

# EtherNet/IP™

Protocolo de comunicación del indicador PUE HY10

## INSTRUCCIONES DE SOFTWARE

ITKU-27-01-06-20-ES





**EtherNet/IP™ es una marca comercial de ODVA, Inc.**

JUNIO 2020

# INDICE

<b>1. ESTRUCTURA DE DATOS .....</b>	<b>4</b>
1.1. La dirección de entradas .....	4
1.1.1. Lista de la variable de entrada .....	4
1.1.2. Descripción de registros de entrada .....	5
1.2. La dirección de salida .....	7
1.2.1. Lista de la variable de entrada .....	7
1.2.2. Descripción de registros de salida .....	7
<b>2. CONFIGURACIÓN DEL MÓDULO EtherNet/IP™ EN UN ENTORNO RS LOGIX .....</b>	<b>10</b>
2.1. CONFIGURACIÓN RSLinx .....	10
2.2. Proyecto RSLogix .....	11

# 1. ESTRUCTURA DE DATOS

## 1.1. La dirección de entradas

### 1.1.1. Lista de la variable de entrada

Variable	Offset	Longitud [WORD]	Tipo de datos
Masa plataforma 1	0	2	float
Tara de plataforma 1	4	2	float
Unidad de la plataforma 1	8	1	integer
Estado de la plataforma 1	10	1	integer
Umbral Lo de plataforma 1	12	2	float
Masa plataforma 2	16	2	float
Tara de plataforma 2	20	2	float
Unidad de la plataforma 2	24	1	integer
Estado de la plataforma 2	26	1	integer
Umbral Lo de plataforma 2	28	2	float
Masa de la plataforma 3	32	2	float
Tara de plataforma 3	36	2	float
Unidad de la plataforma 3	40	1	integer
Estado de la plataforma 3	42	1	integer
Umbral Lo de plataforma 3	44	2	float
Masa de la plataforma 4	48	2	float
Tara de plataforma 4	52	2	float
Unidad de la plataforma 4	56	1	integer
Estado de la plataforma 4	58	1	integer
Umbral Lo de plataforma 4	60	2	float
Estado del proceso (detener, iniciar)	64	1	integer
Estado entradas	66	1	integer
MIN	68	2	float
Máx.	72	2	float
Número de serie	84	2	long
Usuario	88	1	integer
Producto	90	1	integer
Cliente	92	1	integer
Embalaje	94	1	integer
Almacén de origen	96	1	integer
Almacén de destino	98	1	integer
Recatas/Proceso de dosificación	100	1	integer

### 1.1.2. Descripción de registros de entrada

**Masa de la plataforma** - el valor de la masa se devuelve en la unidad actual

**Tara de plataforma** - el valor de tara se devuelve en la unidad de calibración

**Unidad de la plataforma** – determina la unidad de masa actual de la plataforma dada (visualizada)

Bit de la unidad	
0	gramo [g]
1	kilogramo [kg]
2	ct (quilates),
3	lb (libra)*,
4	oz (uncia)*,
5	N (Newton).

#### Ejemplo:

Valor de lectura HEX 0x02. Forma binaria:

B1/7	B1/6	B1/5	B1/4	B1/3	B1/2	B1/1	B1/0	B0/7	B0/6	B0/5	B0/4	B0/3	B0/2	B0/1	B0/0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

La unidad de peso es kilogramo [kg].

**Estado de la plataforma** – determina el estado de una plataforma de pesaje dada.

Bit del estado	
0	Medición correcta (la balanza no informa de un error)
1	Medición estable
2	Balanza está en cero
3	Balanza está tarado
4	Balanza está en el segundo rango
5	Balanza está en el tercer rango
6	Balanza informa un error NULL
7	Balanza informa un error LH
8	Balanza informa un error FULL

#### Ejemplo:

Valor HEX leído:0x13

B1/7	B1/6	B1/5	B1/4	B1/3	B1/2	B1/1	B1/0	B0/7	B0/6	B0/5	B0/4	B0/3	B0/2	B0/1	B0/0
0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1

La balanza no informa un error, medición estable en el segundo rango.

**Umbral LO** - devuelve el valor umbral **LO** en la unidad de calibración de la plataforma dada.

**Estado del proceso** – determina el estado del proceso de dosificación o receta

0x00 - proceso inactivo

0x01 – proceso en ejecución

0x02 – proceso interrumpido

0x03 – proceso completo

**Estado de entradas**-máscara de bits de las entradas del indicador. Los primeros 4 bits más bajos representan las entradas del terminal de pesaje.

**Ejemplo:**

Valor HEX leído:0x000B

B1/7	B1/6	B1/5	B1/4	B1/3	B1/2	B1/1	B1/0	B0/7	B0/6	B0/5	B0/4	B0/3	B0/2	B0/1	B0/0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1

Las entradas 1, 2 y 3 en el terminal de pesaje están en estado alto.

**MIN** – devuelve el valor del umbral **MIN** establecido en la unidad actual.

**MAX** – devuelve el valor del umbral **MAX** establecido en la unidad actual.

**Número de serie** – devuelve el valor del número de serie.¡ Solo se aceptan valores numéricos! Todos los caracteres anteriores se omiten.

**Usuario** – devuelve el valor del código de usuario registrado.

**Producto** – devuelve el valor del código del producto seleccionado

**Cliente** – devuelve el valor del código del Cliente seleccionado

**Embalaje**– devuelve el valor del código del embalaje seleccionado

**Almacén de origen**– devuelve el valor del código del almacén de origen

**Almacén de destino**– devuelve el valor del código del almacén de destino

**Receta/proceso de dosificación**– devuelve el valor del código de la receta.

**Proceso de dosificación**– devuelve el valor del código del proceso de dosificación seleccionado

## 1.2. La dirección de salida

### 1.2.1. Lista de la variable de entrada

Variable	Offset	Longitud [WORD]	Tipo de datos
Comando	0	1	integer
Comando con parámetro	2	1	integer
Plataforma	4	1	integer
Tara	6	2	float
Umbral LO	10	2	float
Estado de salidas	14	1	integer
Min	16	2	float
Máx	20	2	float
Número de serie	32	2	long
Usuario	36	1	integer
Producto	38	1	integer
Cliente	40	1	integer
Embalaje	42	1	integer
Almacén de origen	44	1	integer
Almacén de destino	46	1	integer
Recatas/Proceso de dosificación	48	1	integer

### 1.2.2. Descripción de registros de salida

**Comando básico:** guardar el registro con un valor apropiado activará las siguientes acciones:


Número de bit	Acción
0	Puesta a cero de la plataforma
1	Tara la plataforma
2	Estadísticas claras
3	Guardar/Imprimir
4	Inicio del proceso
5	Detener el proceso

#### Ejemplo:

Escribir el registro con el valor 0x02



B1/7	B1/6	B1/5	B1/4	B1/3	B1/2	B1/1	B1/0	B0/7	B0/6	B0/5	B0/4	B0/3	B0/2	B0/1	B0/0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Esto tarar la balanza.

	<b><i>El comando se ejecuta una vez, después de que se detecta la configuración de su bit. Si es necesario volver a ejecutar el comando con el mismo conjunto de bits, primero se debe borrar y luego volver a establecer el valor deseado.</i></b>
---	---

**Comando complejo** -establecer el bit de comando apropiado realiza la tarea directamente de acuerdo con la tabla:

Número de bit	Acción
0	Ajustar el valor de tara para la plataforma dada
1	Establecer el valor umbral de LO para una plataforma dada
2	Ajuste del estado de la salida
3	Configuración el valor umbral MIN
4	Configuración el valor umbral MAX

	<b><i>El comando compuesto requiere la configuración del parámetro adecuado (la dirección de 4 a 50- ver la tabla de registros de salida).</i></b>
	<b><i>Un comando con un parámetro se ejecuta una vez después de detectar el ajuste de su bit dado. Si es necesario volver a ejecutar el comando con el mismo conjunto de bits, primero se debe borrar y luego volver a establecer el valor deseado.</i></b>

### Ejemplo:

Enviar a una tara con valor 1.0 para la 1ra plataforma.

La ejecución del comando requiere guardar 3 registros:

offset 2 – comando con un parámetro - valor 0x01 - es decir, establecer la tara.

offset 4 – número de la plataforma de pesaje a la que queremos asignar una tara - valor 0x01 para la primera plataforma.

offset 6 – valor de tara en formato flotante - 1.0.

**Plataforma** – parámetro de comando compuesto: número de plataforma de pesaje (1 o 2).

**Tara** – parámetro de comando compuesto: valor de tara (en la unidad de calibración)



**Umbral LO** – parámetro de comando compuesto valor de umbral LO (en la unidad de calibración)

**Estado de salidas** – parámetro de comando compuesto: definir el estado del indicador de pesaje y las salidas del módulo de comunicación.

**Ejemplo:**

Ajuste de estado alto de las salidas 1 y 3 del terminal de pesaje.

La máscara de las salidas será:

B1/7	B1/6	B1/5	B1/4	B1/3	B1/2	B1/1	B1/0	B0/7	B0/6	B0/5	B0/4	B0/3	B0/2	B0/1	B0/0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Después de convertir a HEX, obtenemos 0x05.

La ejecución del comando requiere guardar 2 registros:

offset 2 – comando con un parámetro - valor 0x08 - es decir, guardar el estado de las salidas.

offset 14 – máscara de salida 0x05.

Como resultado, las salidas 1 y 3 se establecerán en estado alto.

**Mini** – parámetro de comando compuesto: el valor del umbral MIN( en la unidad del modo de trabajo actual usado).

**MAX** – parámetro de comando compuesto: el valor del umbral MAX( en la unidad del modo de trabajo actual usado).

**Numero de serie** – parámetro de comando compuesto: valor de número de serie ¡Solo se aceptan valores numéricos! Todos los caracteres anteriores se omiten.

**Usuario** – parámetro de comando compuesto: código de usuario (solo numérico).

**Producto** – parámetro de comando compuesto: código de producto (solo numérico).

**Cliente** – parámetro de comando compuesto: código del cliente (solo numérico).

**Embalaje** – parámetro de comando compuesto: código del embalaje (solo numérico).

**Almacén de origen** – parámetro de comando compuesto: código del almacén de origen (solo numérico).

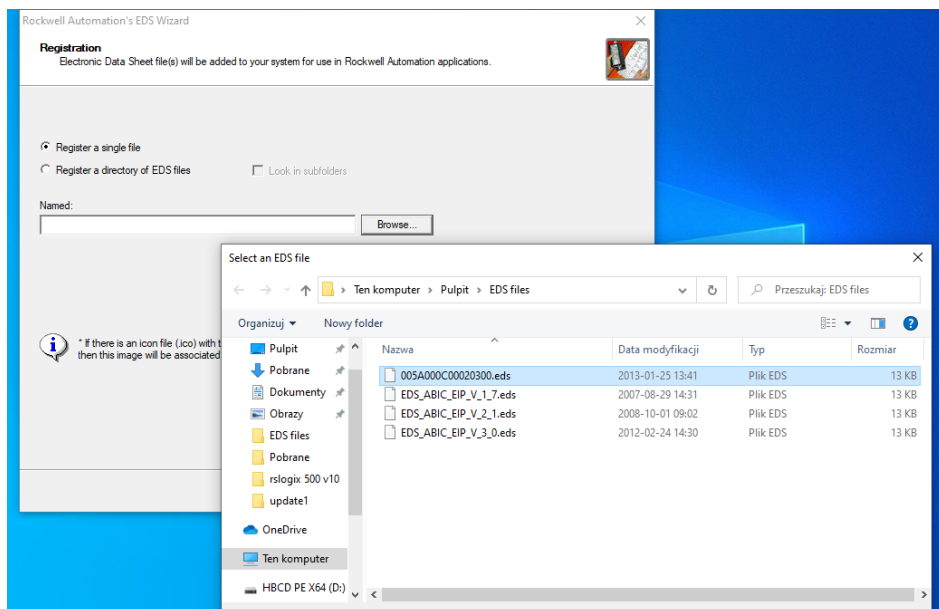
**Almacén de destino** – parámetro de comando compuesto: código del almacén de destino (solo numérico).

**Recetas/ proceso de dosificación** – parámetro de comando compuesto: código de receta (solo numérico).

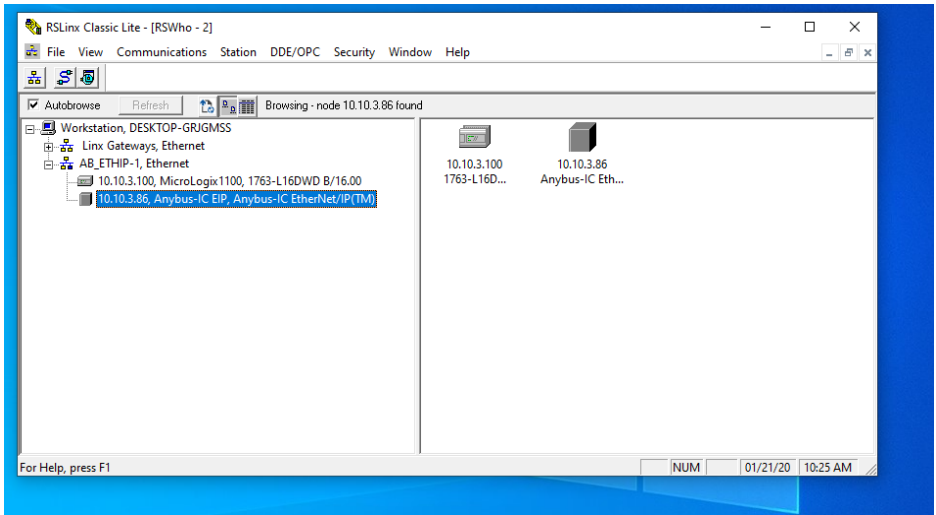
## 2. CONFIGURACIÓN DEL MÓDULO EtherNet/IP™ EN UN ENTORNO RS LOGIX

### 2.1. CONFIGURACIÓN RSLinx

El trabajo en el entorno debe comenzar con la configuración del dispositivo en el software RSLinx. Para hacer esto, agregue el módulo EtherNet / IP de balanza usando el archivo EDS y la herramienta de instalación de hardware EDS.

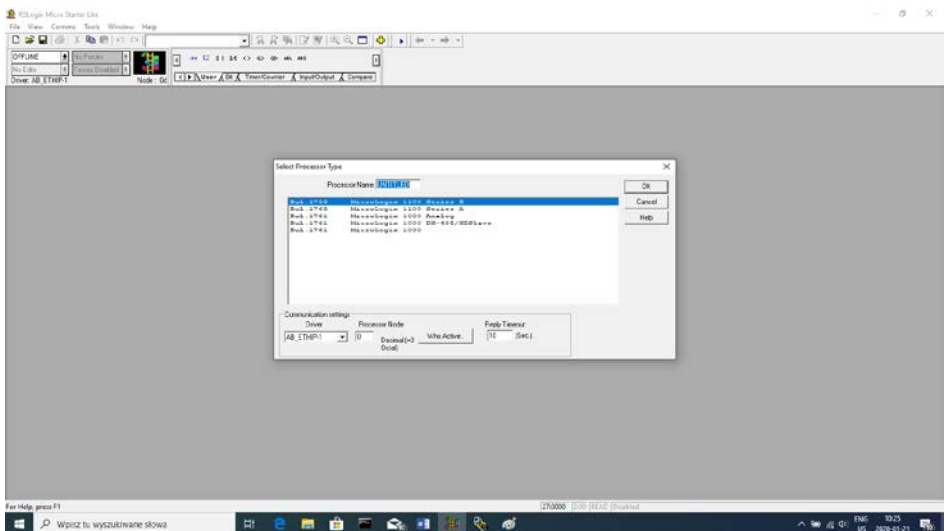


Después de conectar la balanza y el controlador maestro a la red a la red (asegúrese de que todos los dispositivos y la PC estén en la misma subred), deben estar visibles como se muestra en la siguiente figura.



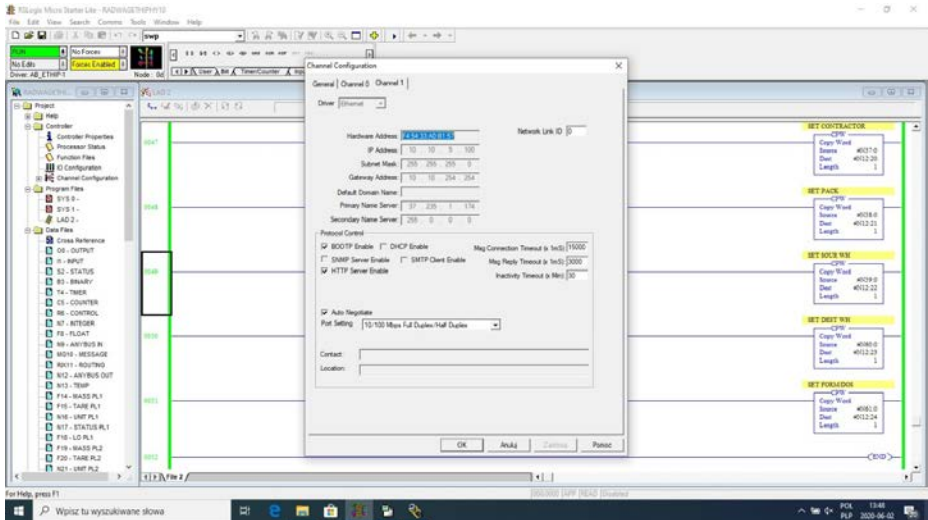
## 2.2. Proyecto RSLogix

Empezamos a trabajar en el medio ambiente iniciando un nuevo proyecto. Seleccione el PLC que se comunicará con la balanza en la ventana de selección del controlador.



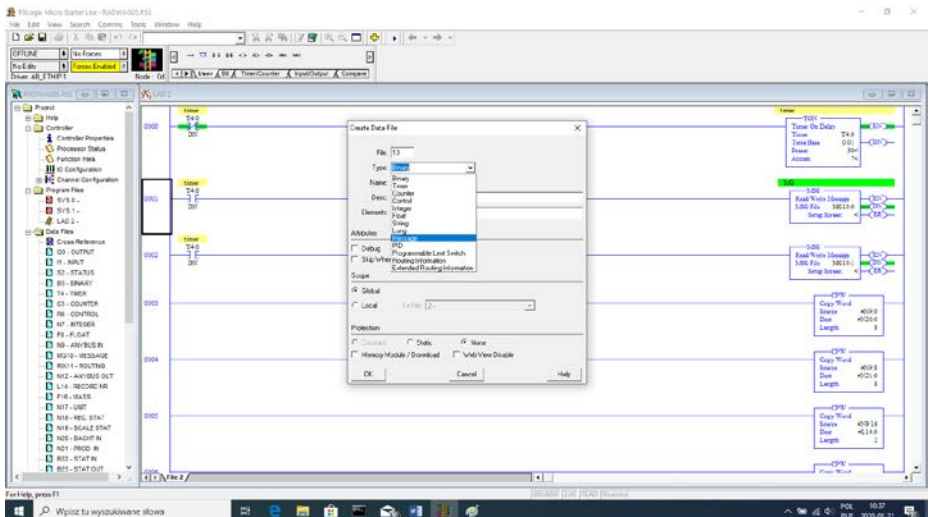
Después de confirmar la selección, vaya a la ventana del proyecto. Luego configure la interfaz de comunicación del controlador. Para ello, en el árbol del proyecto, seleccionamos CHANNEL CONFIGURATION>CHANNEL 1.

Aquí podemos declarar las propiedades de este canal de comunicación, como dirección IP o máscara de subred.

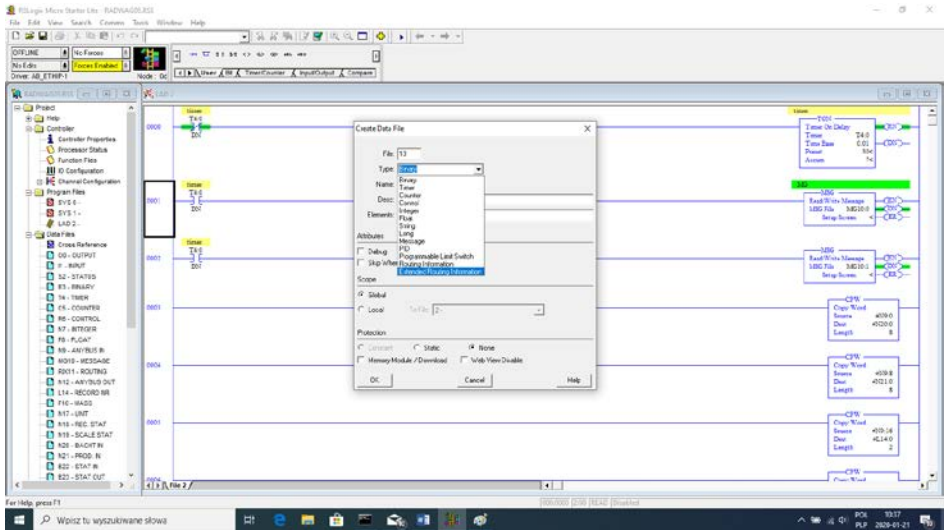


Después de completar la configuración, vale la pena verificar si podemos conectarnos al PLC (en línea) y cargar el proyecto (descargar).

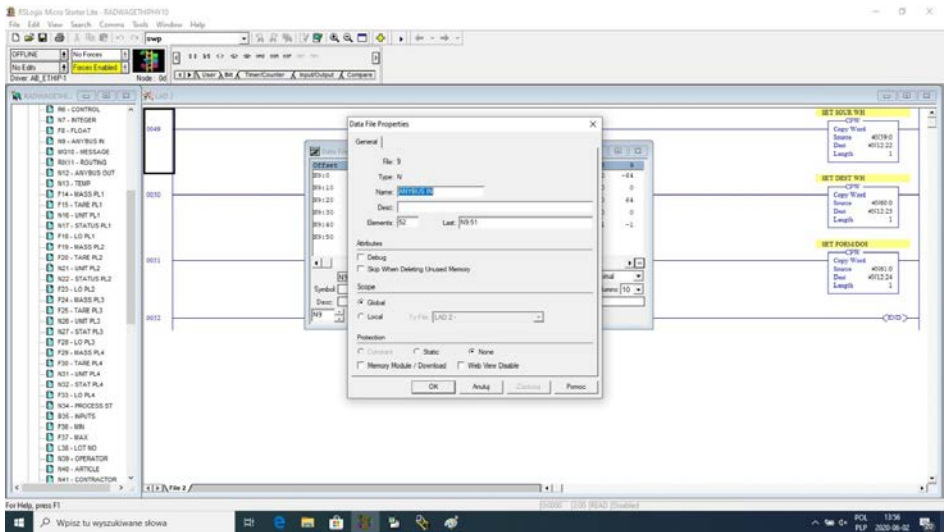
El siguiente paso es agregar un nuevo peldaño a la escalera del proyecto (rung) y agregar la función MSG para leer los datos de la balanza. Antes de agregar una función, se deben agregar nuevos archivos de datos en el árbol del proyecto. Estos serán archivos MG (mensajes) de dos elementos.

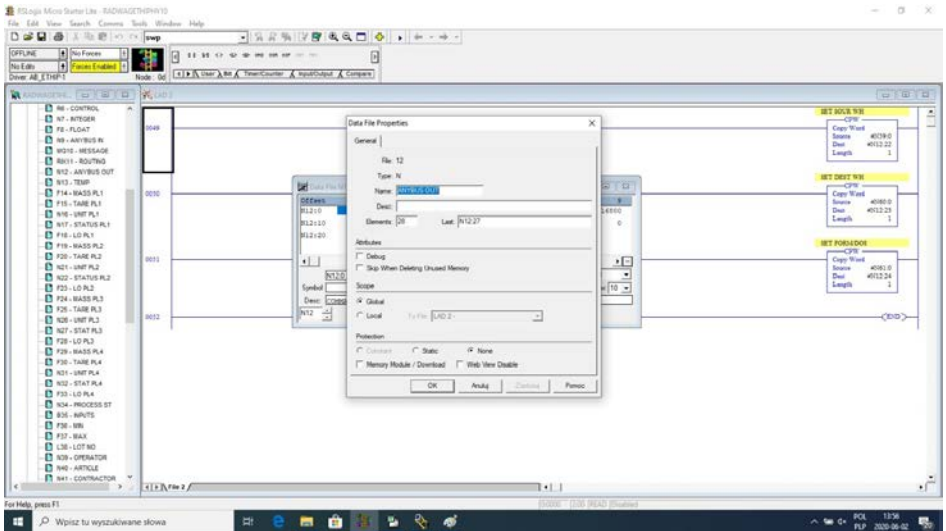


y RIX.

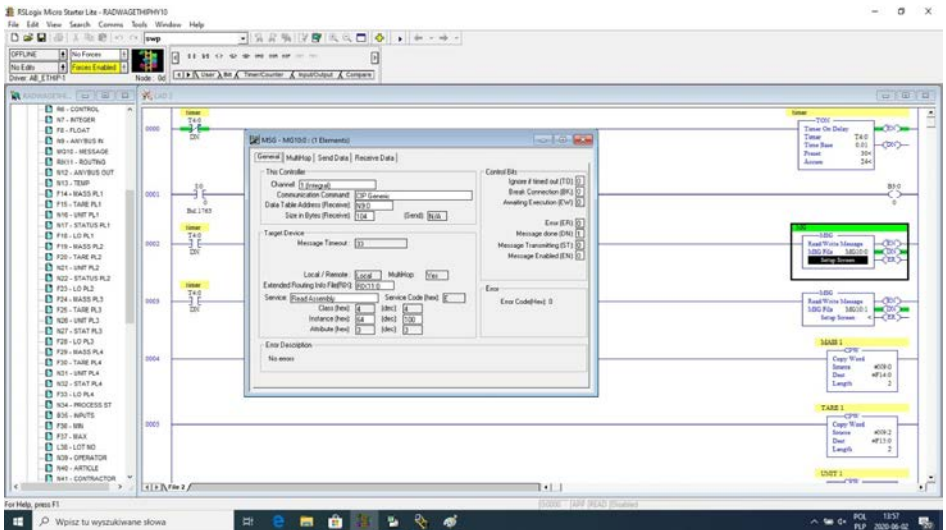


También debe agregar 2 archivos del tipo INTEGER, donde se almacenarán los datos leídos de la balanza y los que se enviarán a la balanza. En el ejemplo, se creó el archivo ANYBUS IN (N9) con un tamaño de 104 bytes y ANYBUS OUT (N12) con un tamaño de 56 bytes..





Ahora podemos agregar funciones MSG, una para leer datos y otra para escribir.

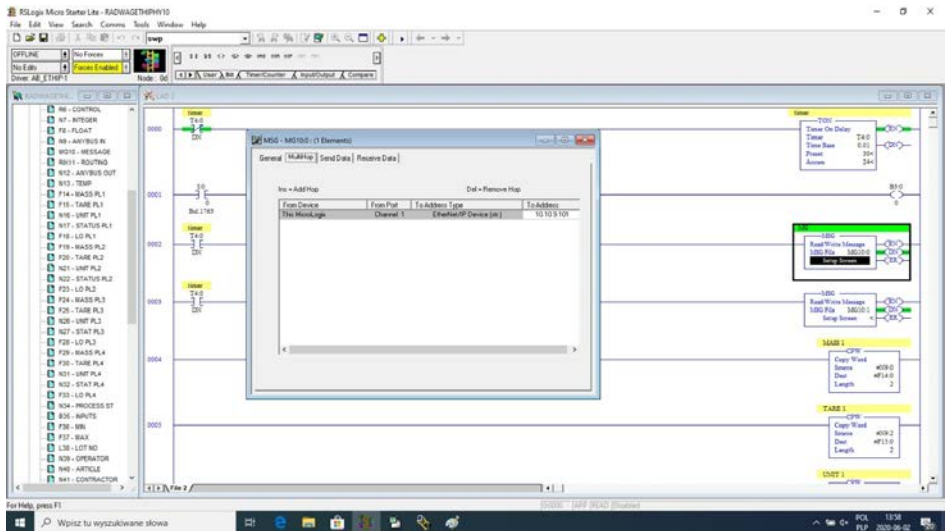


La configuración se reduce a dar:

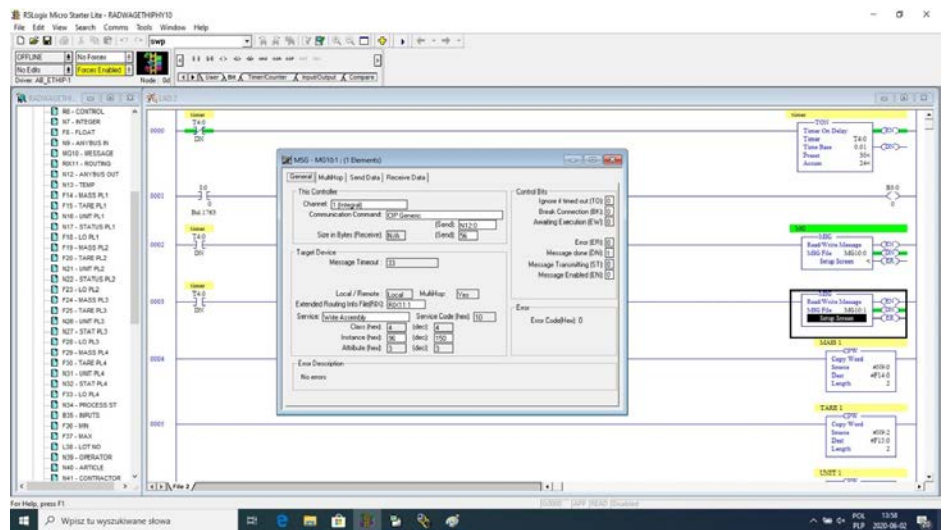
- Channel – elegimos 1 (integral) que corresponde a EtherNet / IP.
- Communication Command – CIP Generic.
- Data Table Address – N9:0 – es nuestro archivo de lectura de datos.
- Size in Bytes – 104 – el tamaño de la tabla de registro de entrada.

- Extended Routing Info File – RIX11:0 – apuntamos al archivo RIX.
- Service: Read assembly.
- Instance : 64.
- MultHop: Yes.

Luego vaya a la pestaña MultiHoop e ingrese la dirección IP de la balanza.



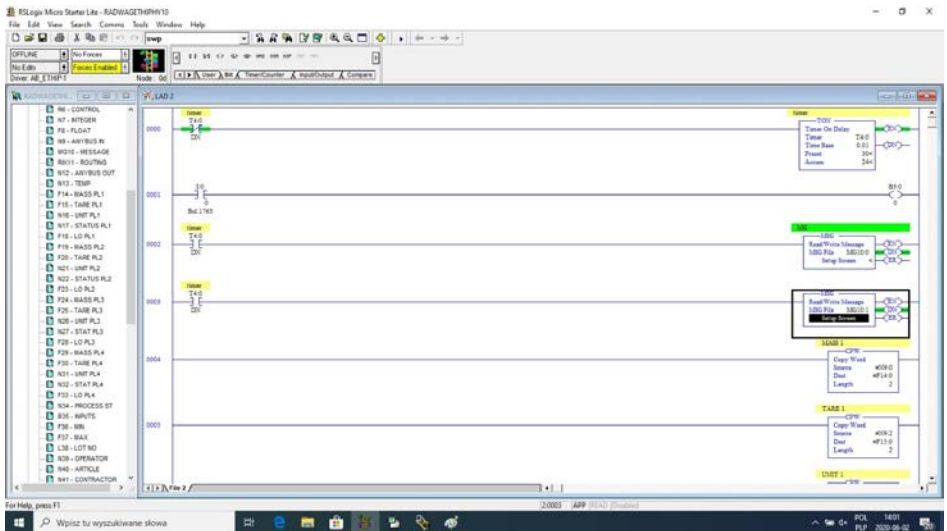
Del mismo modo, creamos funciones para guardar datos en la balanza:



- Channel – elegimos 1 (integral) que corresponde a EtherNet / IP™.
- Communication Command – CIP Generic.
- Data Table Address – N24:0 – es nuestro archivo de almacenamiento de datos.
- Size in Bytes – 56 – el tamaño de la tabla de registro de salida.
- Extended Routing Info File – RIX11:1 – apuntamos al archivo RIX.
- Service: Read assembly.
- Instance : 96.
- MultHop: Yes.

Luego vaya a la pestaña MultiHoop e ingrese la dirección IP de la balanza.

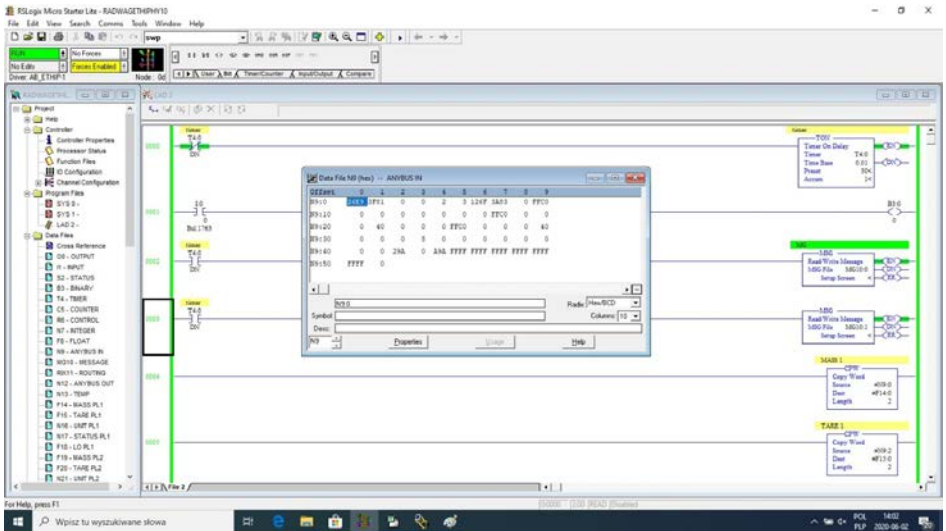
En el ejemplo, las funciones se activan mediante un temporizador, que permite regular la frecuencia de las consultas enviadas a la balanza.



Ahora puede cargar el programa en el controlador y ejecutar el programa.

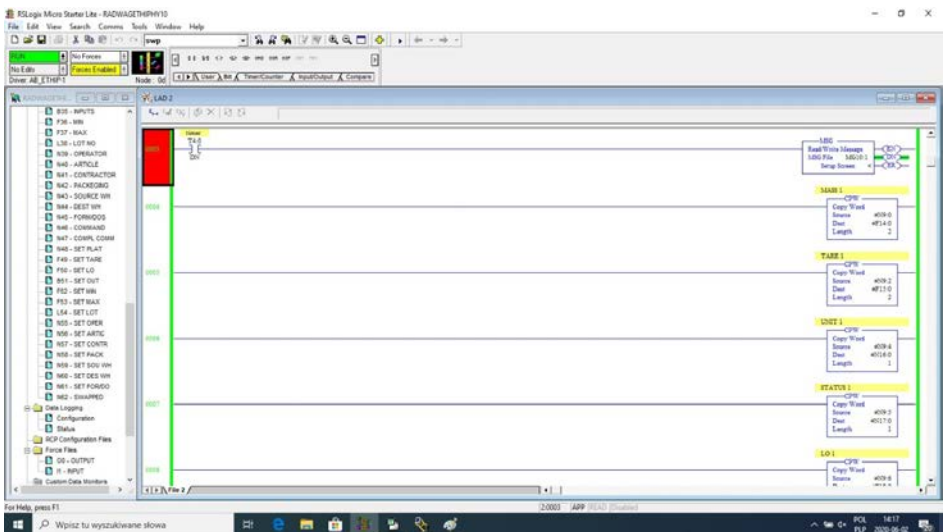
Después de conectarnos al PLC (en línea) en el archivo N9, debemos leer los datos y la función MSG no debe devolver errores.





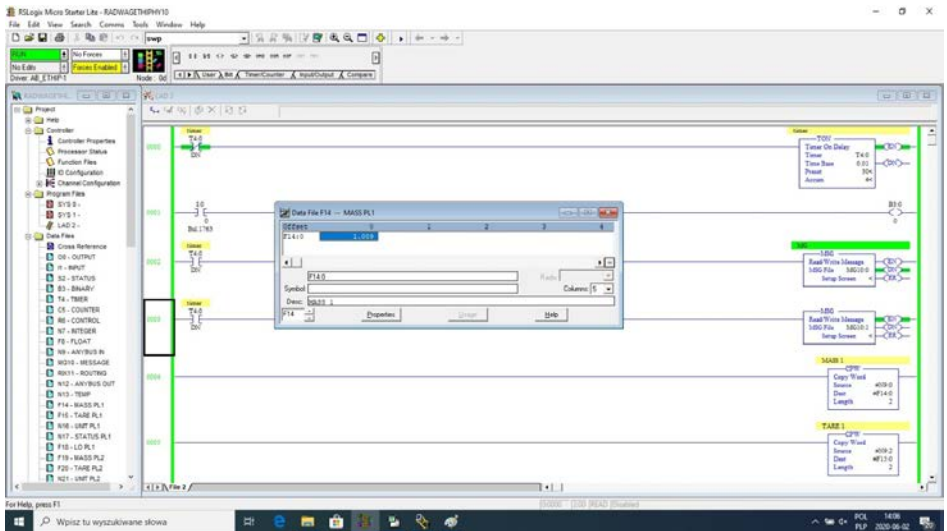
Para mantener las cosas ordenadas, se pueden crear archivos separados para cada variable de balanza.

Los datos entre archivos N9, N24 y archivos variables se reescriben usando la función CPW. Y así, por ejemplo, la función para leer la masa se ve así:



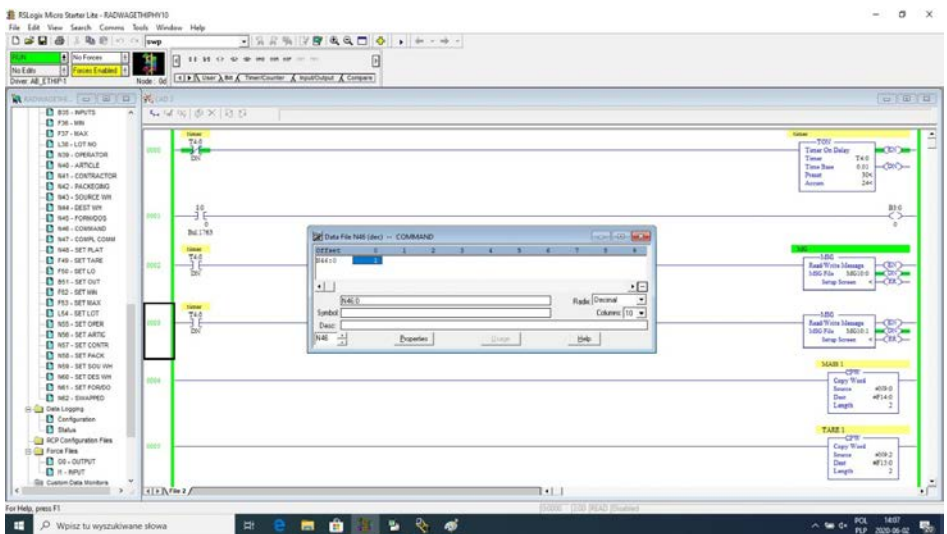
Como resultado, deberíamos leer correctamente los datos del saldo en los archivos correspondientes.

## Ejemplo de lectura masiva:



Al guardar los valores apropiados en archivos correspondientes a los registros de salida, activamos funciones particulares de la balanza.

## Ejemplo de restablecimiento de la balanza:







**RADWAG BALANZAS ELECTRÓNICAS**  
TECNOLOGÍAS DE PESAJE AVANZADAS

