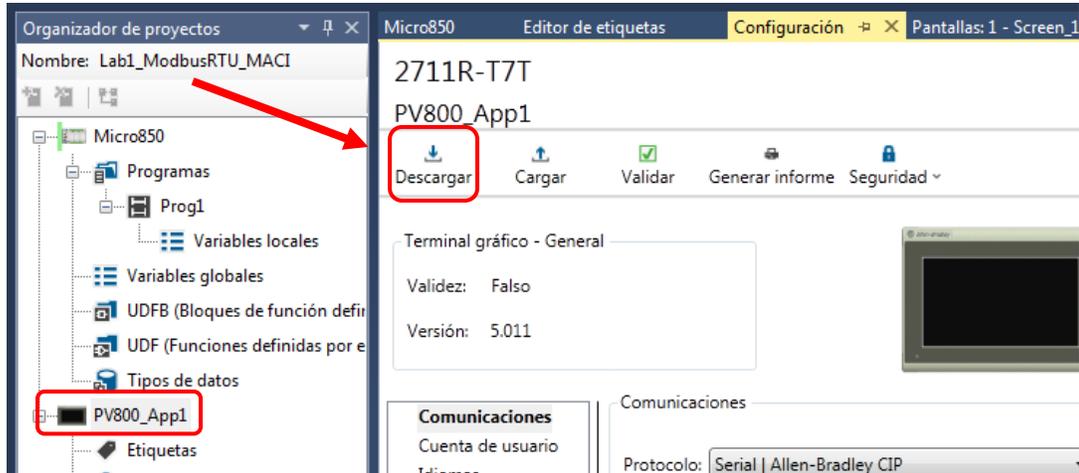


5. En el siguiente cuadro, damos clic en "Sí", en este cuadro nos está sugiriendo si queremos está en modo ejecución, los controladores de Rockwell Automation tienen dos modos cuando se está en el estado de remoto, uno es Modo Programación, el cual nos permite está en línea y a la vez programar; el otro modo es Modo de Ejecución, el cual nos permite ver los cambios de estados de las entradas y salidas del PLC en tiempo real.

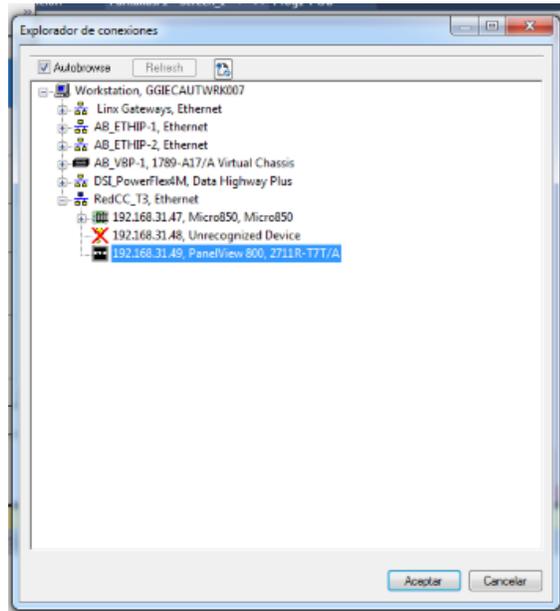




- Luego seleccionamos el PanelView de la ventana de "Organizador de proyectos", y damos clic en descargar.

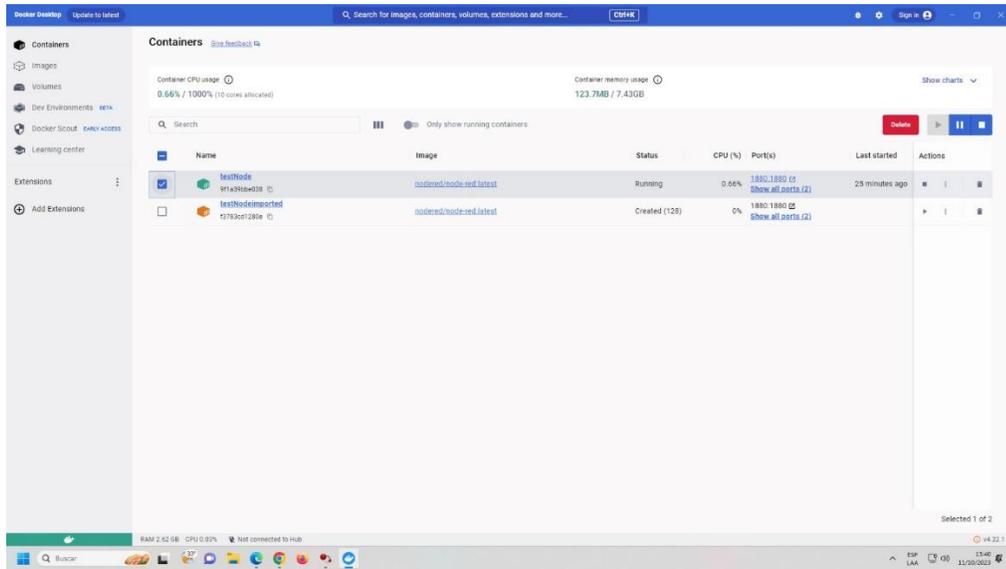


- nos aparecerá una ventana similar a RSLinx Classic, donde seleccionaremos la RedCC_T#, desplegaremos su lista, y seleccionaremos la pantalla, y damos clic en "Aceptar".

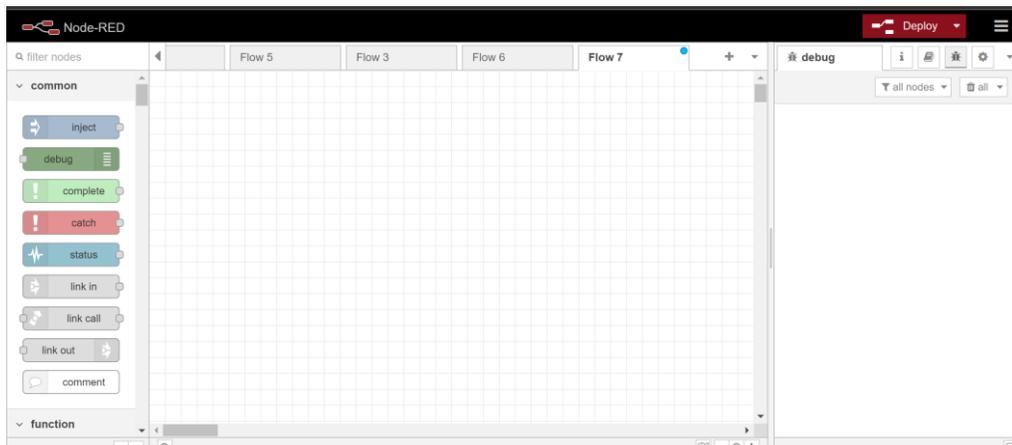


Configuración de nodos para lectura y escritura Modbus en Node RED

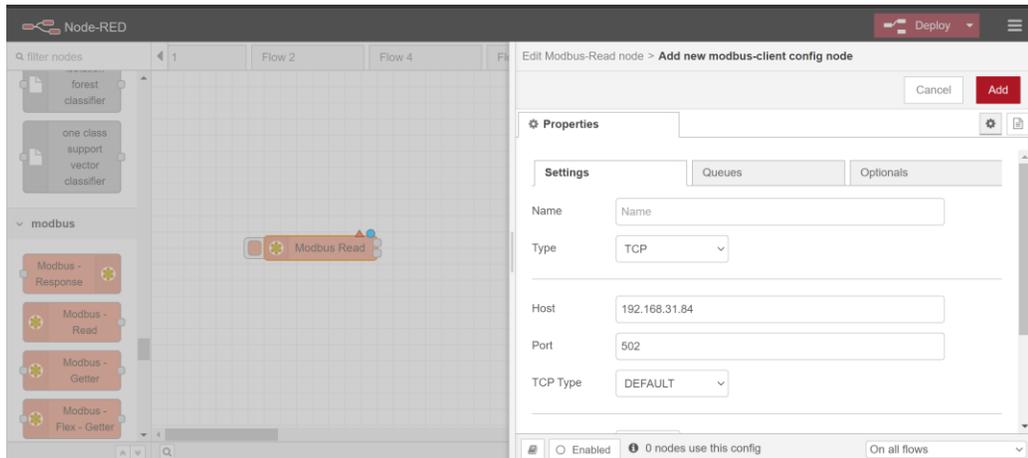
1. Primero se abre la aplicación de Docker para buscar el Container que contiene a la instancia de Node RED e iniciarla.



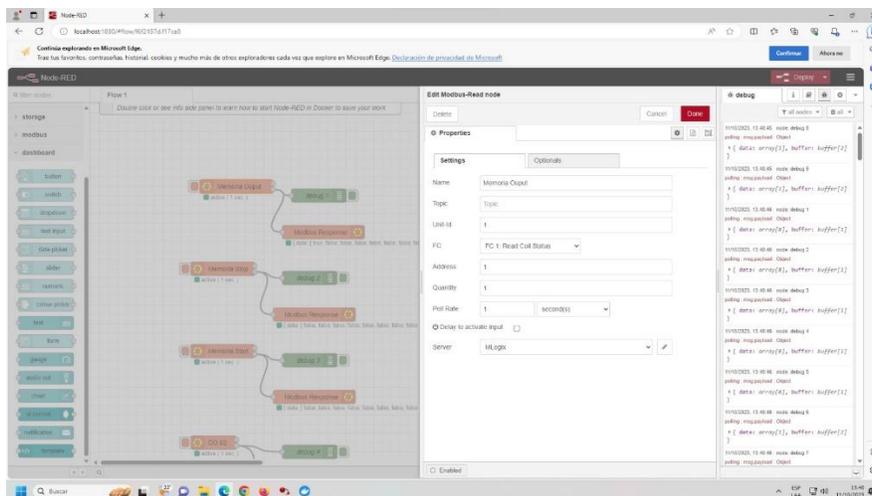
2. Desde algún navegador se coloca localhost:1880 para acceder a la interfaz visual de Node RED.



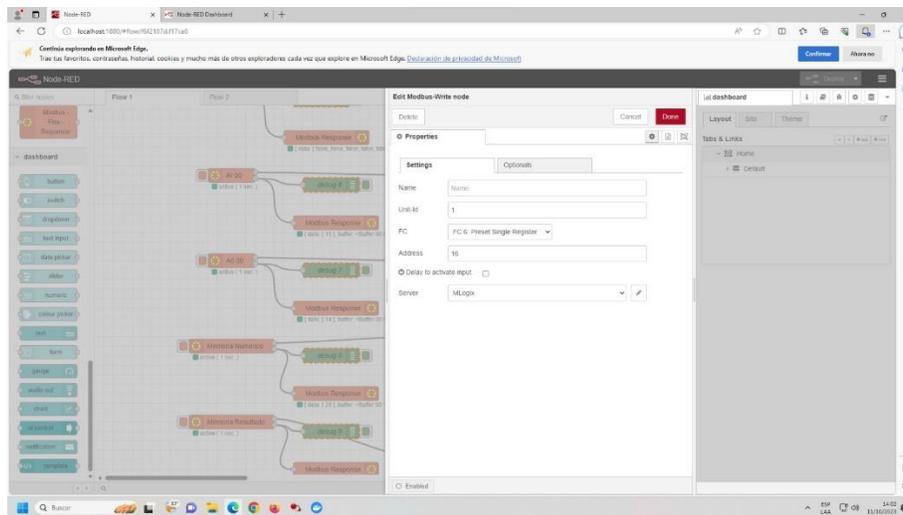
- Se arrastra desde la librería de modbus un nodo modbus read o write. Al editar el nodo se nos solicitará agregar un servidor modbus el cual será nuestro PLC, por lo que, se le detalla su dirección IP y el puerto de acceso.



- En el nodo read y en el write de modbus se especifica: ID Modbus (1), Servidor Modbus, Función/Mapeo Modbus dirección Modbus, cantidad de datos (1) y el cada que tiempo hace la solicitud (1 seg). OJO: Si especificas una dirección n , apuntaras a la dirección $n-1$.
 - FC1 - Read Coils (Leer Bobinas): Utilizada para leer el estado de bobinas discretas (bits de salida) en un dispositivo esclavo.
 - FC2 - Read Discrete Inputs (Leer Entradas Discretas): Permite la lectura de estados de entradas discretas (bits de entrada) en un dispositivo esclavo.
 - FC3 - Read Holding Registers (Leer Registros de Retención): Utilizada para la lectura de datos almacenados en registros de retención en un dispositivo esclavo.
 - FC4 - Read Input Registers (Leer Registros de Entrada): Permite la lectura de datos almacenados en registros de entrada en un dispositivo esclavo.



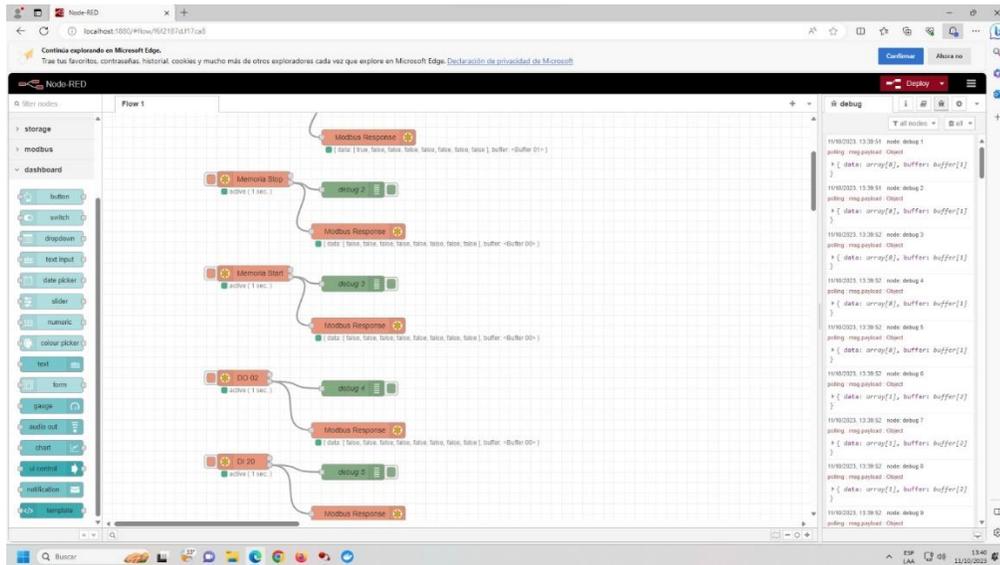
- F 5 - Write Single Coil (Escribir Una Bobina): Esta función se utiliza para escribir un solo bit (coil) en un dispositivo esclavo. Especifica la dirección de la bobina que se va a escribir y el valor (ON/OFF) que se desea asignar. Es importante destacar que esta función está diseñada para modificar un solo bit a la vez.
- F 6 - Write Single Register (Escribir un Solo Registro): La función 6 se emplea para escribir un valor en un solo registro de retención en un dispositivo esclavo. Se debe proporcionar la dirección del registro y el valor que se desea escribir. Al igual que la función 5, esta función está diseñada para modificar un solo registro a la vez.
- F 15 - Write Multiple Coils (Escribir Múltiples Bobinas): La función 15 permite la escritura simultánea de múltiples bits (bobinas) en un dispositivo esclavo. Se especifica la dirección inicial de las bobinas y se proporciona una serie de valores (ON/OFF) para escribir en esas bobinas. Esto reduce la cantidad de mensajes necesarios para realizar múltiples escrituras.
- F 16 - Write Multiple Registers (Escribir Múltiples Registros): Similar a la función 15, la función 16 permite la escritura simultánea de múltiples valores en registros de retención en un dispositivo esclavo. Se especifica la dirección inicial del primer registro y se proporciona una serie de valores para escribir en los registros consecutivos.



5. La respuesta de la solicitud las podemos tener con el nodo modbus response, donde se coloca el numero de elementos que se espera recibir. Deben colocar los nodos para poder

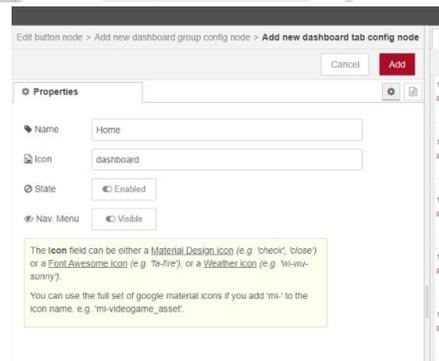
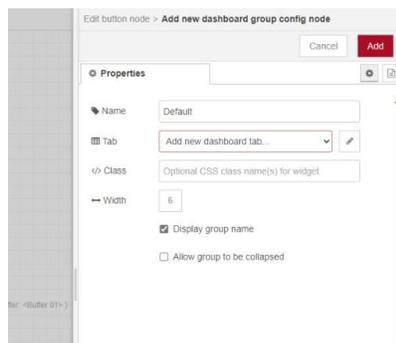
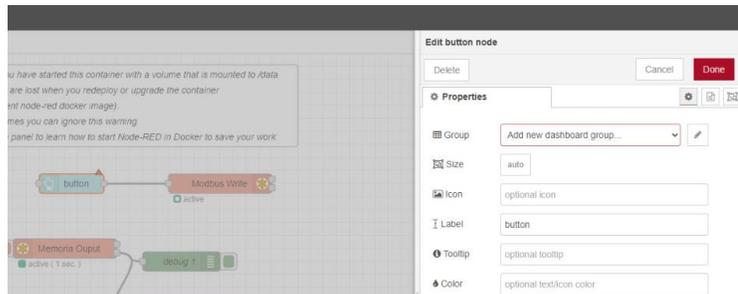


leer las entradas y salidas digitales y analógicas del programa. Para el caso de Valor Numérico debe ser un nodo de escritura para editar mediante un Slider.



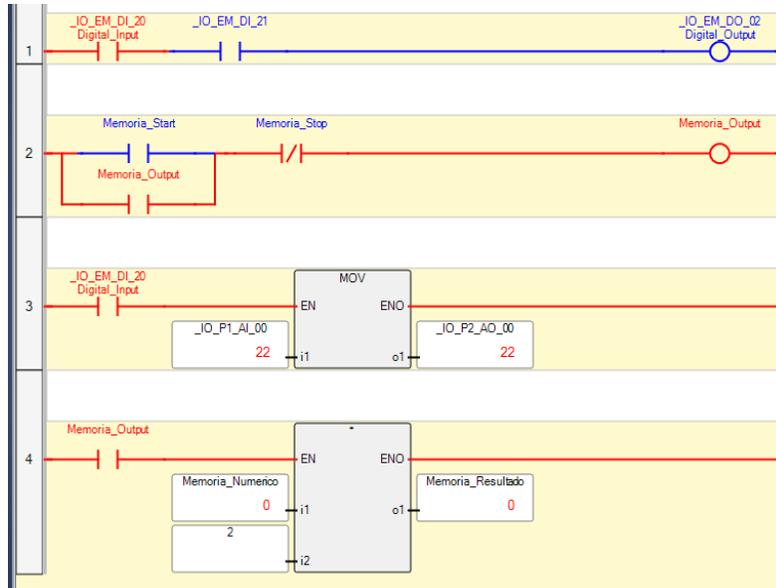
Preparar Dashboard

1. Al agregar algún nodo de la librería Dashboard, se requerirá crear un grupo o pagina para colocar los elementos.



2. Se enlazan los display y graficas como Gauge a la primera salida de los nodos Modbus Read correspondientes. Y para los elementos de interacción como Slider y Switch se colocan a la entrada de los Modbus Write.





LAB 1: PROTOCOLO MODBUS RTU

**Variables tipo memoria del controlador
(Output coils)**

START
Press

STOP
Press

OUT

Entada numérica

0

Salida numérica

**Variables físicas del controlador
(Input coils)**

DI_20

DO_02

Entada digital

0

Salida digital

0

Entada analógica

0

Salida analógica

0

Salir



Default

Stop

Start

Valor Numerico



0 28 units 100

Valor Resultante



0 56 units 100

Set

Analog I Channel 15

Analog O Channel 14

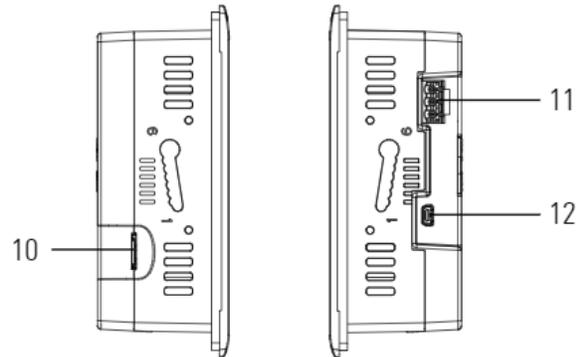
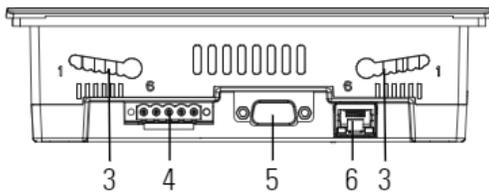
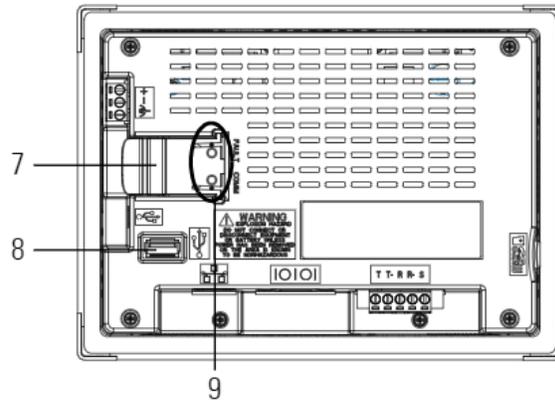
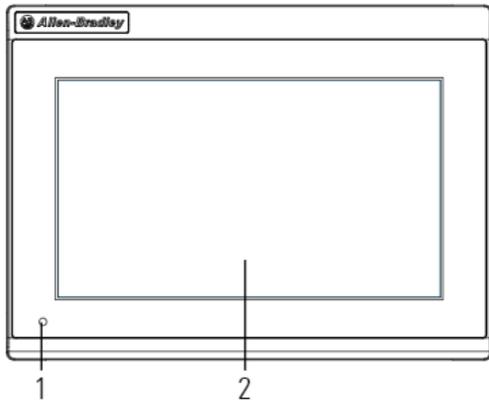


ANEXOS

Panel View

[https://literature.rockwellautomation.com/idc/groups/literature/documents/td/2711r-td001 - en-p.pdf](https://literature.rockwellautomation.com/idc/groups/literature/documents/td/2711r-td001-en-p.pdf)

PanelView 800 Terminal – 2711R-T7T



Item	Description	Item	Description
1	Power status LED ⁽¹⁾	7	Replaceable real-time clock battery
2	Touch display	8	USB host port
3	Mounting slots	9	Diagnostic status indicator
4	RS-422 and RS-485 port	10	Micro-SD (Secure Digital) card slot
5	RS-232 port	11	24V DC power input
6	10/100 Mbit Ethernet port	12	USB device port ⁽²⁾

(1) The Power Status LED is red when in screen saver or dimmer mode and green when in normal (operational) mode.

(2) The USB device port is not intended for Customer use.

Table 4 - Technical Specifications - PanelView 800 Terminals

Attribute	2711R-T4T	2711R-T7T	2711R-T10T
Display type	Color transmissive TFT active matrix LCD, widescreen format		
Display size	4 in.	7 in.	10 in.
Display area (WxH)	95 x 53.9 mm (3.74 x 2.12 in.)	153.6 x 86.6 mm (6.05 x 3.41 in.)	211.2 x 158.4 mm (8.31 x 6.24 in.)
Resolution (pixels)	480 x 272	800 x 480	800 x 600
Backlight	LED		
Backlight lifespan, min	40,000 hours		
Colors	65K colors		
Operator input	Analog touch and function keys	Analog touch	
Memory card	USB port and micro-SD (Secure Digital) card – Industrial grade micro-SD cards recommended. Supports SDSC and Class 6 & Class 10 SDHC micro-SD cards, FAT32/16 formats, up to 32 GB maximum size.		
Real-time clock with battery	Yes		
Battery lifespan, min	5 years @ 25 °C (77 °F)		
Programming port	Ethernet port		
Recipe	50 recipe files		
Software	Connected Components Workbench software, release 8.0 and later		
Preferred controller	Micro800, MicroLogix		
Power Supply	24V DC		
Processor, CPU speed	800 MHz CPU		
Flash Memory (ROM) min	256 MB		
SDRAM (RAM) min	256 MB DDR		
Operating System	WINCE 6.0		
Power Consumption (max)	9 W (0.39 A @ 24V DC)	11 W (0.40 A @ 24V DC)	14 W (0.48 A @ 24V DC)
RS232/RS422/485 (isolated)	Separate RS-232 and RS422/RS485 connectors		
Ethernet 10/100 Mbps	1		
USB Host (USDB 2.0)	Yes		
MicroSD slot	Yes		
Input voltage range	18...32V DC (24V DC nom)		
Weight, approx	0.333 kg (0.73 lb)	0.651 kg (1.44 lb)	1.64 kg (3.62 lb)
Dimensions (HxWxD), approx	116 x 138 x 43 mm 4.6 x 5.4 x 1.7 in.	144 x 197 x 54 mm 5.7 x 7.8 x 2.1 in.	225 x 287 x 55 mm 8.9 x 11.3 x 2.2 in.
Panel Cutout dimensions (HxW)	99 x 119 mm 3.9 x 4.7 in.	125 x 179 mm 4.9 x 7.1 in.	206 x 269 mm 8.1 x 10.6 in.
Front Bezel Protection	IP65, NEMA 4X, 12, 13		



Table 11 - Protocol

Protocol	PanelView 800 Port	Micro820	Micro830	Micro850	Micro800 Plug-in Module (2080-SERIALISOL)
Modbus (RTU) (HD)	RS-232	_(2)	1761-CBL-PM02	1761-CBL-PM02	_(2)
	RS-485 ⁽¹⁾	_(2)	_(2)	_(2)	_(2)
Modbus unsolicited (HD)	RS-232	_(2)	1761-CBL-PM02	1761-CBL-PM02	_(2)
	RS-485 ⁽¹⁾	_(2)	_(2)	_(2)	_(2)
Ethernet (AB CIP)	Ethernet	2711P-CBL-EX04 (CAT5 Ethernet)	N/A ⁽³⁾	2711P-CBL-EX04 (CAT5 Ethernet)	N/A ⁽³⁾
Modbus TCP	Ethernet	2711P-CBL-EX04 (CAT5 Ethernet)	N/A ⁽³⁾	2711P-CBL-EX04 (CAT5 Ethernet)	N/A ⁽³⁾
Serial (AB CIP)	RS-232	_(2)	1761-CBL-PM02	1761-CBL-PM02	_(2)

(1) RS-485 is isolated and is recommended for connecting to only one device with an isolated port.

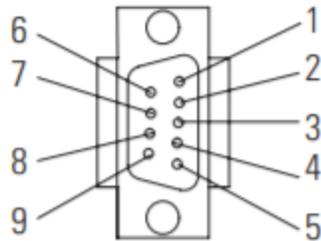
(2) This controller includes a terminal block and requires wiring.

(3) This connectivity is not available for this controller.

1761-CBL-PM02



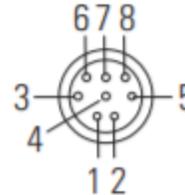
DB-9 RS-232
Port 1



1761-CBL-AP00 or 1761-CBL-PM02



Cable Straight-D Connector
Port 2



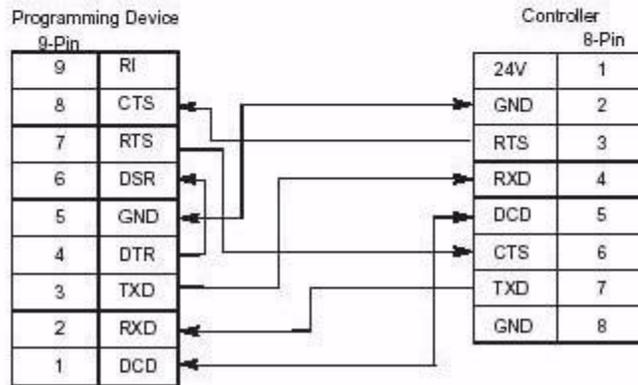
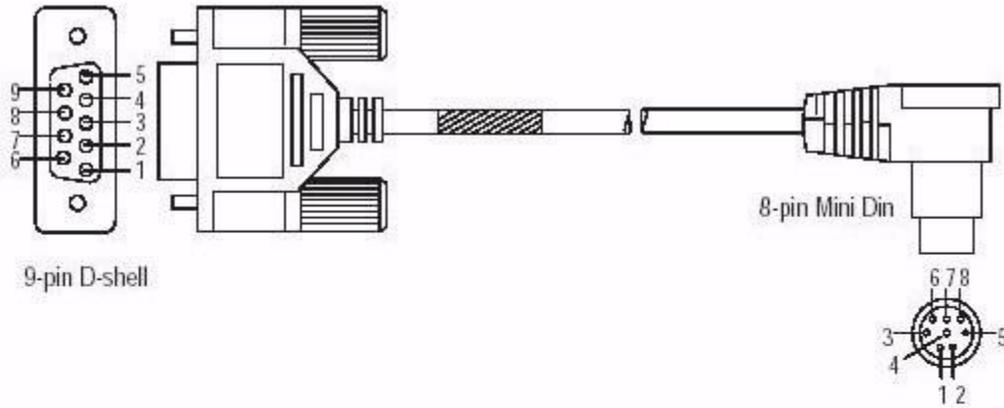
Ports

Pin #	DB-9 RS-232	RS-232 (8-pin mini-DIN) Communication Port ⁽¹⁾ (1761-CBL-PM02 cable)	Port 3 DH-485 Connector
1	Received line signal detector (DCD)	Not applicable	Chassis ground
2	Received data (RxD)	Signal common (GRD)	Cable shield
3	Transmitted data (TxD)	Request to send (RTS)	Signal ground
4	DTE ready (DTR)	Received data (RxD)	DH-485 data B
5	Signal common (GRD)	Same state as port 1's DCD signal	DH-485 data A
6	DCE ready (DSR)	Clear to send (CTS)	Termination
7	Request to send (RTS)	Transmitted data (TxD)	Not applicable
8	Clear to send (CTS)	Not applicable	Not applicable
9	Not applicable	Not applicable	Not applicable

⁽¹⁾ An 8-pin mini-DIN connector is used for making connections to the RS-232 (8-pin mini-DIN) communication port. This connector is not commercially available. If you are making a cable to connect to the RS-232 (8-pin mini-DIN) communication port, you must configure your cable to connect to the Allen-Bradley cable.



1761-CBL-PM02 Series B Cable



1761-CBL-AP00 and 1761-CBL-PM02 Cable



1761-CBL-AP00 and 1761-CBL-PM02 Cable

Cable	Length	Connects		External Power Supply Required	Selection Switch Setting
		from	to AIC+ Interface Converter		
1761-CBL-AP00, 1761-CBL-PM02 ⁽¹⁾	45 cm, (17.7 in.), 2 m (6.5 ft)	SLC 5/03, SLC 5/04, or SLC 5/05 processors, channel 0	RS-232 (8-pin mini-DIN) communication port	Yes	External
		MicroLogix 1000, 1200, and 1500 processors	RS-232 (DB-9, DTE) Communication Port	Yes ⁽²⁾	External
		PanelView through NULL modem adapter	RS-232 (8-pin mini-DIN) communication port	Yes	External
		PC com port	RS-232 (8-pin mini-DIN) communication port	Yes	External

⁽¹⁾ Series B cables are required for hardware handshaking.

⁽²⁾ An external power supply is required unless the AIC+ interface converter is powered by a MicroLogix controller connected to the RS-232 (8-pin mini-DIN) communication port with a 1761-CBL-AM00 or 1761-CBL-HM02 or equivalent cable.

