

**UNIVERSIDAD NACIONAL DE LA PATAGONIA
SAN JUAN BOSCO**

**AUTOMATIZACIÓN INDUSTRIAL
TRABAJO FINAL**

Banco de pruebas para motor trifásico

Alumnos

**CAAMIÑA, Daniela
YAPURA, Cristian**

Docentes

**Ing. LORENC, Marcelo
Dr. PEÑA, Ramiro**

MES AÑO



Índice

Índice de figuras

Lista de Acrónimos

CANOpen:

HMI: *Interfaz Humano-máquina*

MBE :

Modbus: *Modicon Bus*

PDO : *Objetos de Datos de Proceso*

PLC : *Controlador Lógico Programable*

SCADA : *Supervisión, Control y Adquisición de Datos*

SDO : *Objetos de Datos de Servicio*

:

:

:

:

:

:

OFS (OPC Factory Server, software de Schneider)

1. Introducción

Actualmente en el Laboratorio de Automatización y Control de la Universidad, se cursan distintas materias en las cuales se necesitan herramientas para realizar diversas prácticas, con el fin de afianzar los conocimientos que se adquieren a lo largo del año.

Para llevar a cabo estas actividades con varias etapas, se requiere demasiado tiempo en realizar pruebas sobre un esquema complejo, es decir con varios elementos, ya que se necesita armar un prototipo de banco de prueba cada vez que sea necesario. Por ejemplo, realizar la conexión de un PLC, variador de frecuencia y un motor puede ser una tarea repetitiva que se busca suprimir.

2. Objetivo

El banco de pruebas cuenta con un punto de apoyo donde se conecta el motor y sus componentes mecánicos, ademas dentro de esta plataforma existe un sistema de medición que posee sensores, variador y PLC para los procedimientos de prueba. Un banco de pruebas puede ser un prototipo de un gran desarrollo industrial o simplemente un banco formado para realizar pruebas educativas.

El objetivo de este trabajo final para la cátedra de Automatización Industrial es construir un banco de pruebas para ser utilizado por cualquier persona dentro el laboratorio de Automatización y Control. Se espera realizar uno que sea capaz de controlar la presión o caudal de agua a través de un sistema ideado y construido por nosotros, que cuente con:

- Motor trifásico 1,5kW (Altium)-*Proporcionado por la cátedra-*
- PLC (Schneider - M340) -*Proporcionado por la cátedra-*
- Variador de velocidad (Schneider - ATV312) -*Proporcionado por la cátedra-*
- Panel de control
 - Botón de emergencia
 - Encendido/ apagado
 - Potenciómetro para variar velocidad
 - Display para observar velocidad
 - Alarmas visuales
- HMI
 - Control general del banco
 - Información en tiempo real
 - Histórico de datos
 - Alarmas

3. Definiciones

Motor eléctrico

Los motores eléctricos son máquinas que transforman la energía eléctrica en movimiento (energía cinética). Estos aparatos se componen, básicamente, del rotor y de un estator donde tiene bobinas inductoras desfasadas entre sí 120°

Variador de velocidad

Es utilizado para controlar la velocidad de giro de un motor. Para regular las revoluciones, se debe tener en cuenta las características del motor, ya que este tiene una curva propia de funcionamiento. Un variador es capaz de generar elementos control de aceleración, frenado, seguridad, control del torque y operaciones que mejoran la eficiencia energética.

PLC

Es una computadora que se utiliza en la ingeniería de automatización para controlar procesos en las industrias.

SoMove

Software que permite configurar variadores de velocidad pertenecientes a la empresa **Schneider Electric**.

Unity Pro

Software común de programación, puesta a punto y explotación de los autómatas Modicon, M340, Premium, Quantum y coprocesadores Atrium de la empresa **Schneider Electric**.

CANopen

CANopen es un protocolo con aplicación industrial de bajo nivel para aplicaciones de automatización. Conecta dispositivos entre sí mediante mensajes entre pares. Basado en el estándar de comunicaciones físicas CAN. Se utiliza en redes de comunicación tipo esclavo, multamaestro.

no me cierra esta definicion, buscar otra

ModBus

ModBus es un protocolo de comunicaciones utilizado para transmitir información a través de redes en serie entre dispositivos electrónicos, basado en la arquitectura maestro/esclavo o cliente/servidor, diseñado en 1979 por Modicon para su gama de PLC. Convertido en un protocolo de comunicaciones estándar en la industria. Además, esta red de comunicación industrial usa los protocolos RS232/RS485/RS422.

HMI - SCADA

Ambas tecnologías, HMI y SCADA, son utilizadas en conjunto en la industria de la automatización. SCADA proporciona funciones de supervisión, alarmas y control, mientras que HMI proporciona las herramientas que necesita para desarrollar imágenes que los operadores pueden usar para monitorear su proceso. El HMI se utiliza para monitorear o visualizar lo ejecutado por SCADA.

iFIX

Software desarrollado por General Electric donde se puede desarrollar aplicaciones sencillas típicas de HMI, o bien, aplicaciones SCADA más complejas como la gestión de elementos y distribución de alarmas.

4. Elementos

4.1. Motor

El motor (Figura 4.1) asincrónico que se utiliza es de la marca **Altium** perteneciente a la firma **Schneider Electric**. Las especificaciones se muestran a continuación

Altium Eff2

- Tipo: TE2A90SP2
- Tensión nominal: 380 V
- Corriente nominal: 3.46 A
- Frecuencia nominal: 50 Hz.
- Potencia: 1.5kW / 2 HP
- Fases: 3
- Factor de Potencia: 0.84



Figura 4.1: Motor Altium

4.2. Variador de velocidad

El variador de velocidad que se utilizó pertenece a la marca **Schneider Electric** (Figura 4.2) y posee las siguientes características.

Altivar 312

- Modelo: ATV312HU15N4
- Tensión: 380-500 V
- Frecuencia: 50/60 Hz
- Potencia: 1.5kW / 2 HP
- Fases: 3



Figura 4.2: Variador Altivar 312

4.3. Módulo Didáctico PLC M340

El Laboratorio de Control de la UNPSJB cuenta con un módulo didáctico que posee un PLC modelo Modicom M340 de la empresa Schneider Electric. Esto cuenta con entradas analógicas, digitales, distintos métodos de comunicación y la capacidad de agregarle otros módulos según las necesidades de los proyectos a desarrollar en un riel.

Los módulos con los que se cuenta son:

- P342030
- DDM16022
- ART0414
- AMI0410

http://instrumentacionycontrol.net/wp-content/uploads/2017/11/IyCnet_Hardware_Modicon_M340-min.pdf

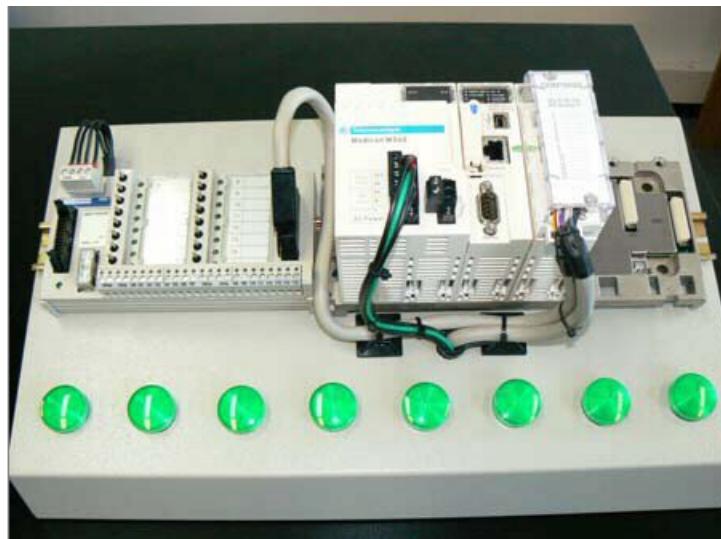


Figura 4.3: Módulo Didáctico PLC M340

4.4. Banco de pruebas

Como se nombró en el objetivo, se busca realizar el control de caudal o presión de un sistema hidráulico. Para esto fue necesario realizar la implementación de un banco de pruebas que cuente de tres partes (Figura 4.4).

- Soporte para el motor y variador de velocidad, diseñado y construido por el profesor Gerardo Arthz. A estos elementos se realizó las correspondientes conexiones, y se agregó elementos adicionales: 3 señales luminosas, llave selectora de dos puntos para seleccionar el modo de comunicación, llave selectora de tres puntos (encendido y sentido del motor) y un pulsador de parada de emergencia (Figura 4.5(a)). Tanto el motor y los elementos adicionales fueron cableados (Figura 4.5(b)) hacia las borneras del variador de velocidad y se tuvo en cuenta para esto las características y funciones del bornero de control proporcionado por el manual del variador de velocidad[1].
- Soporte para una bomba en desuso, de características no conocidas con su bobinado quemado.
- Circuito hidráulico, que incluye un tanque, válvulas y sensores de caudal y presión.



Figura 4.4: Banco de pruebas completo

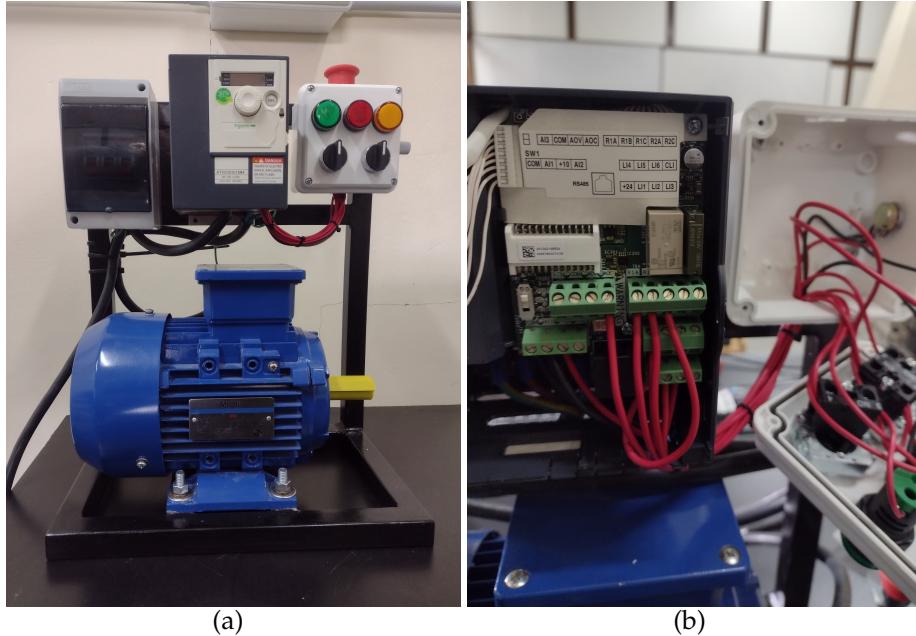


Figura 4.5: Banco de Pruebas

4.4.1. Transmisor de presión

Para este proyecto se utilizan dos transmisores de presión de montaje en línea modelo EJA530E de la familia DPharp de Yokogawa (Figura 4.6.a).

Las características del EJA530E son:

- Precisión: $\pm 0,055\%$ de precisión
- Fiabilidad: $\pm 0,1\%$ Estabilidad por 10 años
- Tiempo de respuesta: 90mseg.
- Lazo de corriente de 4-20mA
- Se puede configurar en la unidad necesaria, en este caso mBar.

4.4.2. Sensor de caudal

Se utilizó un sensor de caudal (Figura 4.6.b) genérico con las siguientes características:

- Rango de caudal: 2- 60L/min
- Máxima presión de agua: 1,75MPa
- Conversión de caudal: aprox 477 pulsos/L $\pm 10\%$



(a) Transmisor de presión
(b) Sensor de caudal

Figura 4.6: Transmisores

4.5. Diagrama

Colocar el diagrama tipo imagen pero ver si se agrega el variador y el plc
FIGURA4.7

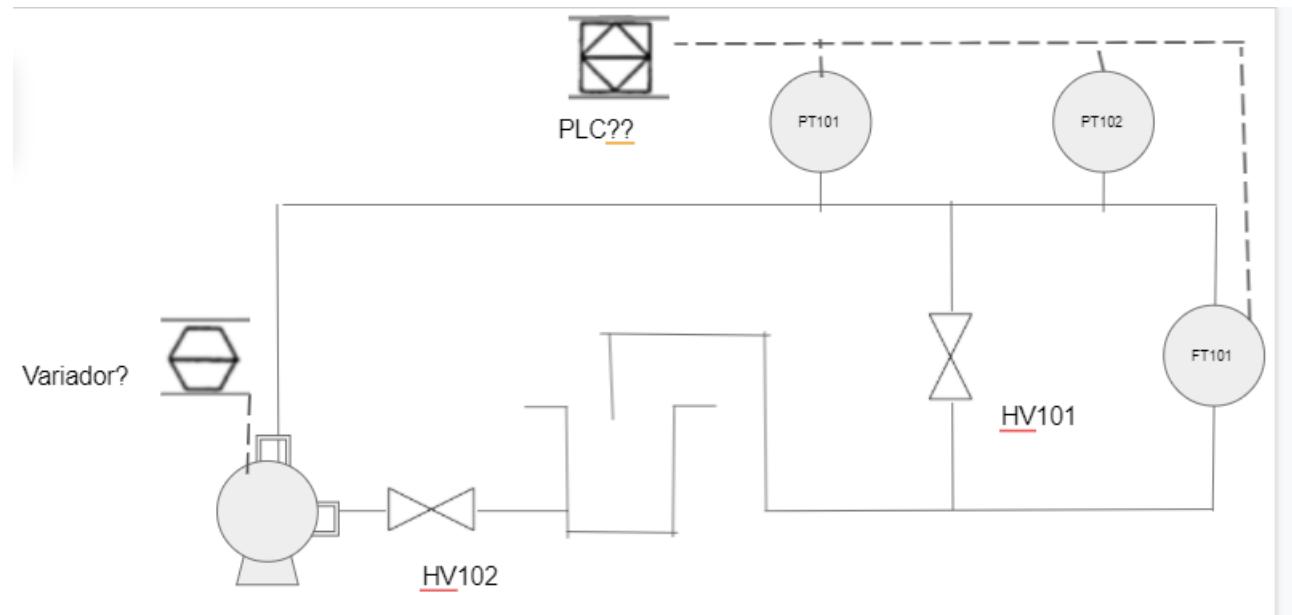


Figura 4.7: Diagrama i&pd

4.6. Presupuesto

falta lo de la bomba https://docs.google.com/spreadsheets/d/1mFoNvgJXUdL2bNnspaBJ_wS5fFA1y1c8fT09Rfod7H0/edit#gid=0

5. Preliminarios

5.1. Programación variador de velocidad

Para realizar la configuración del variador de velocidad con los parámetros del motor se utilizó el software SoMove a través del protocolo ModBus. Se descargó la ultima versión desde la página oficial de Schneider¹ y luego, la librería DTM correspondiente al variador a utilizado².

Una vez realizado esto se procedió a generar un nuevo proyecto donde se eligió las características del variador (Figura 5.1 y 5.2). El próximo paso fue realizar por medio del software la carga de los parámetros del motor (Figura 5.3) y establecer el modo de funcionamiento de las entradas y el protocolo de comunicación.

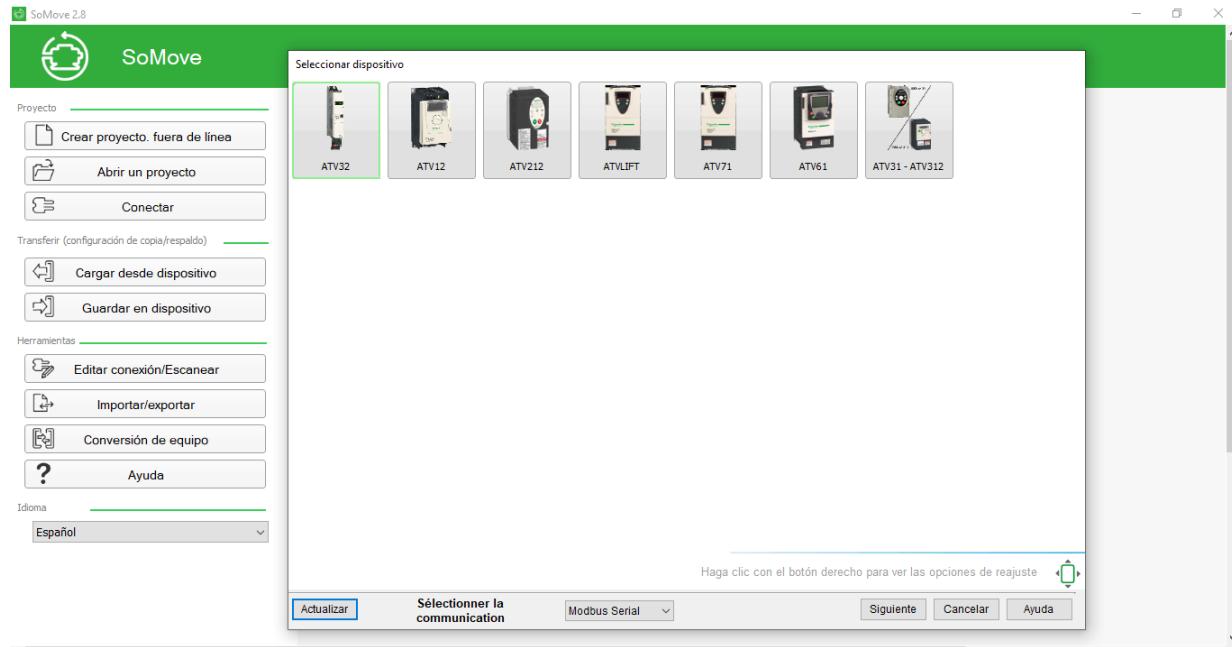


Figura 5.1: Elección de Altivar 312

¹<https://www.se.com/ar/es/product-range-presentation/2714-somove/>

²https://www.se.com/ar/es/download/document/Altivar_DTM_Library/

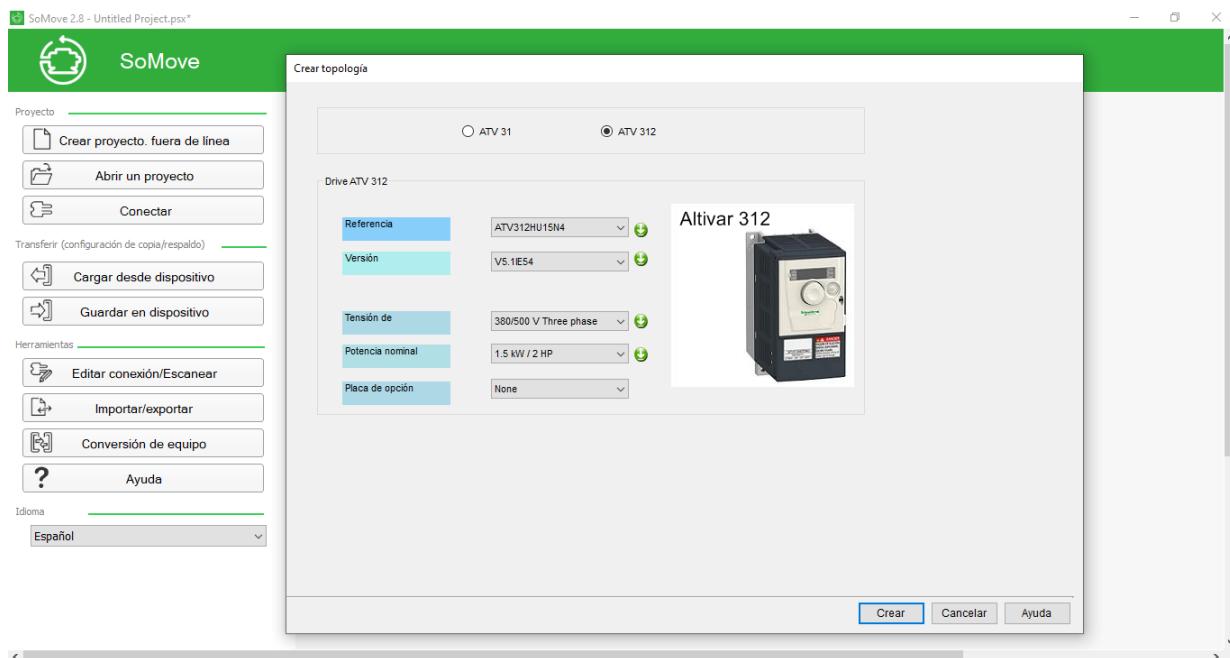


Figura 5.2: Parámetros del variador

Code	Long Label	Current Value	Default Value
UNS	Tensión nominal motor	380 V	400 V
NCR	Intensidad Nom Motor	3.4 A	3.5 A
COS	Motor 1 cos fi	0.84	0.79
ACC	Rampa aceleración (s)	10 s	3 s
DEC	Rampa deceleración (s)	10 s	3 s
NSP	Velocidad nom motor	2840 rpm	1420 rpm
BRL	Aertura de freno (Hz)	2.6 Hz	2139.8 Hz
LAC	Nivel acceso funciones	Nivel 3	Nivel 1
FR2	Canal Referencia 2	Modbus	No
RFC	Asig commut.ref.(1a 2)	L14	Canal1 act.
CHCF	Config. modo control	Separados	No separad.
CCS	Commutacion canal ctrl	L14	Canal1 act.
ROT	Sent marcha autorizado	Ambos	Avance
DO	Salida Analogica	Fallo equipo	No
R1	Asignación del relé R1	Var.marcha	Sin fallo
FST	asignación stop rápida	L15	No
PS2	2 velocidad preselecc.	No	L13
PS4	4 velocidad preselecc.	No	L14
LET	Config. fallo externo	Activo a 0	Activo a 1
EPL	Gestión fallo externo	Paro rampa	Rueda libre

Figura 5.3: Lista de parámetros modificados

Para realizar esta primera configuración se realizó la comunicación de la computadora con el variador a través del protocolo **Modbus** (Figura 5.4) por medio de un cable que en un extremo poseía una ficha RJ45 con un conversor RS485 y en el otro, ficha USB (Figura 5.5).

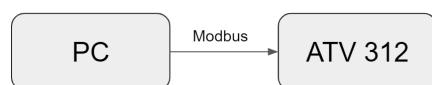


Figura 5.4: Diagrama comunicación PC- Variador



Figura 5.5: Cable de comunicación

5.2. Comunicación variador de velocidad - PLC

Para poder realizar la comunicación entre el variador y el PLC es necesario contar con un cable que realice la conexión desde la salida CANOpen a RS485, para esto se necesitó hacer un cable con las fichas correspondientes en cada extremo según las conexiones que muestran en la Figura 5.6, colocando a su vez resistencia de $120\ \Omega$ en cada punta para evitar ruidos eléctricos y fenómenos de reflexión en la línea.

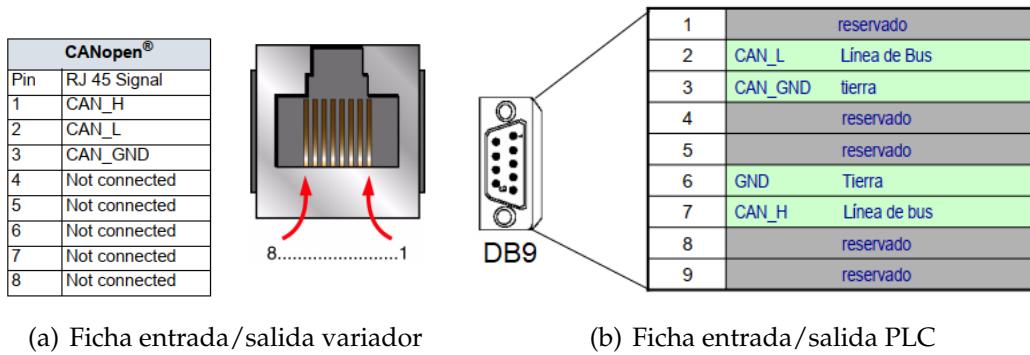


Figura 5.6: Conexión fichas RJ45- DB9

5.3. Lectura de registros

5.3.1. Programación Unity Pro

Para generar la base del proyecto para trabajar, se debe descargar desde la página oficial e instalar el software Unity Pro XL y la librería DTM utilizada en el software soMove correspondiente al variador que se posee. Una vez que esto está instalado se abre un nuevo proyecto y se configura siguiendo los siguientes pasos.

1. Se selecciona el bastidor que se posee.
2. En la configuración gráfica del bastidor (Figura 5.8) podemos introducir los módulos deseados haciendo un clic en la posición seleccionada (Figura 5.7). El laboratorio cuenta con un PLC modular didáctico de la marca **Schneider Electric** de la familia **Modicon** modelo **M340** con los módulos nombrados anteriormente (Sección 4.3).
3. Se debe configurar el módulo Ethernet desde el explorador de Proyectos desplegamos la carpeta Comunicación y se realiza clic con el botón derecho sobre Redes y luego en Nueva Red, Ethernet (Figura 5.10).
4. Se creó una nueva sección para lenguaje FDB para ver parámetros básicos.
 - Los Diagramas de Bloques de Función consisten en un Editor gráfico orientado al dibujo de bloques. El lenguaje consiste en los Bloques de Funciones reusables elementales y derivados.

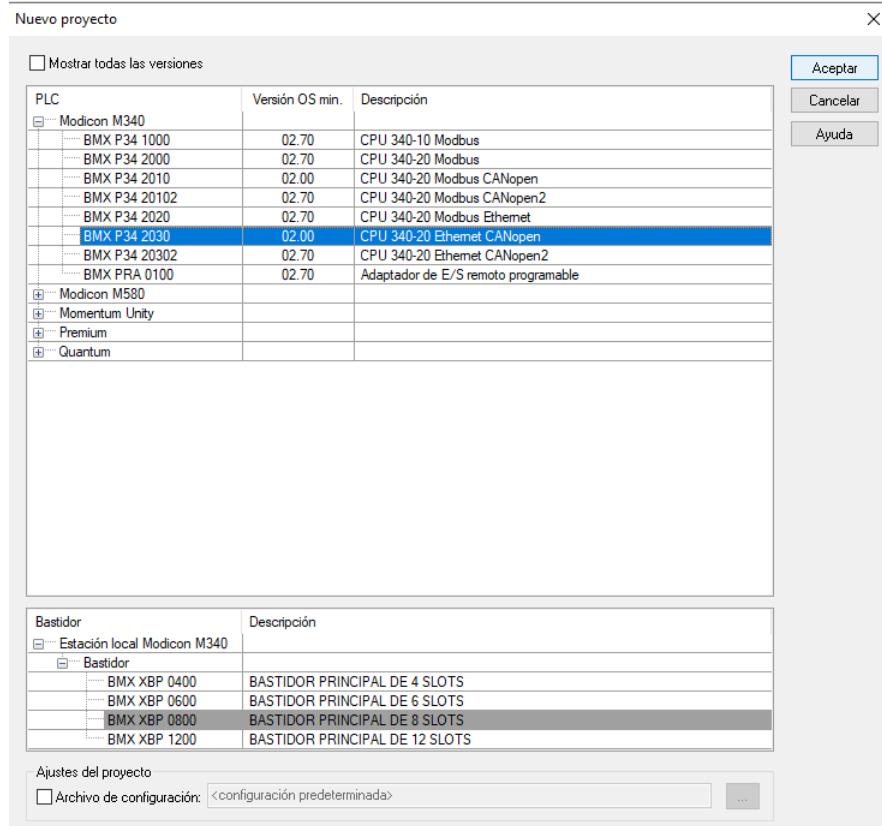


Figura 5.7: Elección del bastidor

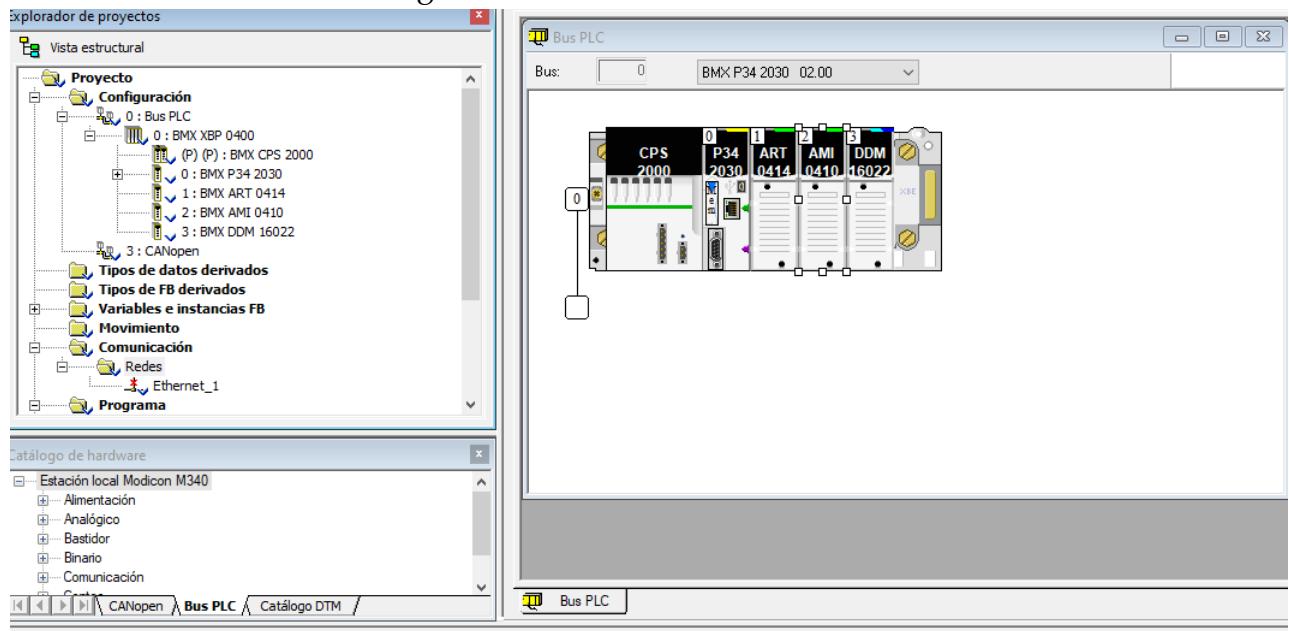


Figura 5.8: Módulos PLC

Una vez que se completó la configuración de la comunicación variador - PLC se procedió a crear una *Pantalla de operador* dentro del mismo programa (Figura 5.11) la cual fue utilizada para interactuar y observar diversos parámetros, como modificar velocidades, observar señales luminosas y ver distintos valores proporcionados por el variador de velocidad.

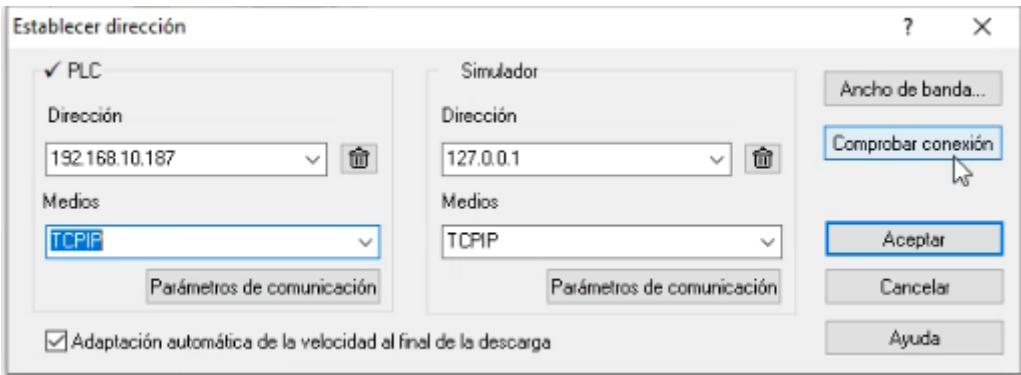


Figura 5.9: Dirección IP

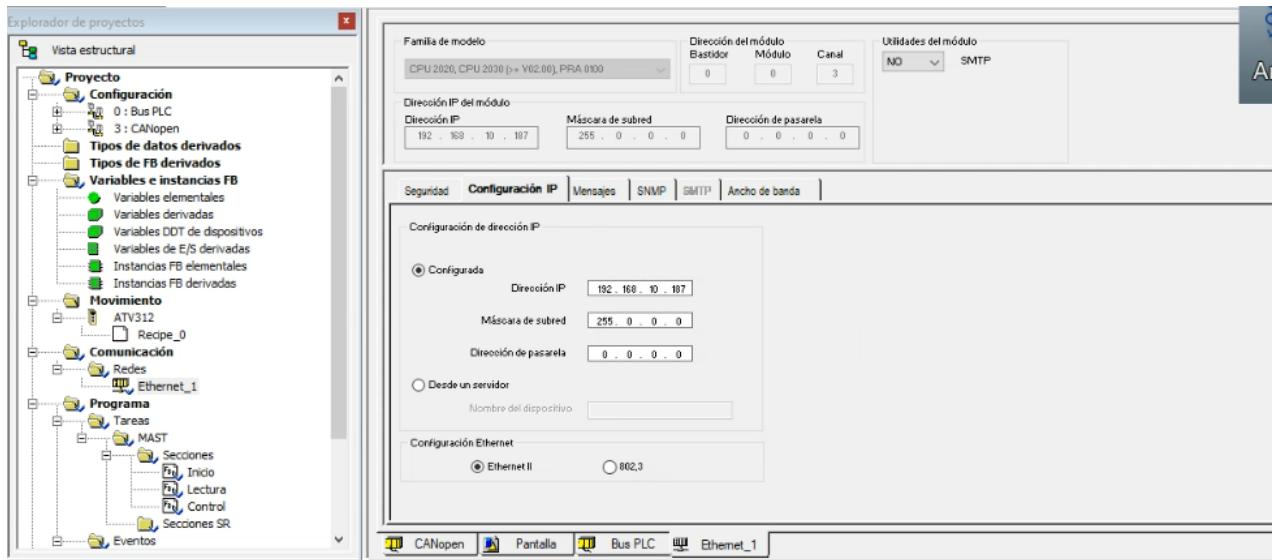


Figura 5.10: Dirección módulo Ethernet

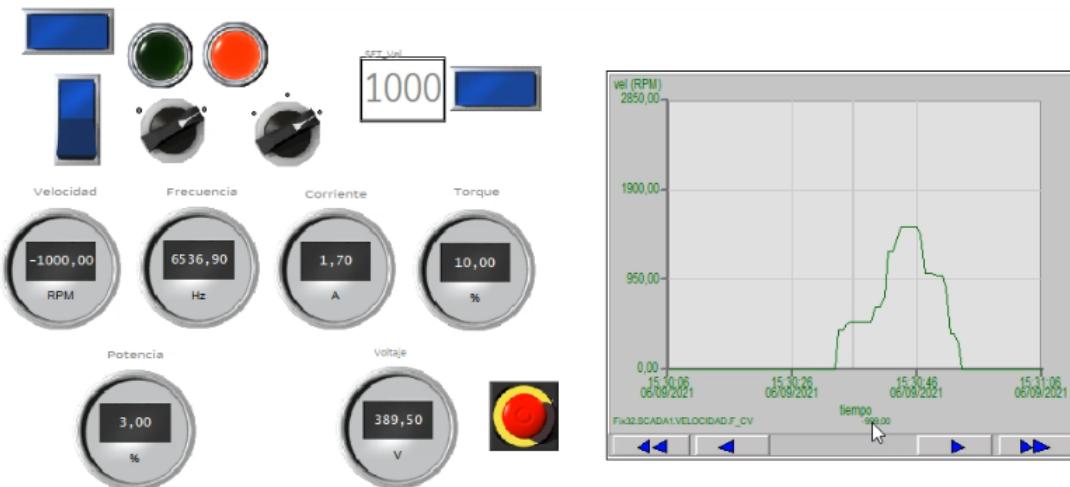


Figura 5.11: HMI simple

Para realizar la programación del HMI se utilizó los bloques de funciones de movimiento (Motion Function Block, MFB) del software UnityPro (Figura 5.12). Estos bloques necesitan de un bloque maestro “CAN_HANDLER” el cual permite comprobar la comunicación CANopen, así como la coherencia entre las configuraciones de software y física.

Otros de los bloques más utilizados dentro del programa fueron “MC_READPARAMETER” que se utiliza para leer, mediante mensajes Service

Data Object (SDO), una variable del variador definida en una dirección CANOpen dada por el fabricante [2].

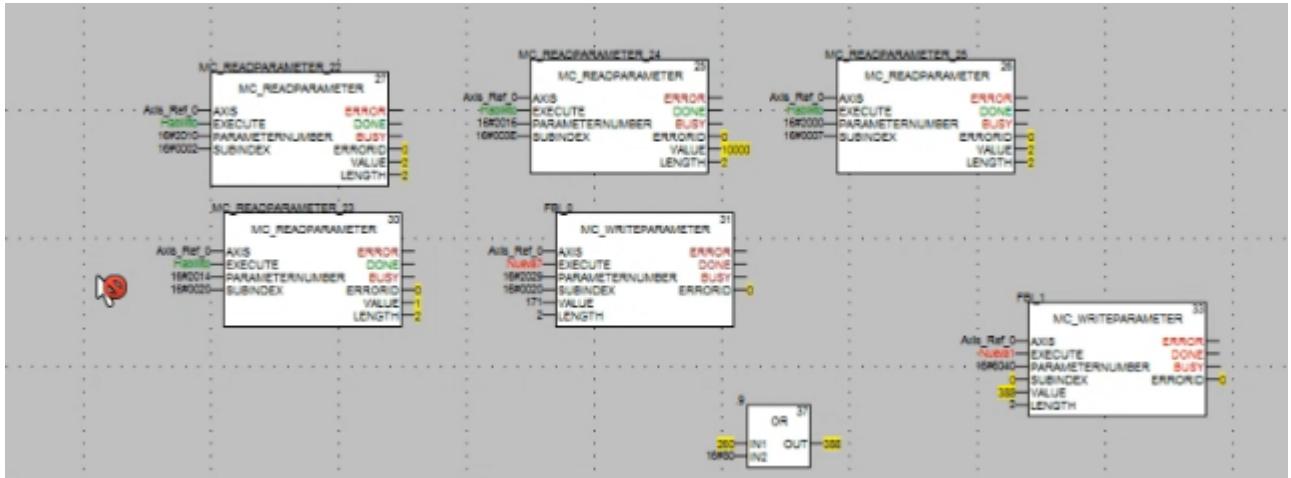


Figura 5.12: Programa con bloques MFB

5.4. Medición de caudal

Según la cantidad de flujo que pase por el caudalímetro, este entregaría pulsos que debían ser leídos con un módulo externo del PLC por la baja resolución que posee. Para esto se planteó utilizar un ESP8266 capaz de obtener los pulsos, colocarlos en un registro y enviarlos por un servidor Modbus TCP/IP cada 20ms al PLC. En la sección de programación del PLC se realizó las cuentas correspondientes para realizar la conversión de pulsos a caudal (Figura 5.13). Tanto para el módulo, como para el caudalímetro se usó una fuente externa de 3,3 V de alimentación.



Figura 5.13: Diagrama de flujo del caudalímetro

6. Desarrollo

A partir de lo programado en Unity PRO utilizado para obtener los primeros registros, se realizó modificaciones necesarias donde se eliminó y agregó variables de la lista de direcciones que se creyeron necesarias a la hora de implementar el proyecto.

6.1. Adquisición de datos

En el objetivo se propuso que el sistema sea capaz de controlar presión o caudal, para lograr lo estipulado. Como en cualquier sistema de control, es necesario conocer las plantas con las que se trabajará, reconociendo en el banco de pruebas tres plantas observadas en FIGURA 4.7 y que serán llamadas:

- PIT001: Presión 1 medida a la salida de la bomba.
- PIT002: Presión 2 medida luego de la columna que deiva el fluido.
- FT001: Caudal que pasa por presión 2.

Para realizar las estimaciones de las plantas del sistema se utilizó el protocolo OPC en conjunto con Matlab. Por medio de OFS, se procedió a crear y configurar un servidor con la dirección correspondiente y se seleccionó el programa realizado en UnityPro donde se encontraban las variables necesarias (Figura 6.1).

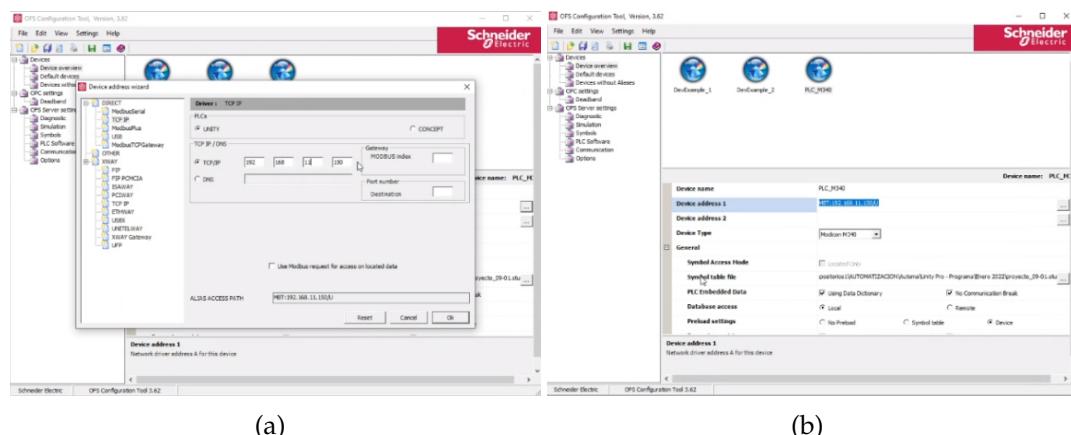


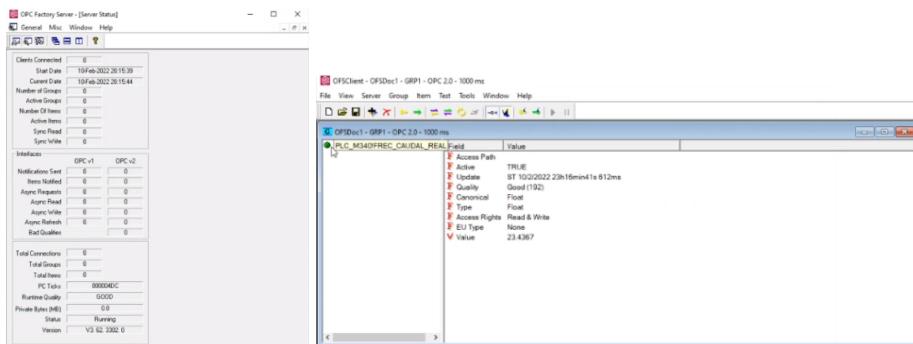
Figura 6.1: Configuración OPC

Una vez configurado el servidor se abre el programa **OPC Factory Server** dando inicio al servidor (Figura 6.2a). Para observar si la comunicación esta establecida de forma correcta, se utilizó el programa **OFS Client** dónde se debió agregar el tag correspondiente a la variable a observar (Figura 6.2b)

Una vez corroborada la comunicación con el servidor OPC, se procedió a crear un cliente OPC en Simulink (perteneciente a Matlab) para adquirir y guardar las variables necesarias.

6.1.1. Uso de Matlab

En el entorno Simulink se procedió a configurar un bloque de cliente OPC con la dirección IP donde se encuentra el servidor previamente creado. Luego, para leer las variables necesarias se creó un bloque de lectura OPC (Figura 6.3 a) y con un bloque *Scope*, se activó la opción para que se guarden los vectores de las variables a estudiar (Figura 6.3 b).



(a) OPC Factory Server

(b) OFS Client

Figura 6.2: Conexión servidor OPC

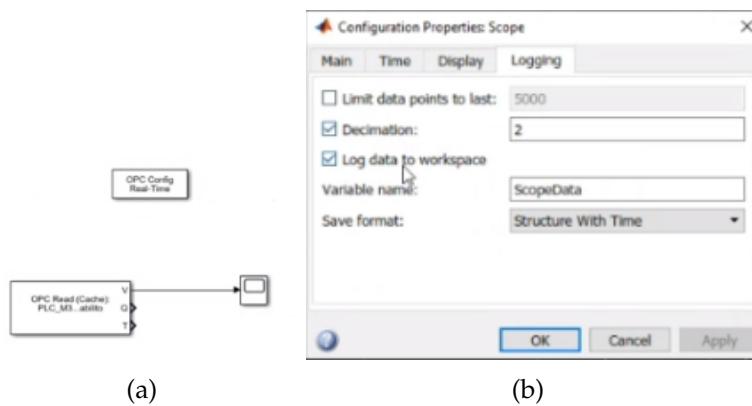


Figura 6.3: Cliente OPC en Simulink

6.1.2. Estimación de la planta

Para realizar la estimación de las plantas se utilizó el Método de Strejc con retardo (Figura 6.4)[3]. Este método se emplea para la identificación de sistemas de polos múltiples, mediante los parámetros T_u y T_a obtenidos sobre la respuesta del sistema. Tras obtener el valor de las variables T_u y T_a , se obtiene el valor de T_u/T_a este valor determinará el número de polos múltiples “n”.

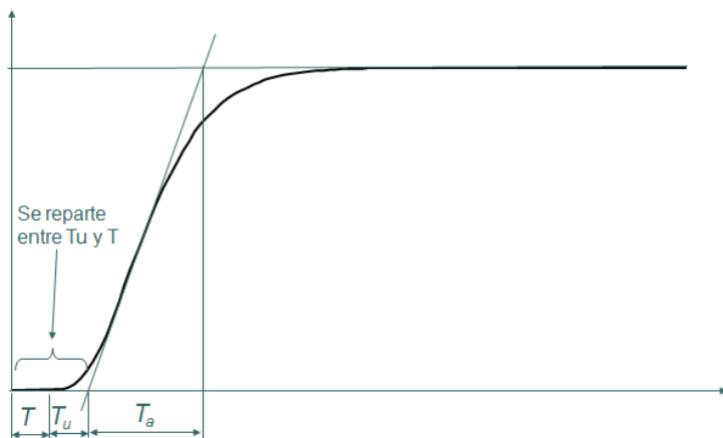


Figura 6.4: Parámetros de Strejc con retardo

La función de transferencia general para un sistema de polos múltiples es:

$$G(s) = \frac{K}{(1 + \tau \cdot s)^n} \cdot e^{-T_s} \quad (1)$$

Para obtener el orden de la función de transferencia se realizó un script en Matlab que dió como resultado las funciones de transferencia nombradas a continuación:

- Planta de presión PIT001:

$$G(s) = \frac{K}{(1 + \tau \cdot s)^n} \cdot e^{-T \cdot s} \quad (2)$$

- Planta de presión PIT002:

$$G(s) = \frac{K}{(1 + \tau \cdot s)^n} \cdot e^{-T \cdot s} \quad (3)$$

- Planta de caudal FT001:

$$G(s) = \frac{K}{(1 + \tau \cdot s)^n} \cdot e^{-T \cdot s} \quad (4)$$

6.1.2.1. Comparación numérica- real

En las siguientes imágenes se observa la gráfica de cada planta estimada comparada con los datos obtenidos en las mediciones. imágenes con fondo blanco

6.1.3. Cálculo del controlador PID

El controlador PID de cada planta se calculó con *Tune PID controllers*, dónde se buscó que las respuestas sean críticamente amortiguado para....

Los valores obtenidos de la aplicación perteneciente a Matlab se muestran en la tabla 6.1, los cuales se ingresaron en los bloques de *UnityPro* con las respectivas modificaciones numéricas según lo establecido por el software.

	PIT001	PIT002	FT001
K_p	???	???	???
K_i	???	???	???
K_d	???	???	???
N	???	???	???

Tabla 6.1: Valores de PID

6.1.4. Pruebas de control

Para observar la respuesta del sistema de control se generaron perturbaciones con las válvulas del banco de prueba. FIGURA 6.5

6.2. HMI

Para realizar la pantalla de supervisión, control y adquisición de datos operador (SCADA) se utilizó el software iFix perteneciente al grupo **General Electric**.

El sistema HMI creado (Figura 6.6) se dividió en las siguientes secciones:

- Esquemático del circuito hidráulico físico con las variables de presión y caudal en tiempo real.
- Valores de funcionamiento del motor obtenidos por el variador de velocidad.
- Alarma, dónde se observa de forma visual valores críticos alcanzados en el sistema.

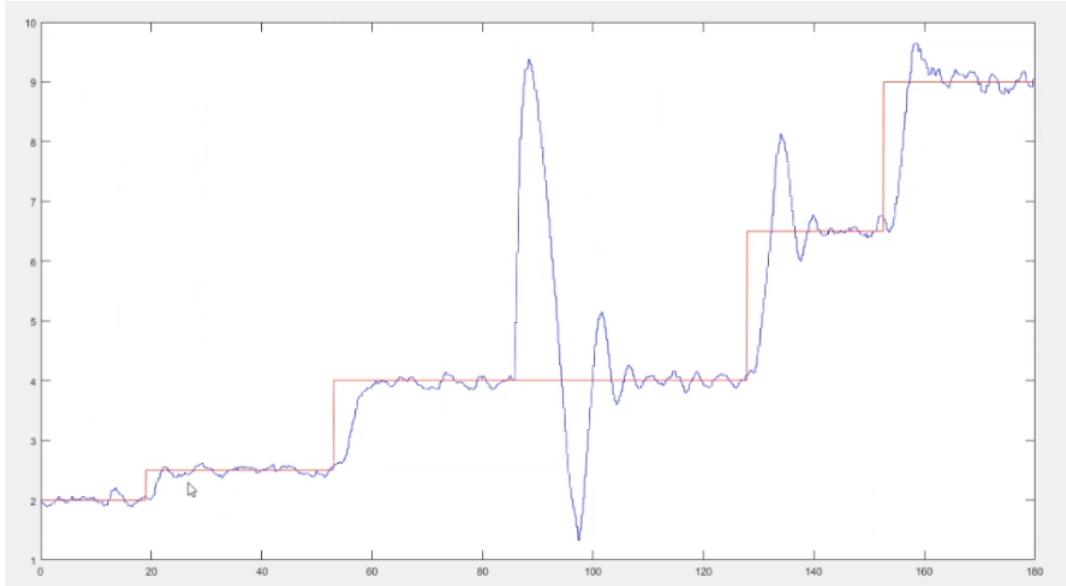


Figura 6.5: Comportamiento del sistema ante perturbaciones

- Indicador de modo de funcionamiento físico o remoto.
- Modo de control a lazo abierto o lazo cerrado.

Para el modo de lazo cerrado se creó una ventana individual para cada sistema de presión y caudal.

- Pantalla para observar gráficos en tiempo real donde se divide según la variable a observar, con botones para abrir el control PID del sistema.
- Pantalla donde se observa datos históricos y se puede generar un archivo .txt con la información de la variable elegida en un determinado período de tiempo.

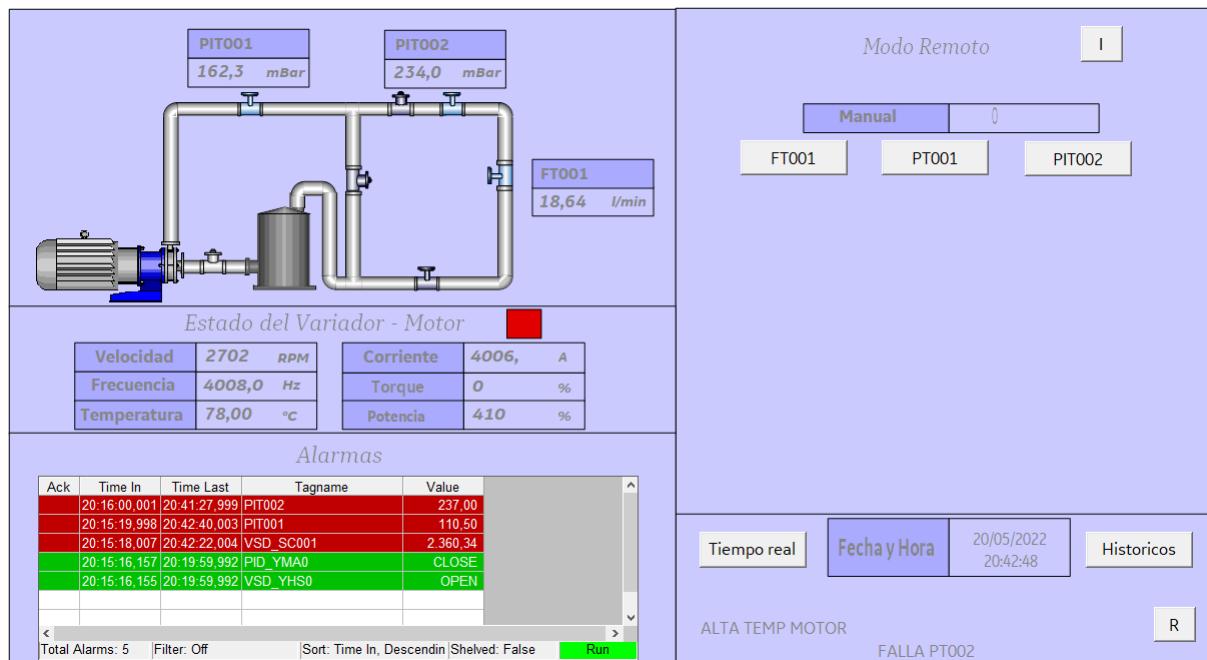


Figura 6.6: Pantalla SCADA

6.2.1. Comunicación SCADA- Computadora

Para realizar la configuración de cada ícono de la pantalla SCADA con su respectiva variable, se debió crear un MBE Driver (Figura 6.7) dónde se estipula la dirección IP

y el mapa de memoria con sus respectivas secciones que luego serán utilizadas por el DataBase (Figura 6.8).

Una vez creado el MBE Driver se debe generar la tabla *DataBase* en dónde estará el nombre, dirección IP, tipo de elemento, descripción, alarma asociada, entre otros puntos de cada elemento.

La lista se encontrará unificada con las direcciones de *UnityPro* (Tabla...).

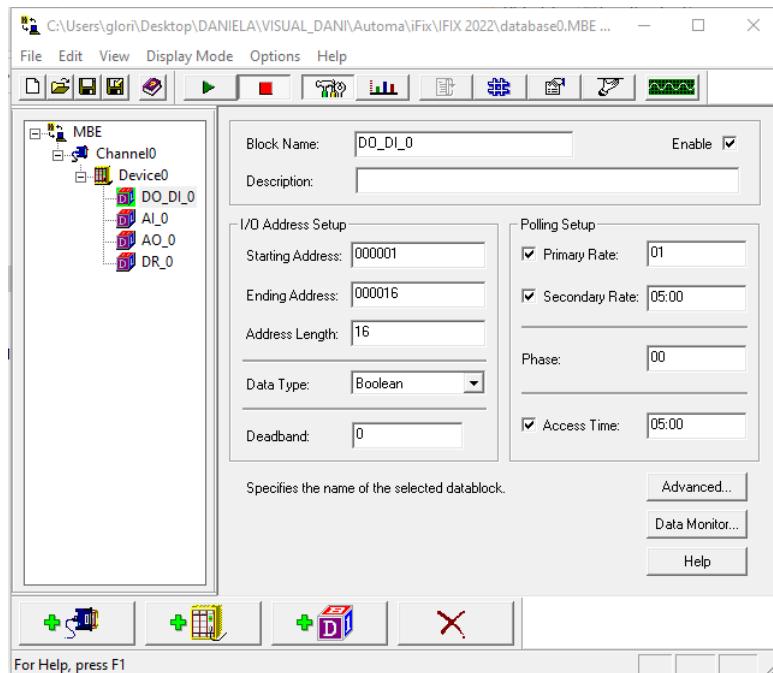


Figura 6.7: Configuración MBE

	Tag Name	Type	Description	Scan	I/O Dev	I/O Addr
1	FT001	AI	Caudal	1	MBE	Device0:400001
2	PID0FT1_KD	AI	F1 KD	1	MBE	Device0:400081
3	PID0FT1_KI	AI	F1 KI	1	MBE	Device0:400077
4	PID0FT1_KP	AI	F1 KP	1	MBE	Device0:400079
5	PID0FT1_SP	AI	Valor Ref F1	1	MBE	Device0:400061
6	PID0FT1_TD_LAG	AI	F1 TD LAG	1	MBE	Device0:400051
7	PID0PIT1_KD	AI	P1 KD	1	MBE	Device0:400067
8	PID0PIT1_KI	AI	P1 KI	1	MBE	Device0:400063
9	PID0PIT1_KP	AI	P1 KP	1	MBE	Device0:400065
10	PID0PIT1_SP	AI	Valor Ref P1	1	MBE	Device0:400057
11	PID0PIT1_TD_LAG	AI	P1 TD LAG	1	MBE	Device0:400047
12	PID0PIT2_KD	AI	P2 KD	1	MBE	Device0:400075
13	PID0PIT2_KI	AI	P2 KI	1	MBE	Device0:400071
14	PID0PIT2_KP	AI	P2 KP	1	MBE	Device0:400073
15	PID0PIT2_SP	AI	Valor Ref P2	1	MBE	Device0:400059
16	PID0PIT2_TD_LAG	AI	P2 TD LAG	1	MBE	Device0:400049
17	PID_SEL	AI	Seleccionar control P1 P2 F1	1	MBE	Device0:400083

Figura 6.8: Configuración MBE

6.2.1.1. Pruebas mediante ModSim

Para realizar pruebas intermedias antes de unir SCADA con el programado para el PLC se utilizó el software ModSim, dónde se generó los distintos mapas de memoria utilizados y allí se podían modificar variables para observar el correcto funcionamiento de distintos elementos en el HMI.

6.2.2. Alarmas

Dentro de la pantalla principal es posible observar el alarmero. Estas alarmas están compuestas por las variables de la siguiente tabla con sus respectivos valores y

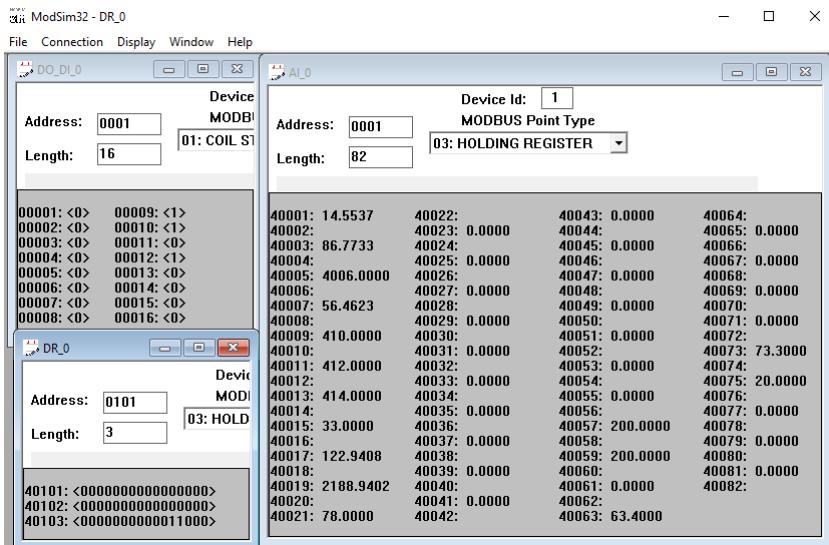


Figura 6.9: ModSim

prioridades.

Poner lista que esta en drive

6.2.3. iHistorian



Figura 6.10: Pantalla SCADA

7. Conclusiones

8. Bibliografía

Referencias

- [1] Schneider Electric. "Manual de instalación". En: *Altivar312- Variadores de velocidad para motores asincronos* (2013).
- [2] Schneider Electric. "Communication variables manual". En: *Altivar312- Variadores de velocidad para motores asincronos* (2009).
- [3] Jorge Pomares y Ángel Martínez Bueno. "Sistemas de Control Automático. Identificación experimental de sistemas". En: *Sistemas de Control Automático* (2011).

Versión de programas utilizados

- *Unity Pro XL* (V11.0) Schneider Electric Industries SAS (2015)
- *SoMove* (V2.8.4.0) Schneider Electric Industries SAS (2020)
 Altivar DTM Library ATV31/312 (V2.0.2.0)
- *iFix Workspace* (V6.0) General Electric Company (2018)
- *Matlab* (V2018a)
 Simulink:
 Versión:
- *Historian*:
 Versión: 2018a
-
-

9. Anexos

10. Anexo 2: Manual SCADA

10.1. Características generales

10.2. Pantallas

10.2.1. Pantalla principal

10.2.1.1. Alarmas

10.2.2. Pantallas de control

10.2.3. Pantallas gráficas

10.2.3.1. Gráficos en tiempo real

10.2.3.2. Gráficos históricos