

Universidad Nacional de Tucumán
Facultad de Ciencias Exactas y Tecnología



Proyecto: Sistema de control de nivel de agua

Asignatura: Electrónica Industrial

Año: 2024

Integrantes:

*Quagliotti, Matias Pablo
Ordoñez Nadra, Facundo
Vale, Fernando Alexis*



INDICE

Introducción y objetivos	3
Descripción del PLC	6
Descripción general del sistema	8
Actuadores	9
Variador de velocidad: ALTIVAR 11	9
Bomba	10
Sensores	12
Sensor de presión del Tanque: MPX5010	12
Sensor de presión para medir caudal: MPX 5700	14
Sensor capacitivo: STF 2500 B	16
Caudalímetro: Sensor YF-S201	17
Acondicionamiento de señal	18
Configuración y programación del PLC	20
Variables utilizadas	20
Configuraciones previas	22
Programación	25
Programa principal	25
Caudalímetro	32
Promedio Presión	33
Instanciación de los programas	34
Configuración del PID	35
Conexión entre CPU y PLC	37
Interfaz SCADA	39
Ventanas	39
Comandos	39
PID Controller	40
Lecturas	41
Alarmas	42
Registro de Eventos y Usuarios	43
Usuarios	45
Ensayos realizados	46
Observaciones y Complicaciones	48
Conclusiones	49

Introducción y objetivos

En este proyecto pondremos en práctica nuestros conocimientos en sistemas de control automáticos para manipular el nivel de agua de un tanque, que se llena a partir de una cisterna mediante una bomba. La velocidad de flujo de esta bomba será nuestra variable de control.

El sistema a manipular consiste de las siguientes partes:

- Un bidón superior que cumple el papel de “Tanque” (Fig.1).
- Un bidón inferior que cumple el papel de “Cisterna” (Fig.1).
- Dos tubos de interconexión: Uno externo que permite el bombeo de la cisterna al tanque y otro interno que simula el consumo del tanque (Fig.1).
- Una bomba que transportará el fluido desde la cisterna hacia el tanque (Fig.3).
- Una llave para limitar el caudal en este primer tubo (Fig.1).
- Una llave en el segundo tubo, que simulará el consumo de agua (Fig.2).
- Un variador de velocidad de la marca ALTIVAR, que manipulará la velocidad del motor de la bomba según indique el usuario (Fig.1).
- Sensores: Dos de presión, uno de caudal y otro capacitivo (Figs. 1, 3 y 4).
- Circuitos acondicionadores de señal (Fig.1).

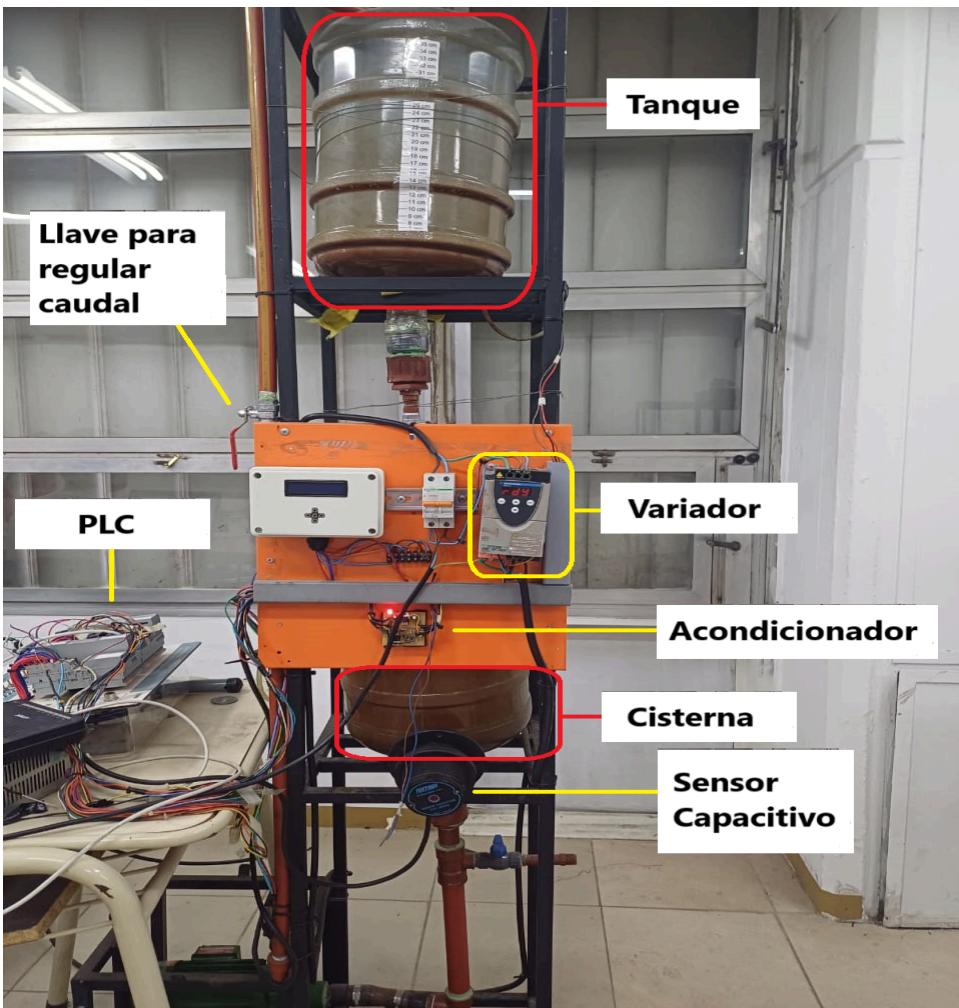


Figura 1: Sistema completo



Figura 2: Llave de consumo



Figura 3: Bomba de agua



Figura 4: Sensores de presión



Figura 5: Sensor de caudal

Descripción del PLC

Para realizar el sistema de control contamos con un controlador lógico programable (o PLC) de la marca Phoenix Contact, modelo AXC F 2152. Dentro del dispositivo encontramos varios módulos (Fig.6).

- Módulo A1: Unidad de procesamiento del PLC. Aquí se encuentran el procesador y la memoria. Su función es almacenar y ejecutar el programa así como las variables. Posee además dos entradas Ethernet cuya función se explicará más adelante.
- Módulo A2: Módulo de entradas y salidas digitales. Posee 8 entradas y 8 salidas digitales que por defecto están conectadas a 8 interruptores y 8 luces led respectivamente.
- Módulo A3: Módulo de entradas y salidas analógicas. Posee dos entradas analógicas de tensión y dos de corriente, pudiendo utilizar hasta dos al mismo tiempo. Así mismo, posee dos salidas analógicas de tensión y dos de corriente, pudiendo utilizar hasta dos a la vez. Este módulo trae por defecto un potenciómetro, un vúmetro y varias entradas que no se van a utilizar en este proyecto.

Los tres módulos anteriores están interconectados mediante el protocolo Axioline. La conexión física se da a través de una barra metálica ubicada en la parte inferior de los módulos.

- Módulo A5: Módulo de entradas y salidas digitales. Mismo funcionamiento que el módulo A2. La diferencia radica en que su protocolo de comunicación no es Axioline, sino Profinet, la cual se da físicamente mediante otro módulo.
- Módulo A4: Módulo de comunicación Profinet. Su función es conectar el módulo A5 con la unidad central de procesamiento del PLC (Módulo A1), ya que éstos módulos no cuentan con conexión Axioline. La conexión profinet utiliza una de las entradas Ethernet del módulo A1.

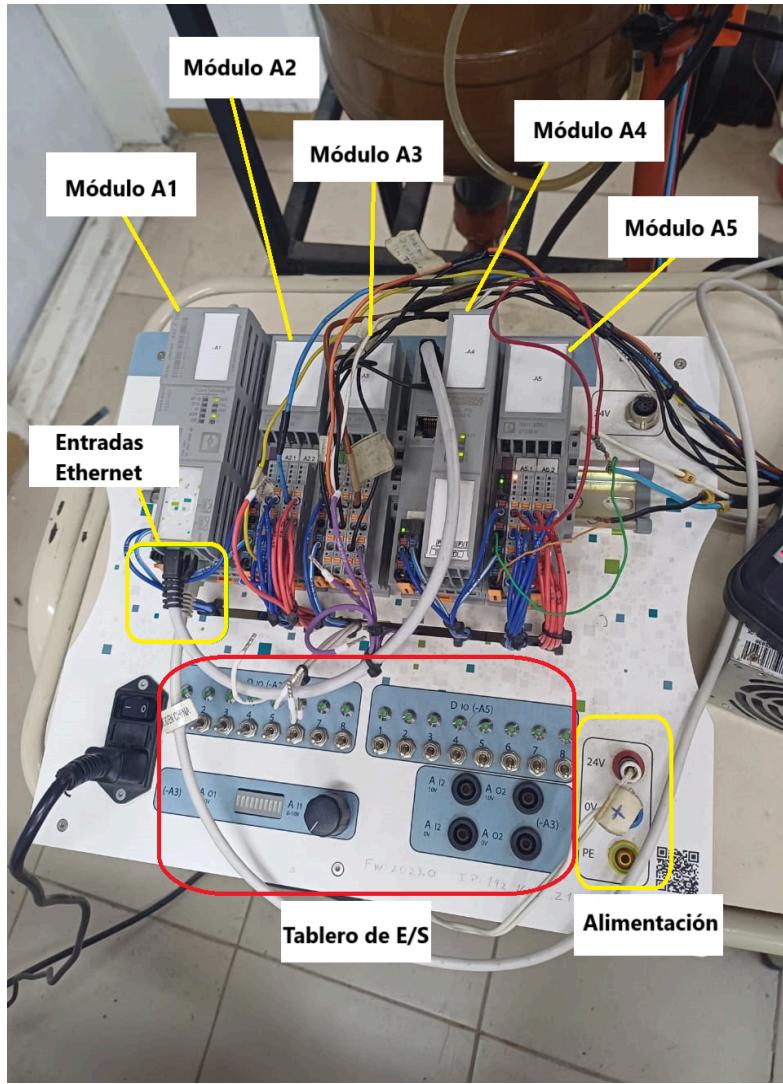


Figura 6: PLC y módulos

El control del nivel del tanque se debe poder realizar de dos maneras: Un modo local, que permita manipular directamente las variables desde el “PLCNext Engineer” (El software que utilizaremos para la programación del PLC), y también físicamente desde el propio PLC. Por otro lado, el control debe poder realizarse también desde un modo remoto, a través de una interfaz SCADA desarrollada en el programa “Visu+”. La conexión entre el programa, el PLC y la interfaz SCADA se detallarán posteriormente.

Descripción general del sistema

Para dar comienzo al proceso de control de nivel utilizamos un esquema básico de arranque y parada de un motor. Tanto para el arranque como para la parada utilizaremos interruptores provistos por el PLC. Al iniciar el proceso se encenderá un LED indicando que el sistema arrancó satisfactoriamente.

Por defecto, nuestro sistema de control estará en modo LOCAL y AUTOMÁTICO. Local significa que el control se realiza desde el mismo PLC y el software de programación, y “Automático” quiere decir que el usuario podrá elegir un Set-Point de altura deseada, y el sistema se encargará de regular el caudal automáticamente para llegar a ese valor.

Mediante otro interruptor se podrá intercalar entre modo local y remoto, para poder realizar el control a través del programa SCADA. Del mismo modo, se podrá utilizar otro interruptor para intercalar entre modo automático y MANUAL. Éste último implica que la regulación del caudal queda en manos del usuario hasta llegar a la altura deseada, sin que el sistema interfiera.

Se utilizará un LED para indicar que el modo remoto está habilitado, y otro para el modo manual.

Se incluirá también una entrada digital que simulará una falla en el sistema (p ej. salto de llave térmica), la cual encenderá un LED de falla que utilizaremos como variable de alarma. Además de esta, incluiremos tres alarmas más: Una para cuando la cisterna está vacía, otra para cuando haya una obstrucción en el tubo de llenado, y una última para cuando se exceda cierto nivel de altura en el tanque. Cada una de estas alarmas detendrá el proceso por completo, hasta que se indique que las fallas han sido resueltas.

Una última entrada digital será utilizada como entrada del sensor de caudal, y una salida digital para habilitar el funcionamiento del variador.

Se utilizarán entradas analógicas para medir el nivel de agua, y una salida analógica para regular el caudal.

Actuadores

Variador de velocidad: ALTIVAR 11



El variador de velocidad utilizado en este proyecto es el ALTIVAR 11 modelo: HU18M2E.

Datasheets:

<https://www.inverterdelta.com/wp-content/uploads/2014/05/User-Manual-ATV11.pdf>

Características principales

- Rango de potencia: 0,18 kW a 2,2 kW.
- Tipos de alimentación disponibles:
 - 100 a 120 V monofásica.
 - 200 a 240 V monofásica.
 - 200 a 230 V trifásica.

- Funciones integradas: Configuración de velocidades máxima y mínima, control local mediante botones integrados (RUN/STOP) y potenciómetro para referencia de velocidad.

Entradas y salidas

Entradas lógicas (LI1 a LI4): Permiten la selección de hasta 4 velocidades preestablecidas.

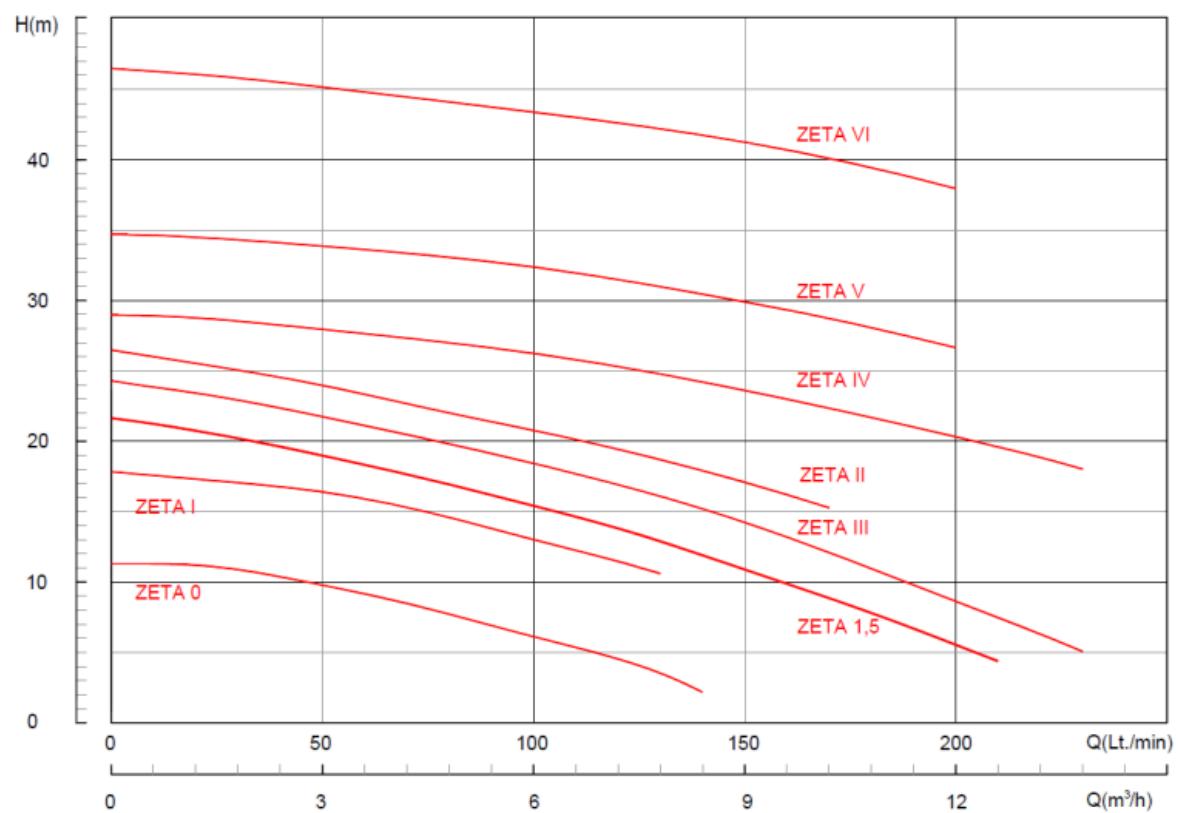
- Entrada analógica (AI1): Para control de velocidad mediante señal externa.
- Salida de relé (R1): Indica fallos o estado del variador.
- Salida analógica (DO): Imagen de la frecuencia del motor.

El Variador de Velocidad es un dispositivo empleado para controlar la velocidad de giro de los motores mediante la variación de frecuencia aplicada en los bornes de alimentación del motor. El modelo que utilizamos es: Variador de velocidad ATV11 HU18M2E. Las entradas de control del variador de velocidad nos permitirán controlar de manera remota las acciones como marcha y parada, seteos de velocidad y hasta la posibilidad de entregar una señal de error mediante un relay que puede ser leída por el PLC.

Bomba

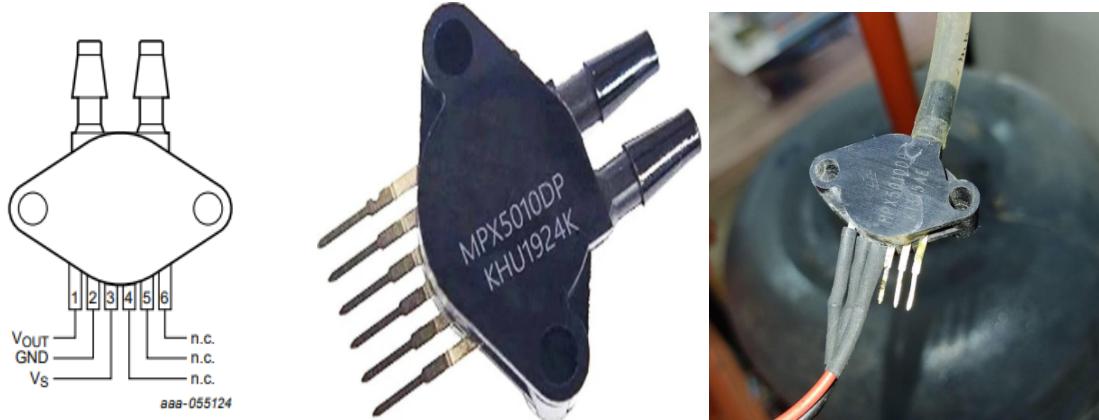
La bomba utilizada es de la marca czerweny con las siguientes características:

- ❖ Q = 20 a 130L/min
- ❖ Tensión 220/380 V
- ❖ Corriente 2,2/1,25 A
- ❖ Frecuencia de trabajo 50Hz
- ❖ Cosφ 0,8



Sensores

Sensor de presión del Tanque: MPX5010



El MPX5010 es un sensor de presión diferencial de la serie MPX de Freescale (ahora NXP), diseñado para medir presiones en un rango de 0 a 10 kPa (0 a 1.02 m H₂O) y convertirlas en una señal eléctrica analógica.

link: <https://www.nxp.com/docs/en/data-sheet/MPX5010.pdf>

Principio de funcionamiento

El MPX5010 funciona con el principio de un transductor piezorresistivo, lo que significa que utiliza una membrana de silicio con resistencias sensibles a la presión. Cuando la presión cambia, la membrana se deforma y altera la resistencia del puente de Wheatstone interno, generando una variación de voltaje proporcional a la presión aplicada.

Características principales

- Rango de medición: 0 a 10 kPa (aproximadamente 0 a 100 cm de columna de agua).
- Salida analógica: 0.2V a 4.7V (con alimentación de 5V).
- Voltaje de alimentación: 4.75V a 5.25V.
- Tipo de sensor: Sensor diferencial (tiene dos puertos de presión).
- Precisión: $\pm 1.5\%$ del rango total.
- Tiempo de respuesta: ~ 1 ms.

Conexionado

- Pin 1 (VCC): 5V DC
- Pin 2 (GND): Tierra
- Pin 3 (VOUT): Salida analógica (conectada a la entrada analógica de un microcontrolador)

Principales características de la hoja de datos

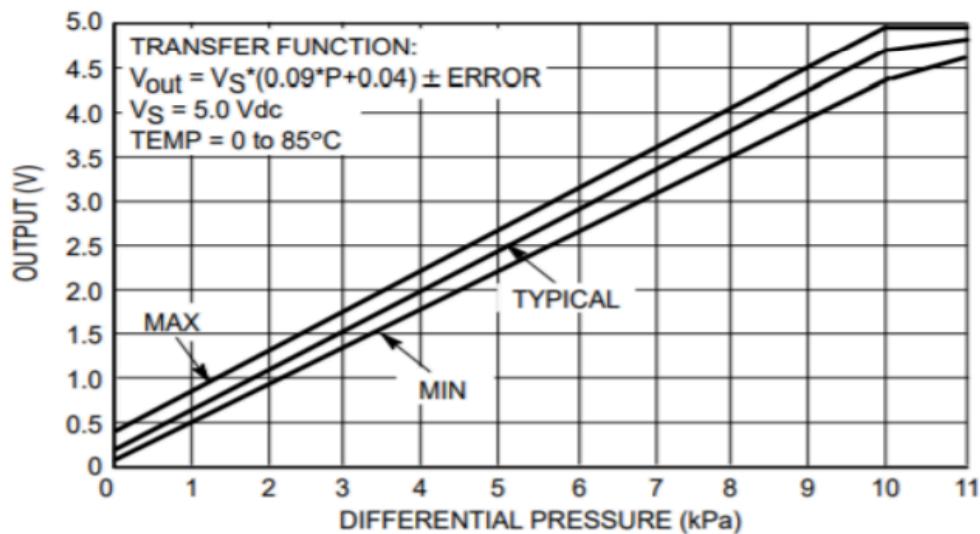
MAXIMUM RATINGS(1)

Parametrics	Symbol	Value	Unit
Overpressure(2) (P1 > P2)	P _{max}	75	kPa
Burst Pressure(2) (P1 > P2)	P _{burst}	100	kPa
Storage Temperature	T _{stg}	-40 to +125	°C
Operating Temperature	T _A	-40 to +125	°C

1. T_C = 25°C unless otherwise noted.
2. Exposure beyond the specified limits may cause permanent damage or degradation to the device.

OPERATING CHARACTERISTICS (V_S = 5.0 Vdc, T_A = 25°C unless otherwise noted, P1 > P2)

Characteristic	Symbol	Min	Typ	Max	Unit
Pressure Range(1)	P _{OP}	0	—	10	kPa
Supply Voltage(2)	V _S	4.75	5.0	5.25	Vdc
Supply Current	I _S	—	7.0	10	mAdc
Sensitivity	V/P	—	450	—	mV/kPa



Sensor de presión para medir caudal: MPX 5700



El MPX5700 es un sensor de presión diferencial de la serie MPX de NXP, diseñado para medir presiones en un rango de 0 a 700 kPa (0 a 7 bar) y convertirlas en una señal eléctrica analógica.

Datasheet: <https://www.nxp.com/docs/en/data-sheet/MPX5700.pdf>

Principio de funcionamiento

El MPX5700 es un transductor piezorresistivo basado en una membrana de silicio con resistencias sensibles a la presión. Al aplicarle una presión diferencial entre sus puertos, la membrana se deforma, alterando la resistencia de un puente de Wheatstone interno y generando una señal de voltaje proporcional a la presión medida.

En aplicaciones de medición de caudal, el sensor mide la diferencia de presión antes y después de un elemento de restricción (como un orificio calibrado), lo que permite calcular el flujo a través de la ecuación de Bernoulli.

Características principales

- Rango de medición: 0 a 700 kPa (0 a 7 bar).
- Salida analógica: 0.2V a 4.7V (para una alimentación de 5V).
- Voltaje de alimentación: 4.75V a 5.25V.
- Tipo de sensor: Diferencial (posee dos puertos de presión).
- Precisión: $\pm 2.5\%$ del rango total.
- Tiempo de respuesta: ~ 1 ms.

Conexionado

El sensor tiene 3 pines principales:

1. VCC (Pin 1): Alimentación de 5V.
2. GND (Pin 2): Tierra.
3. VOUT (Pin 3): Señal de salida analógica proporcional a la presión diferencial.

Para medir caudal, se conecta un puerto del sensor antes de un elemento restrictor y el otro después. La diferencia de presión se usa para calcular el caudal mediante la ecuación:

$$Q = C \sqrt{\Delta P}$$

Principales características de la hoja de datos

Operating Characteristics

Table 1. Operating Characteristics ($V_S = 5.0$ Vdc, $T_A = 25^\circ\text{C}$ unless otherwise noted, $P_1 > P_2$. Decoupling circuit shown in required to meet electrical specifications.)

Characteristic	Symbol	Min	Typ	Max	Unit
Pressure Range ⁽¹⁾ Gauge, Differential: MPX5700D Absolute: MPX5700A	P_{OP}	0 15	— —	700 700	kPa
Supply Voltage ⁽²⁾	V_S	4.75	5.0	5.25	Vdc
Supply Current	I_O	—	7.0	10	mAdc
Zero Pressure Offset ⁽³⁾ Gauge, Differential (0 to 85°C) Absolute (0 to 85°C)	V_{off}	0.088 0.184	0.2 —	0.313 0.409	Vdc
Full Scale Output ⁽⁴⁾ (0 to 85°C)	V_{FSO}	4.587	4.7	4.813	Vdc
Full Scale Span ⁽⁵⁾ (0 to 85°C)	V_{FSS}	—	4.5	—	Vdc
Accuracy ⁽⁶⁾ (0 to 85°C)	—	—	—	±2.5	%V _{FSS}
Sensitivity	V/P	—	6.4	—	mV/kPa
Response Time ⁽⁷⁾	t_R	—	1.0	—	ms
Output Source Current at Full Scale Output	I_{O+}	—	0.1	—	mAdc
Warm-Up Time ⁽⁸⁾	—	—	20	—	ms

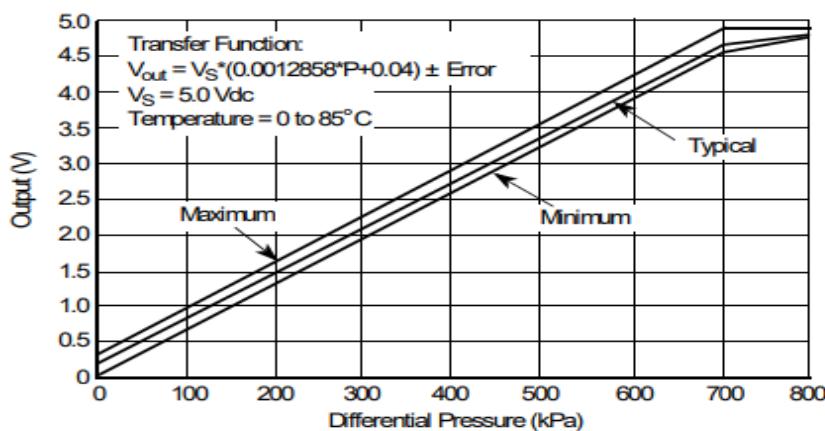


Figure 2. Output vs. Pressure Differential

Sensor capacitivo: STF 2500 B



Principio de Funcionamiento

Cuando un material entra en el campo de detección del sensor, la capacitancia cambia y activa la salida del sensor, permitiendo la detección sin contacto.

Datasheet

https://fertron.com.br/images/shop/Folder_Sensor-Capacitivo-STF2500C.pdf?utm_source=chatgpt.com

Características principales

- Alimentación: 12 a 30 Vcc
- Consumo: 25 mA en 24 Vcc
- Sensibilidad: 0-50mm
- Salidas: Modo HI y modo LOW, RL (150 mA max)

Caudalímetro: Sensor YF-S201

Principio de Funcionamiento

Funciona en función al efecto HALL. Por lo tanto, cuando el agua fluye a través del conjunto del rotor, el rotor gira y el sensor produce un pulso eléctrico. Con estos pulsos generados, se puede calcular la cantidad de agua que atraviesa el sensor.

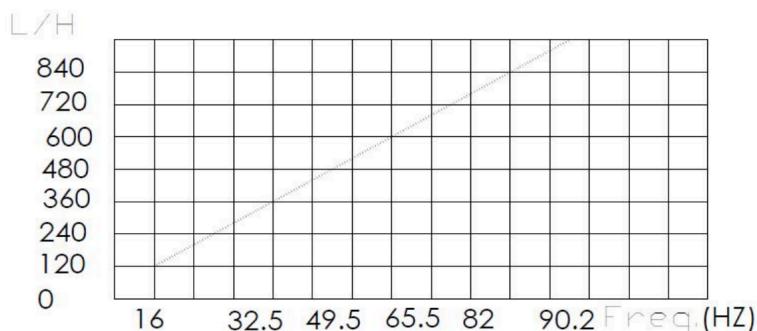
Datasheet

https://components101.com/sites/default/files/component_datasheet/YF-S201-Datasheet.pdf

Circuit:

Red: Positive
Black: GND
Yellow: Output signal

Flow Range: 100L/H-/1800H-L/H		
Flow (L/H)	Frezq.(Hz)	Erro range
120	16	
240	32.5	
360	49.3	
480	65.5	
600	82	
720	90.2	

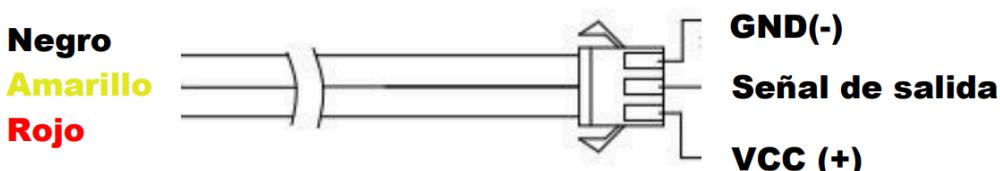


Características principales

- Voltaje de trabajo: DC 4.5V ~ 24V
- Voltaje nominal: DC 5V ~ 18V
- Corriente máxima de trabajo: 15mA (DC 5V)
- Capacidad de carga: ≤10mA (DC 5V)
- Rango de caudal: 1 ~ 30 L/min
- Temperatura de operación: ≤80 °C
- Temperatura del líquido: ≤120 °C

El gráfico anterior muestra la frecuencia de salida en función de la cantidad de flujo medido.

Método de Conexión:



Medición del caudal

El caudal Q (L/min) se obtiene a partir de la frecuencia de los pulsos (Hz) con la siguiente ecuación:

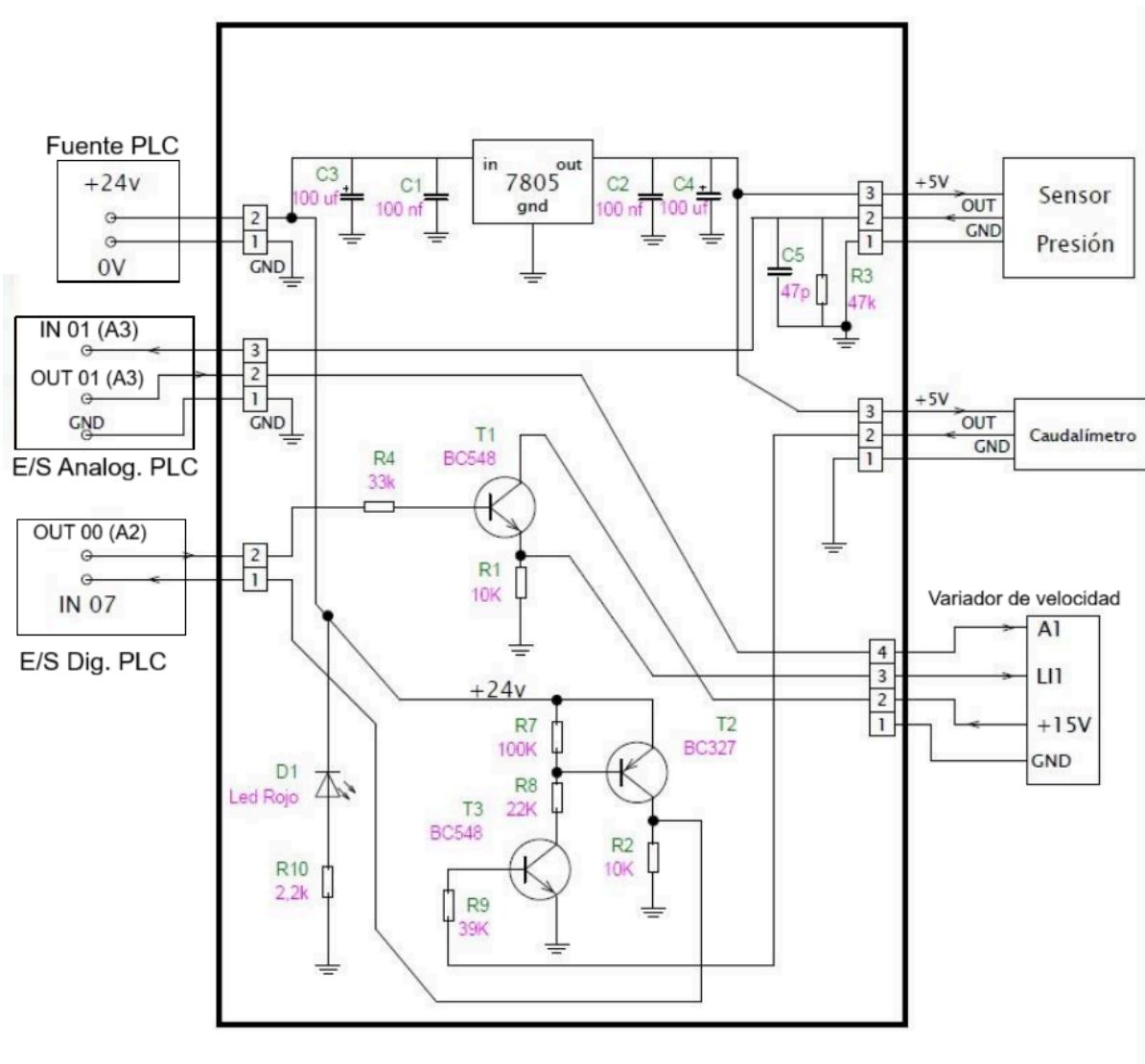
$$Q = \frac{\text{Frecuencia (Hz)}}{7.5}$$

Acondicionamiento de señal

Como se señaló al principio del informe, el sistema cuenta con un circuito para acondicionar señales. Este circuito es de suma importancia, puesto que el nivel de tensión que manejan las entradas y salidas del PLC (24 [V]) no es el mismo que con el que trabajan otras etapas del circuito, como pueden ser los sensores o el variador de velocidad. Por lo tanto, es necesario adaptar estos niveles según lo requiera la situación.

A continuación se muestra una imagen de dicho circuito y su esquemático (Las variables y sus correspondientes entradas y salidas asociadas serán detalladas en la sección posterior).





En la figura podemos observar que la placa de adaptación de señales es alimentada desde la fuente del PLC (+24V).

El sensor diferencial de presión y el caudalímetro necesitan ser alimentados con una tensión de +5V, para lo cual se utiliza un regulador de tensión 7805.

La salida del sensor de presión (utilizada para medir la altura del tanque) es analógica, con un nivel de tensión máximo de +5V. Por lo tanto, no hace falta más que conectarla a la entrada del módulo analógico del PLC IN01 (del módulo A3).

La salida del caudalímetro tiene un nivel de +5V, la cual tendremos que adaptar a un nivel de +24V para conectarla en la entrada digital del PLC IN07.

La salida digital del PLC OUT 00 (del módulo A2), que controla el arranque y parada del variador de velocidad, tiene un nivel de tensión de +24V. Por lo tanto, tendremos que adaptarlo para poder comandar la entrada LI1 del variador de velocidad.

La salida analógica OUT 01 (del módulo A3) del PLC, que manipula la frecuencia del variador de velocidad, está configurada para un rango de 0 a +10V por lo que se conecta directamente a la entrada A1 del variador de velocidad.

Configuración y programación del PLC

Variables utilizadas

Las variables más importantes utilizadas en este **proyecto** son las siguientes:

- Entradas digitales

Entradas digitales		
Variable	Nombre	Descripción
IN00 (A2)	ON	Interruptor que inicia el proceso
IN01 (A2)	OFF	Interruptor que detiene el proceso
IN02 (A2)	Manual_EN_Switch	Intercala entre modo automático (Por defecto) y manual
IN03 (A2)	Remote_EN_Switch	Habilita el modo remoto cuando está encendido
IN04 (A2)	b_Electrical_Failure	Simula una falla eléctrica
IN07 (A2)	Flowmeter_In	Entrada del sensor de caudal
IN00 (A5)	b_Cistern_Water	Sensor capacitivo. Verifica la presencia de agua en la cisterna
IN01 (A5)	b_Pump_Pressure_Fault_Resolved	Indica que la falla por obstrucción en la tubería ha sido resuelta
IN02 (A5)	b_Cistern_Water_Resolved	Indica que la falla por cisterna vacía ha sido resuelta
IN03 (A5)	b_Electrical_Failure_Resolved	Indica que la falla eléctrica ha sido resuelta
IN04 (A5)	b_Max_Height_Resolved	Indica que la falla por altura máxima ha sido resuelta

- Entradas analógicas

Entradas analógicas		
Variable	Nombre	Descripción
IN01 (A3)	Analog_Pressure_Read	Sensor de presión del tanque (Utilizado para medir el nivel de agua)
IN02 (A3)	Pressure_Pump_Analog	Sensor de presión de la tubería (Para medir la presión por obstrucción)

- Salidas digitales

Salidas digitales		
Variable	Nombre	Descripción
OUT00 (A2)	Process	Activa el variador de velocidad
OUT01 (A2)	Process_Light	Luz indicadora de que el proceso dió inicio
OUT02 (A2)	Manual_EN_Light	Luz indicadora de que el modo manual está activo
OUT03 (A2)	Remote_EN_Light	Luz indicadora de que se habilitó el modo remoto
OUT01 (A5)	b_Light_Pump_Pressure_Failure	Luz indicadora de falla por obstrucción en tubería
OUT02 (A5)	b_Light_Failure_Water	Luz indicadora de falla por cisterna vacía
OUT03 (A5)	b_Light_Electrical_Failure	Luz indicadora de falla eléctrica (Simulada)
OUT04 (A5)	b_Light_Maximum_Height_Exceeded	Luz indicadora de falla por exceso de altura en tanque

- Salidas analógicas

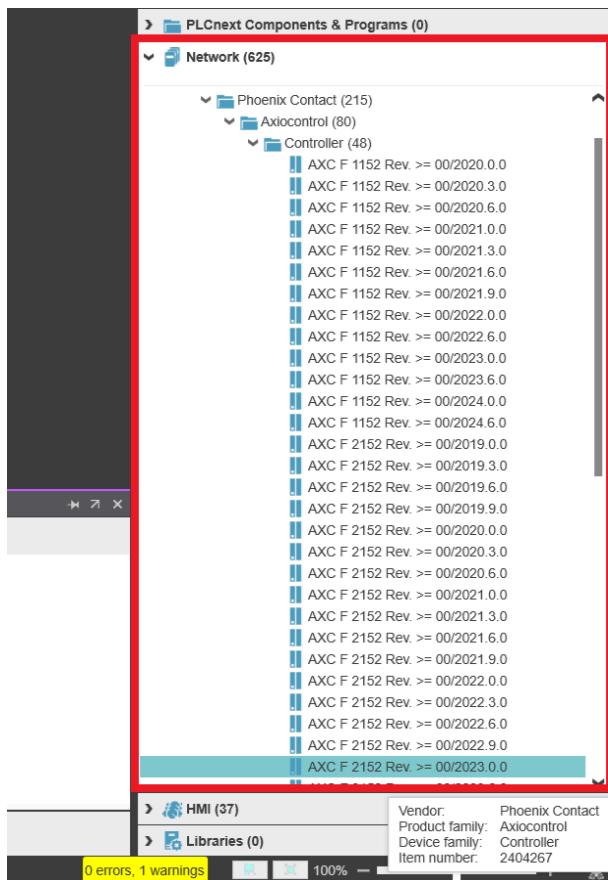
Salidas analógicas		
Variable	Nombre	Descripción
OUT01 (A3)	PID_Output	Indica al variador de velocidad la frecuencia que debe proporcionar al motor de la bomba

- Variables locales y auxiliares

Variables locales y auxiliares		
Variable	Nombre	Descripción
-	Remote_ON	Botón de marcha manipulado por la interfaz SCADA
-	Remote_OFF	Botón de parada manipulado por la interfaz SCADA
-	Pressure_Voltage_In_Pump	Nivel de tensión que representa la presión en tubería
-	Pressure_In_The_Pump	Presión por obstrucción de tubería (kPa)
-	DIRECTN	Variable que indica al PID que realice control directo
-	Remote_Manual_EN_Switch	Botón de habilitación de modo manual manipulado por la interfaz SCADA
-	Offset_Height	Valor del offset requerido para calcular la altura
-	Height_Percentage	Porcentaje de la altura del tanque (Necesario para interfaz SCADA)
-	Height_Level	Altura calculada a partir de la lectura de presión
-	Sampling_freq	Frecuencia de muestreo del PID (En nuestro caso 0.1s)
-	Local_Set_Point	SetPoint de altura para modo local (Variable local)
-	Local_Set_Point_Value	Valor del setpoint local (Variable externa para que pueda leerse desde la interfaz SCADA)
-	Remote_Set_Point	SetPoint de altura para modo remoto
-	KP	Constante de proporcionalidad del PID
-	TI	Constante integrativa del PID
-	TD	Constante derivativa del PID
-	Manual_Velocity	Velocidad de flujo manipulada en modo manual
-	Maximum_Height_Exceeded	Detiene el sistema cuando hay falla por altura máxima. Se resetea con la entrada "Resolve" correspondiente
-	System_Failure	Detiene el sistema cuando hay falla eléctrica. Se resetea con la entrada "Resolve" correspondiente
-	Failure_Water	Detiene el sistema cuando hay falla por cisterna vacía. Se resetea con la entrada "Resolve" correspondiente
-	Pump_Pressure_Failure	Detiene el sistema cuando hay falla por obstrucción en tubo. Se resetea con la entrada "Resolve" correspondiente
-	Flowmeter_Pulses	Contador de pulsos del caudalímetro
-	Flowmeter_Output	Medida de caudal
	Flowmeter_Time	Variable auxiliar para medir caudal
	Flowmeter_In_Last	Variable auxiliar que recuerda el estado anterior de la entrada del caudalímetro
	Samples_Pressure	Arreglo que contiene muestras del sensor de presión a través del tiempo
-	Mean_Pressure	Promedio de valores leídos por el sensor de presión (Para reducir el ruido)

Configuraciones previas

Antes de comenzar a programar, debemos crear un nuevo proyecto en el software PLCNext Engineer, y luego elegir el controlador correspondiente a nuestro PLC, según el modelo y el firmware que posea (En este caso, el AXC F 2152, firmware 2023.0.0).



A continuación, debemos indicar al software la presencia de los módulos de comunicación Axioline y Profinet (de ser necesario). Para ello podemos o bien detectar automáticamente la presencia de estos módulos, o seleccionarlos manualmente desde la pestaña “Axioline” y “Profinet” como se ve en las imágenes.

PLANT

Project : axc-f-2152-1 / AxioLine F

Device List

Type	Function	Location
AXL F DI8/1 DO8/1 1H		
AXL F DI8/1 DO8/1 XC 1H		
AXL F DI8/2 24DC 1F		
AXL F DI8/2 48/60DC 1F		
AXL F DI8/2 110/220DC 1F		
AXL F DI8/3 DO8/3 2H		
AXL F DI16 NAM XC 1F Rev. >= 01/1.00		
AXL F DI16 NAM XC 1F Rev. >= 01/1.04		

Search

Project

axc-f-2152-1 : AXC F 2152

- PLCnext (2)
- IEC 61131-3
- HMI Web Server
- OPC UA
- Profinet
- AxioLine
- PLCnext

Devices

- Axiocontrol
- Devices
- Phoenix Contact
- AXL F
- Analog input
- Analog input and output
- Analog output
- Digital input
- Digital input and output
- Digital output
- Function
- Gateway
- Profisafe
- Pulse width modulation
- SafetyBridge

PLANT

Project : axc-f-2152-1 / AxioLine F

Device List

Type	Function	Location
AXL F BK PN Rev. >= 1/V1.0.1		
AXL F BK PN Rev. >= 1/V1.0.4		
AXL F BK PN SC-RJ Rev. >= 1/V1.0.1		
AXL F BK PN TPS Rev. >= 1/V1.1.0		
AXL F BK PN TPS Rev. >= 1/V1.3.0		
AXL F BK PN TPS Rev. >= 1/V1.3.2		
AXL F BK PN TPS XC Rev. >= 1/V1.3.1		
AXL F BK PN TPS XC Rev. >= 1/V1.3.2		
AXL F BK PN XC Rev. >= 1/V1.0.3		
CLOUD SERVICE/WEATHER Rev. >= 01/V1.0.0		
IL PN BK DI8 DO4 2SCRJ Rev. >= 3/V3.4.1		

Search

Project

axc-f-2152-1 : AXC F 2152

- PLCnext (2)
- IEC 61131-3
- HMI Web Server
- OPC UA
- Profinet
- AxioLine
- PLCnext

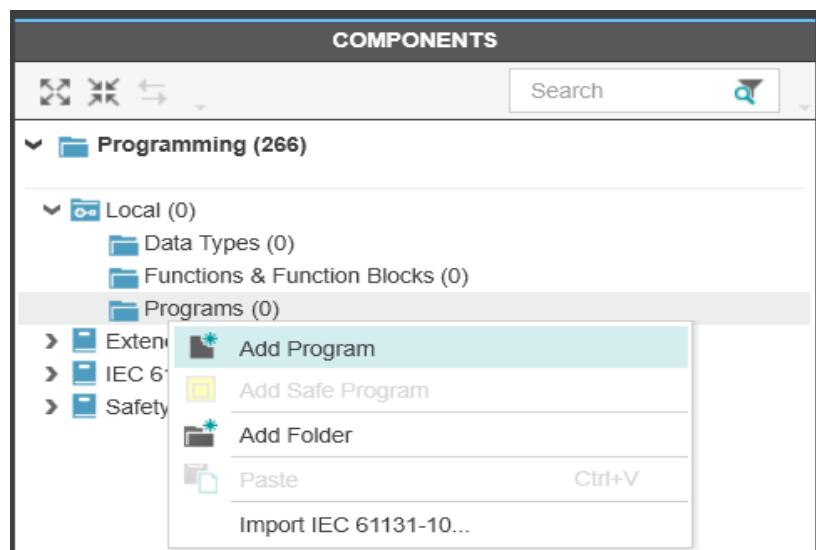
Devices

- Bus coupler
- AXL F XC
- Bus coupler
- Inline Profinet
- Devices
- Phoenix Contact
- Inline
- Bus coupler
- Inlinecontrol
- Controller Device
- Proficloud
- Devices
- Phoenix Contact
- PROFICLOUD
- I/O
- Service

Luego, se deben configurar las entradas y salidas analógicas según el funcionamiento de los sensores y actuadores (En este caso, todos funcionan entre 0 y 5 V). De ese modo, la configuración del módulo analógico queda de la siguiente manera:

Input channel 1	
Measuring range:	0 V ... 5 V
Filter:	30 Hz
Mean value:	16-sample
Input channel 2	
Measuring range:	0 V ... 5 V
Filter:	30 Hz
Mean value:	16-sample
Output channel 1	
Output range:	0 V ... 5 V
Substitution behavior:	maintain last values
Output channel 2	
Output range:	0 V ... 10 V
Substitution behavior:	maintain last values

Con todas las configuraciones anteriores, podemos comenzar a programar. Para ello, creamos un nuevo programa dando clic en el botón correspondiente ubicado en el panel derecho.



Una vez allí elegimos el lenguaje de programación. En nuestro caso utilizaremos programación por bloques y ladder.

Select the programming language of your first worksheet below

```
1 IF condition = TRUE THEN  
2   opC := opA AND opB;  
3 END_IF
```

Add ST Code Worksheet

Add LD/FBD Code Worksheet

Network (1) Network One

Add NOLD Code Worksheet

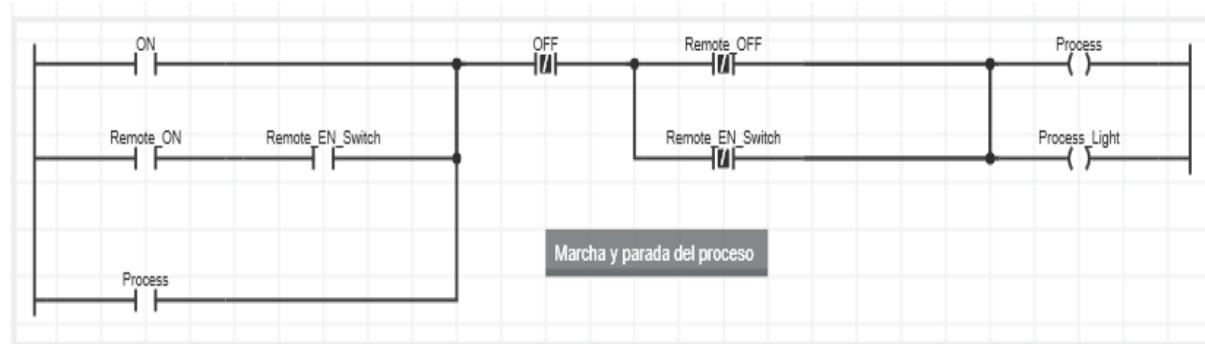
Programación

Para realizar este sistema de control no requerimos solamente de un programa, sino de tres: Un programa principal, otro para leer el sensor de caudal y un tercero para la lectura del sensor de presión del tanque. A continuación se detallan cada uno de ellos.

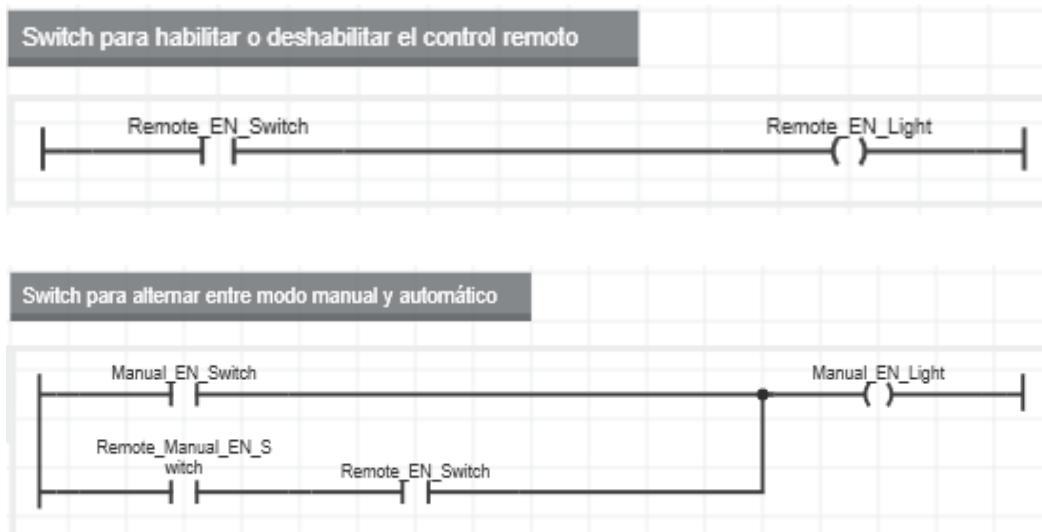
Programa principal

Es el programa central de nuestro proyecto. Se encarga del control en sí mismo así como de administrar las entradas y salidas digitales. En nuestro caso llamamos a este archivo “ProyectoTanque”, y lo programamos en lenguaje de bloques y ladder. A continuación se detallan cada una de sus partes.

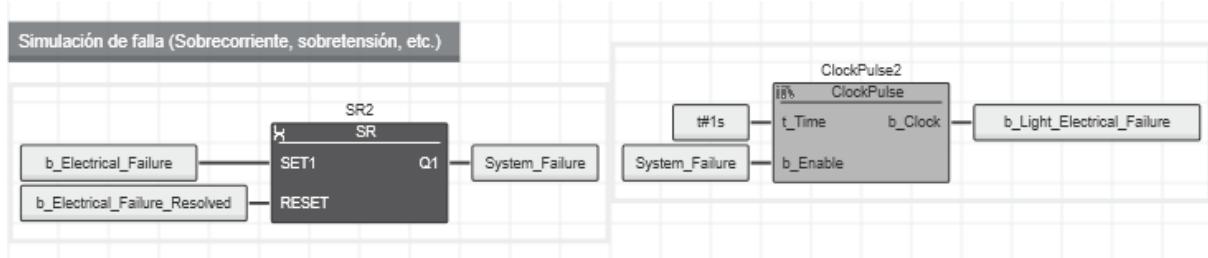
- **Marcha y parada del proceso:** Utiliza un esquema típico de arranque y parada de un motor. Adicionalmente incluye los botones de marcha y parada remotos.



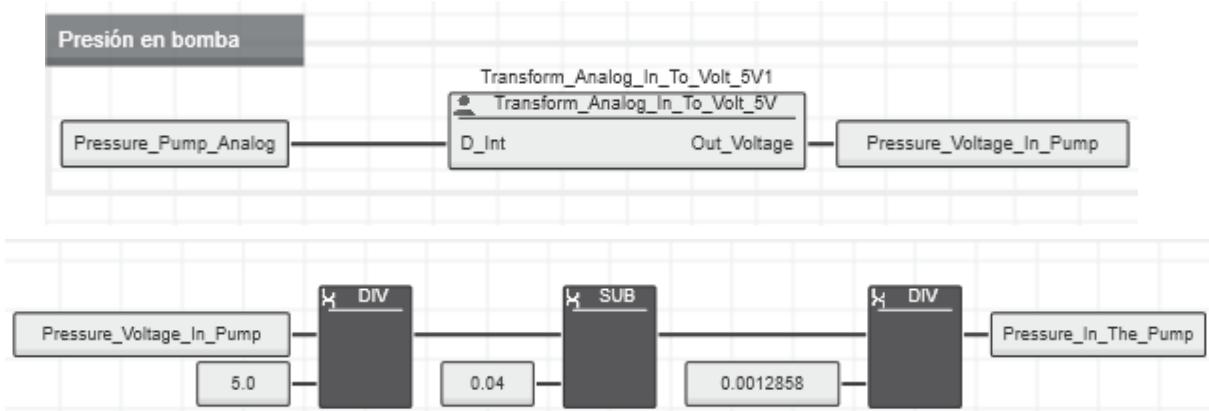
- **Switches para alternar modos:** Incluye un switch para alternar entre modo manual y automático, y otro para alternar entre modo local y remoto. Los LED indicadores se encienden cuando se activa el modo manual y cuando se habilita el modo remoto respectivamente.



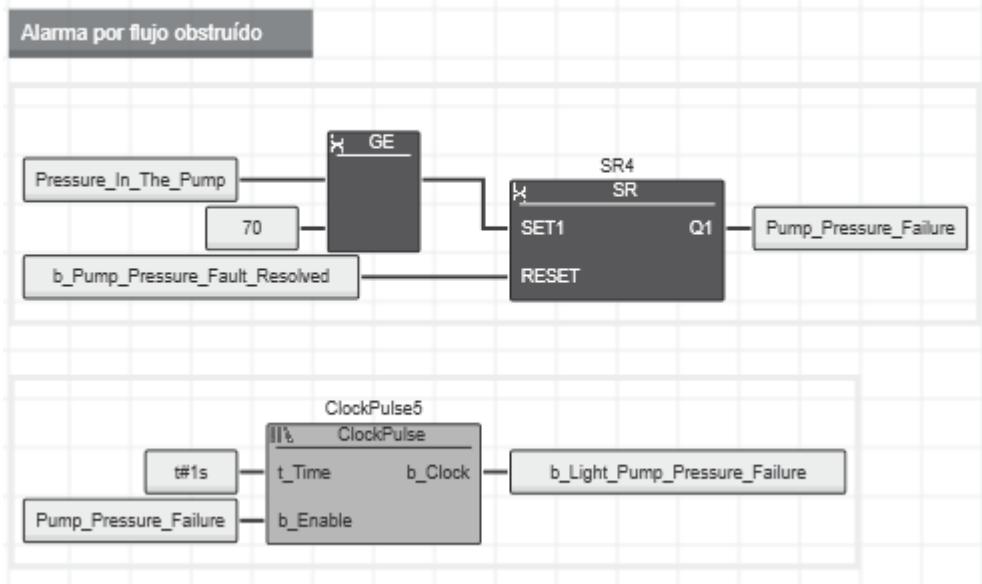
- **Alarma por falla eléctrica:** Una entrada digital simulará una falla genérica en el sistema (Sobretensión, sobrecorriente, calentamiento excesivo, etc.), la cual detendrá el proceso y activará una luz titilante. Esta alarma podrá ser desactivada con otra entrada digital (b_Electrical_Failure_Resolved), que indica que la falla ya ha sido atendida y corregida.



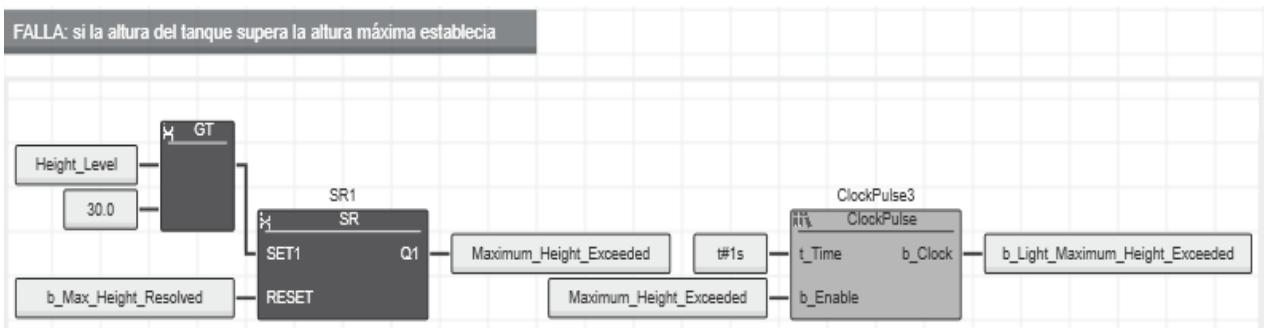
- **Presión en bomba:** La función de este circuito es la de leer el valor de presión del sensor ubicado en la tubería. Comienza leyendo el valor analógico, lo transforma a un valor de entre 0 y 5 [V], y finalmente utilizando las indicaciones del datasheet se calcula el valor de presión en KiloPascales.



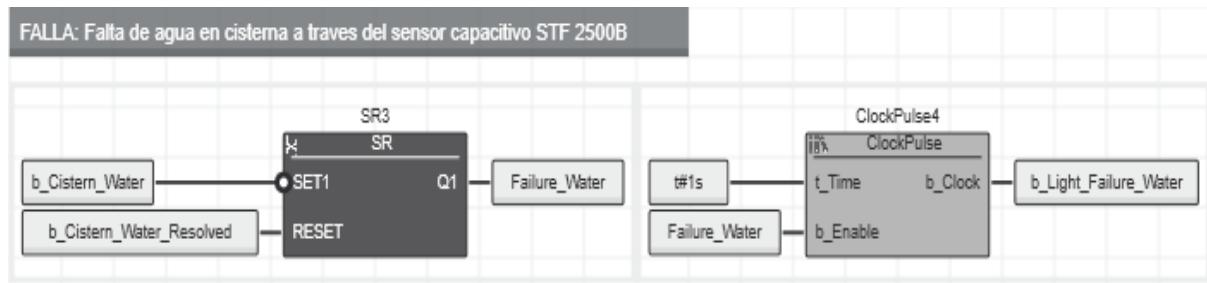
- **Alarma por obstrucción en tubería:** Este circuito utiliza el valor de presión leído en el circuito anterior, y activa una alarma cuando este valor supera los 70 kPa (Tubería obstruida). La alarma activa una luz titilante y detiene el proceso por completo. Puede desactivarse mediante otra entrada digital (b_Pump_Pressure_Fault_Resolved).



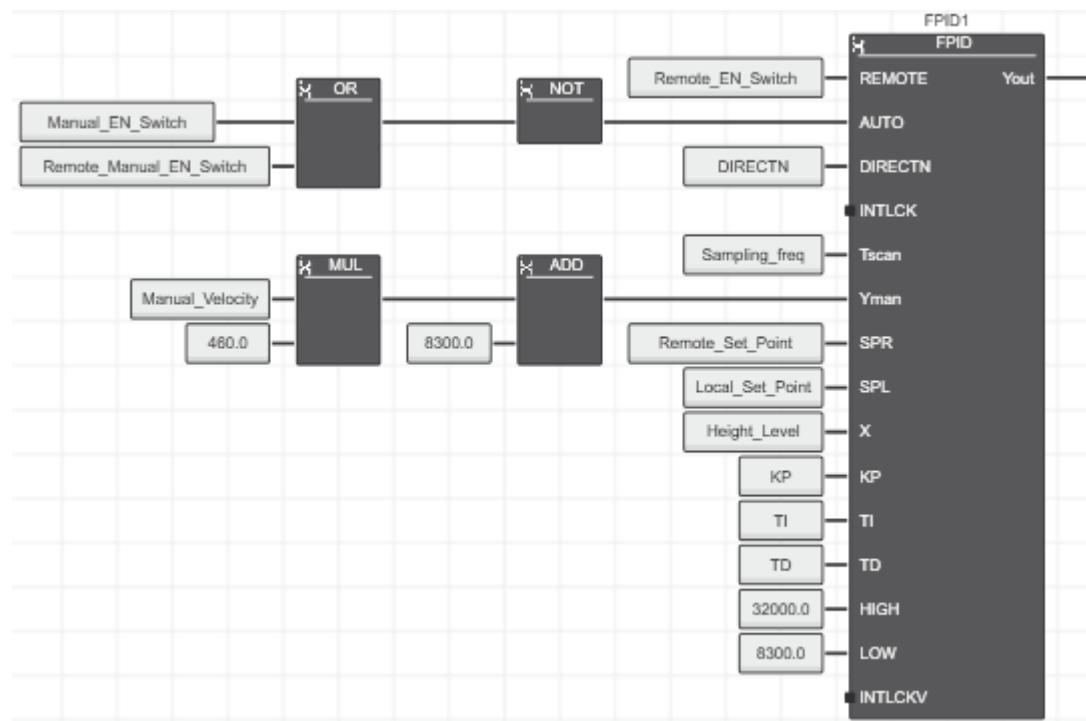
- **Alarma por exceso de altura:** Alarma que se activa cuando el nivel de agua supera los 30 cm. Esto activa una luz titilante y detiene el proceso. Dicha alarma puede desactivarse mediante otra entrada digital (b_Max_Height_Resolved).



- **Alarma por falta de agua en cisterna:** Del mismo modo que las otras alarmas, ésta detiene el proceso y activa una luz cuando el sensor capacitivo ubicado en la cisterna detecta ausencia de agua. Puede desactivarse mediante su entrada “Resolved”.

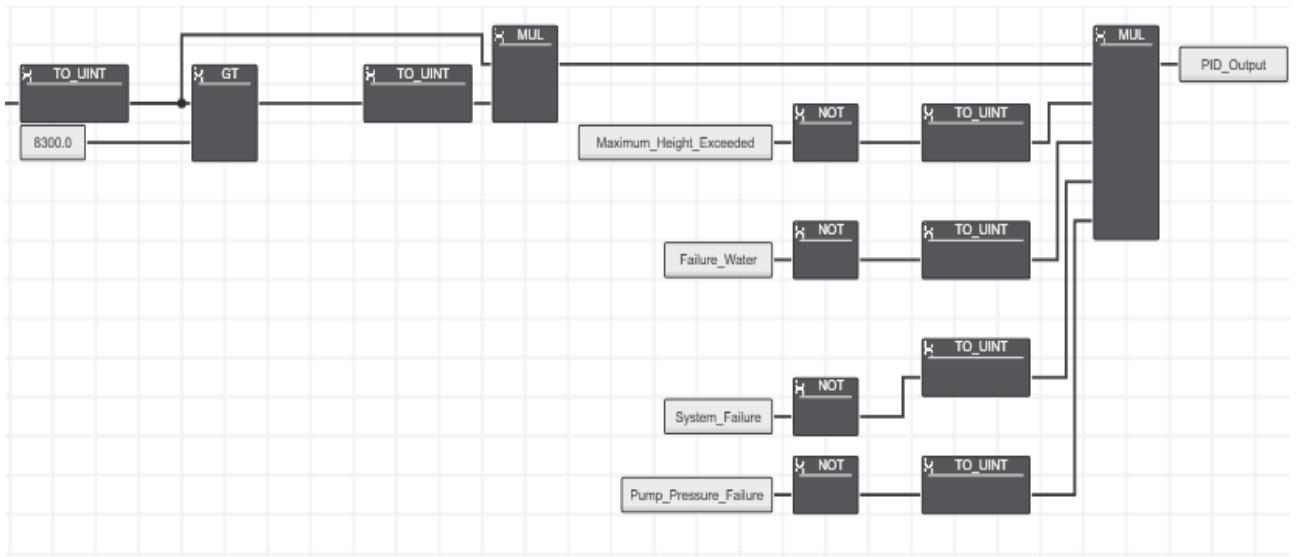


- **PID:** El bloque de control proporcional, integral y derivativo (PID), es el encargado de manipular la salida del sistema. Aquí se introducen los valores del setpoint, la frecuencia de muestreo, las medidas del sensor, entre otros. Su configuración será detallada posteriormente.



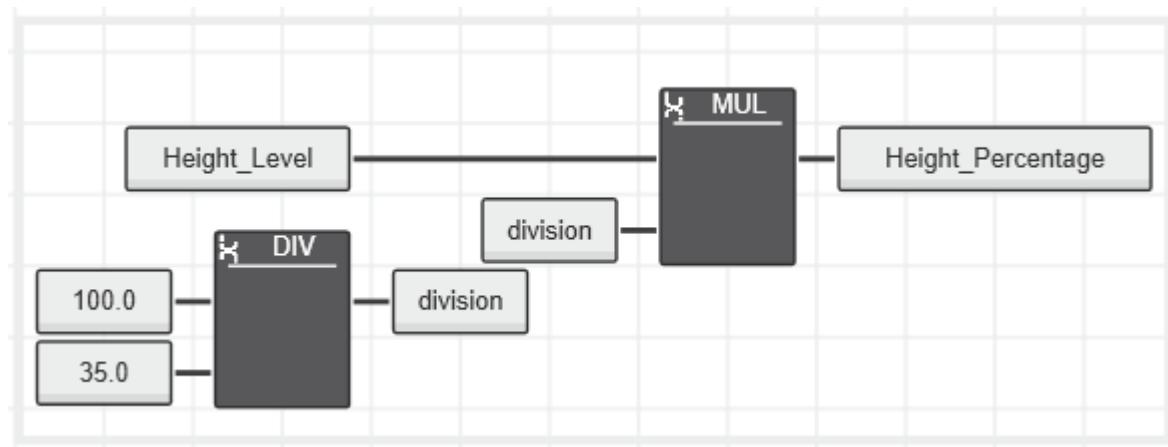
- **Salida:** La salida del PID será un número de 16 bits, es decir, aproximadamente desde -32000 a +32000. Sin embargo, en ese intervalo hay valores no válidos como los negativos, es por eso que nos limitamos solo a trabajar con valores positivos. Por otro lado, hay valores positivos para los cuales el variador trabaja pero no tiene la fuerza suficiente para bombear el agua. Luego de algunas pruebas, determinamos que el valor mínimo para bombear el agua es de 8300.

Debido a todo lo anterior, necesitamos un circuito externo que limite estos valores, y además que detenga el proceso cuando alguna de las cuatro alarmas se active. El circuito que cumple todo lo anterior es el siguiente.

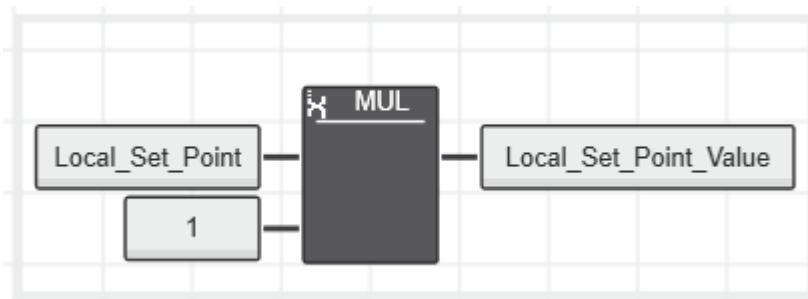


Si el valor de Yout es menor a 8300, o bien alguna de las cuatro alarmas se activa (TRUE), la salida enviada al variador (PID_Output), pasará a ser automáticamente cero.

- **Porcentaje de altura:** Un sencillo circuito auxiliar que nos permite calcular cuánto porcentaje del tanque está lleno. Esto será útil para la interfaz SCADA.



- **Valor actual del Set Point Local:** Al no poder configurar como externa la variable del SetPoint local (puesto que tiene que ser local para poder cambiar su valor desde el PLCNext), requerimos de una variable extra para poder visualizar su valor en el SCADA. En este caso es “Local_Set_Point_Value”.



Caudalímetro

Es el programa utilizado para leer e interpretar el sensor de caudal. Está escrito en texto estructurado, se ejecuta cada 1 milisegundo y su funcionamiento es el siguiente.

```
1 //Se ejecutará cada un milisegundo
2
3 Flowmeter_Time := Flowmeter_Time + T#1ms;
4
5 [ If Flowmeter_In = TRUE AND Flowmeter_In_Last = FALSE then
6     Flowmeter_Pulses := Flowmeter_Pulses + 1;
7     END_IF
8
9 [ If Flowmeter_Time = T#1000ms then
10    Flowmeter_Output := (Flowmeter_Pulses/7.5)*60;//Litros/seg
11    Flowmeter_Pulses := 0;
12    Flowmeter_Time := T#0ms;
13    END_IF
14
15 Flowmeter_In_Last := Flowmeter_In;
```

En un primer bloque “if” detectamos los pulsos arrojados por el caudalímetro, actuando como un detector de flanco que se ejecuta cada 1 milisegundo.

El segundo bloque “if” se ejecuta una vez por segundo. Lo que hace es tomar la cantidad de pulsos detectados en ese tiempo, calcular el valor del caudal en Litros/Minuto a partir de esa cantidad, y finalmente resetea los valores correspondientes.

La última línea es una variable de memoria, que recuerda el estado de la entrada del sensor.

Promedio Presión

Este último programa se encarga de leer el sensor de presión del tanque, y a partir de ese valor calcular la altura del agua.

Debido a las continuas oscilaciones del agua al llenarse el tanque, los valores medidos por el sensor tendrán una fluctuación considerable, lo que complicaría el control automático. Para reducir estas fluctuaciones, optamos por calcular un promedio de varias muestras tomadas en un cierto intervalo de tiempo, cuyo valor se vaya actualizando continuamente a medida que llegan nuevas muestras. Para lograr esto escribimos el siguiente código:

```
1 //Vamos con un array de 50 muestras el for(i) SE TOMO 10 25 50
2 number_Samples:= 100;
3 for i:=0 to (number_Samples-2) do
4     Samples_Pressure[number_Samples-1-i] := Samples_Pressure[number_Samples-2-i];
5 end_for
6
7 //INGRESAR EL VALOR ACTUAL EN LA POSICION 0 DE Samples_Pressure
8 Transform_Analog_In_To_Volt_5V3(D_Int := Analog_Pressure, Out_Voltage => Voltage);
9 //a := Transform_Analog_In_To_Volt_5V3.Out_Voltage - 0.22;
10 offset_Volt:= (2.74-1.95)*0.09*5;
11 a := Voltage - offset_Volt;
12 Samples_Pressure[0] := ((a/5)-0.04)/0.09;
13
14 for j :=0 to (number_Samples-1) do
15     sum := sum + Samples_Pressure[j];
16 end_for;
17
18 Mean_Pressure := sum/number_Samples;
19 Height_Level := (Mean_Pressure*100000)/(9.8 * 997)+OffSet_Height;
20 sum:=0.0;
```

En primer lugar debemos tener en cuenta que todas las muestras serán almacenadas en un arreglo. Lo que hace la primera parte del código es desplazar los valores de las muestras un lugar hacia la derecha (Es decir, la muestra de índice 0 pasará a tener índice 1, la del índice 1 pasará al lugar de índice 2, etc.).

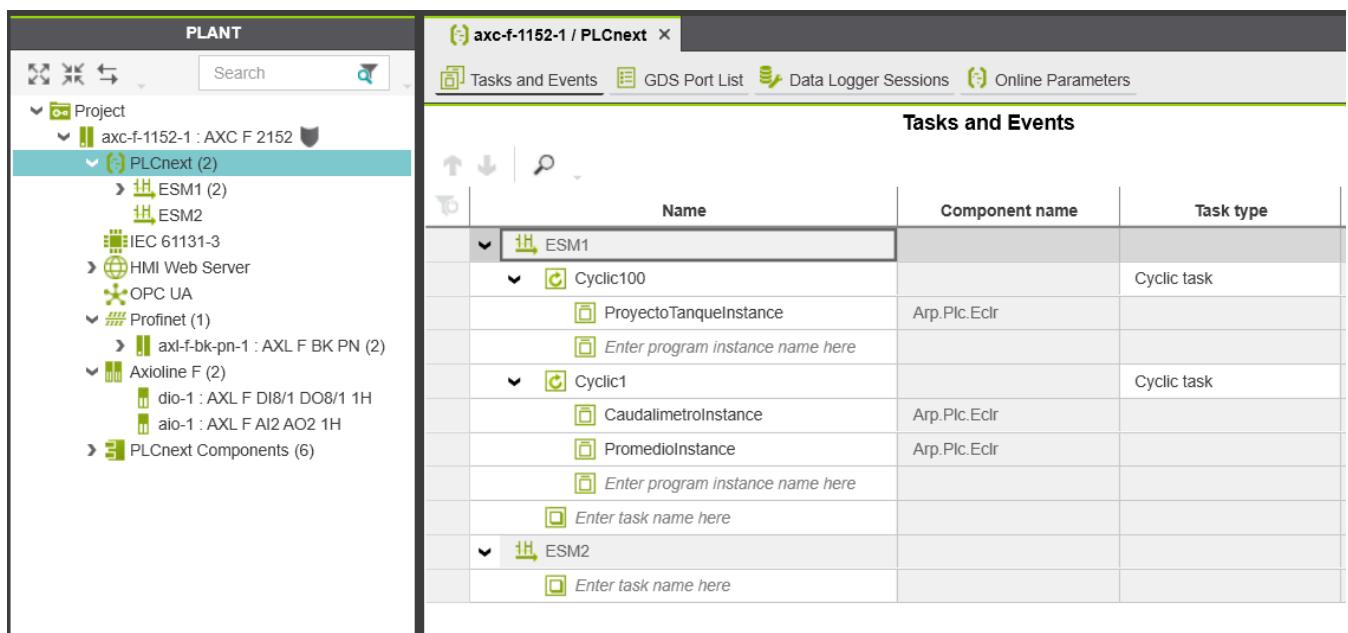
A continuación, el lugar del índice 0 queda libre para insertar la nueva muestra, lo cual es función de la segunda parte del código.

La tercera parte es un bucle “for”, que se encarga de calcular la suma de todos los valores del arreglo, y finalmente con este valor se calcula el promedio de la presión y el nivel del agua. El código finaliza resemando el acumulador “suma”.

Cabe mencionar que la fórmula para calcular la altura se realizó mediante observaciones empíricas, datos del sensor y fórmulas existentes.

Instanciación de los programas

Una vez escritos los tres programas, debemos instanciarlos en el software. Para ello nos dirigimos a la sección “PLCNext” ubicada en el panel de la izquierda, y especificamos las tareas que se deben realizar. En nuestro caso colocaremos dos tareas cíclicas, una que contenga al programa principal y otra que contenga a los restantes.



Debemos especificar tanto los nombres de las tareas como de las instancias. Posteriormente, deslizando a la derecha, seleccionamos los programas a ejecutar en cada instancia y configuramos su ciclo de ejecución. En nuestro caso, la tarea “Cyclic100” se ejecutará cada 100ms, y “Cyclic1” cada 1 ms.

Name	me	Program type	Interval (ms)
1H ESM1			
▼ Cyclic100			100
<input checked="" type="checkbox"/> ProyectoTanqueInstance		ProyectoTanque	
<input type="checkbox"/> Enter program instance name here		Select program type here	
▼ Cyclic1			1
<input type="checkbox"/> CaudalimetroInstance		Caudalimetro	
<input type="checkbox"/> PromedioInstance		PromedioPresion	
<input type="checkbox"/> Enter program instance name here		Select program type here	
<input type="checkbox"/> Enter task name here			
▼ 1H ESM2			
<input type="checkbox"/> Enter task name here			

Configuración del PID

Como se explicó anteriormente, el PID es el bloque principal para controlar el sistema. Su funcionalidad incluye todos los modos descritos anteriormente (Manual, Automático, Local y Remoto). Para configurarlo debemos conectar sus entradas a las variables o constantes correspondientes:

- REMOTE: Esta entrada habilita el modo remoto al estar en 1, por lo que la conectamos directamente al “Switch_Local_Remote”.
- AUTO: Al estar en 1, el PID pasa a funcionar en modo automático, mientras que en 0 funciona en modo manual. Conectamos esta entrada a la negación del “Switch_Manual” porque queremos el modo automático por defecto.
- DIRECTN: Nos permite setear el direccionamiento del control (Directo o inverso). La conectamos a una variable que por defecto está en FALSE (Es decir, control directo).
- INTLCK e INTLCKV: Permiten colocar un valor definido a la salida del PID en caso de error. Estas entradas no se utilizarán.
- TSCAN: Frecuencia de muestreo del PID. Nosotros utilizamos 100 milisegundos.

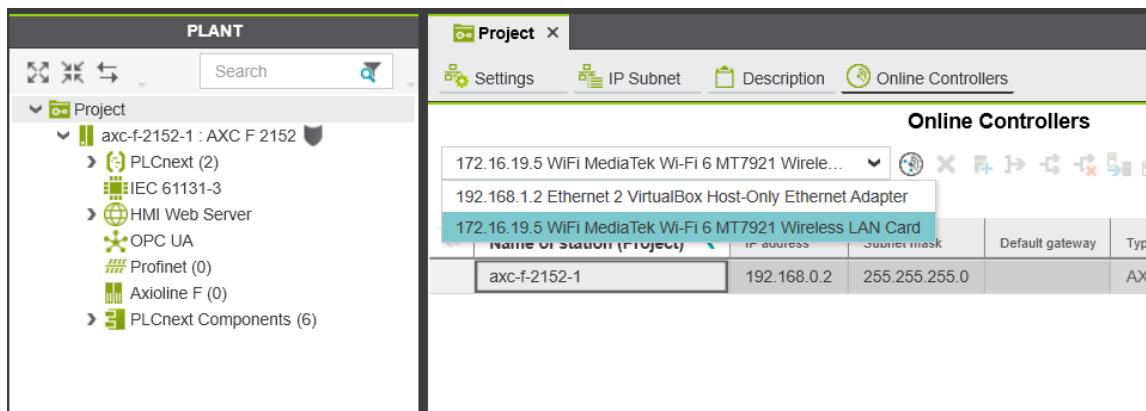
- YMAN: Valor manual. El valor que se conecte a esta entrada se reproducirá a la salida solo cuando la entrada AUTO se encuentre en 0 (Se accione “Switch_Manual”). Este valor será la frecuencia del variador que el usuario desee entre 0 y 50 [Hz]. Utilizamos un circuito externo para transformarla en un valor de entre 8300 y 32000.
- SPR: Set-Point remoto. El PID seguirá este setpoint cuando la entrada REMOTE se encuentre en 1 (Modo remoto habilitado).
- SPL: Set-Point local. El PID seguirá este setpoint cuando la entrada REMOTE se encuentre en 0 (Modo remoto deshabilitado).
- X: Entrada del sensor. Aquí conectamos la altura medida.
- KP, TI y TD: Constantes proporcional, integral y derivativa respectivamente. Luego de varias observaciones, determinamos que los valores más óptimos para estas constantes son: KP = 35.0, TI = 0.01 y TD = 5.0
- HIGH y LOW: Límites inferior y superior de la salida del PID. Como se mencionó antes, se las coloca en 32000 y 8300 respectivamente.
- YOUT: Salida del PID. Pasa por un circuito externo que la acomoda y luego al variador de velocidad.

Conexión entre CPU y PLC

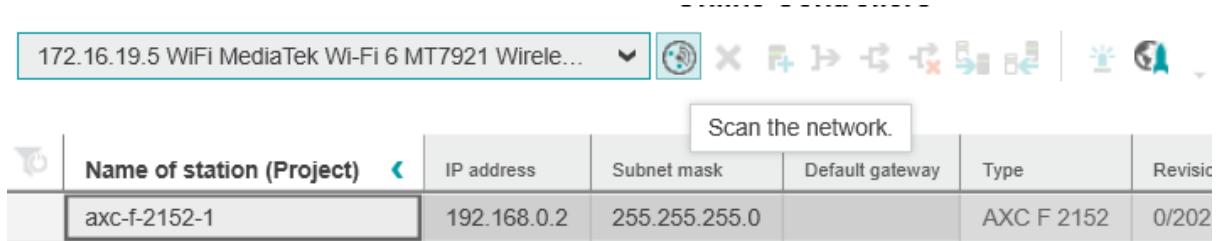
Para cargar los programas y variables que se acaban de crear tenemos dos formas: mediante conexión Ethernet o conexión por red de área local (LAN). En nuestro caso, utilizaremos el segundo método.

Para ello, el segundo puerto Ethernet del PLC debe ir conectado a un router, que será el que genere la red local, y la PC debe conectarse a dicha red.

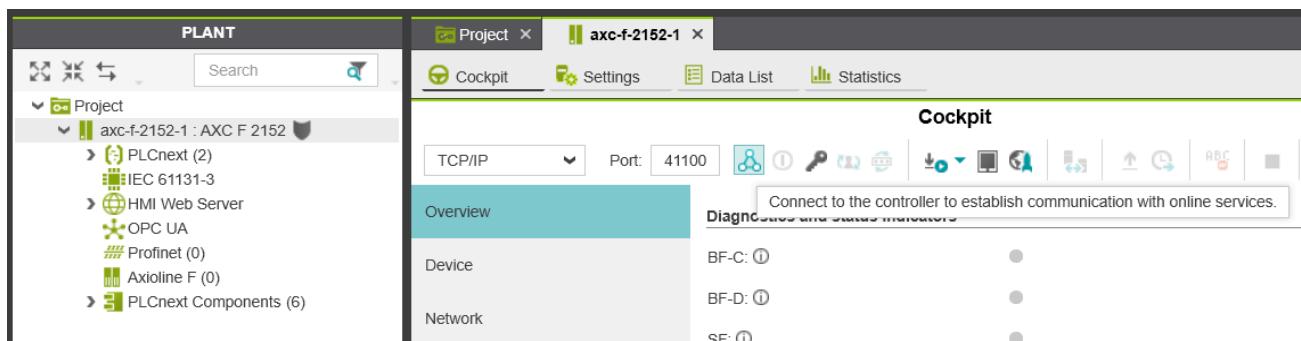
Cuando ambos dispositivos estén conectados, debemos ir a la pestaña “Project”, ubicada en el panel de la izquierda, luego a “Online controllers”, y seleccionamos la opción de buscar controladores por Wi-Fi.



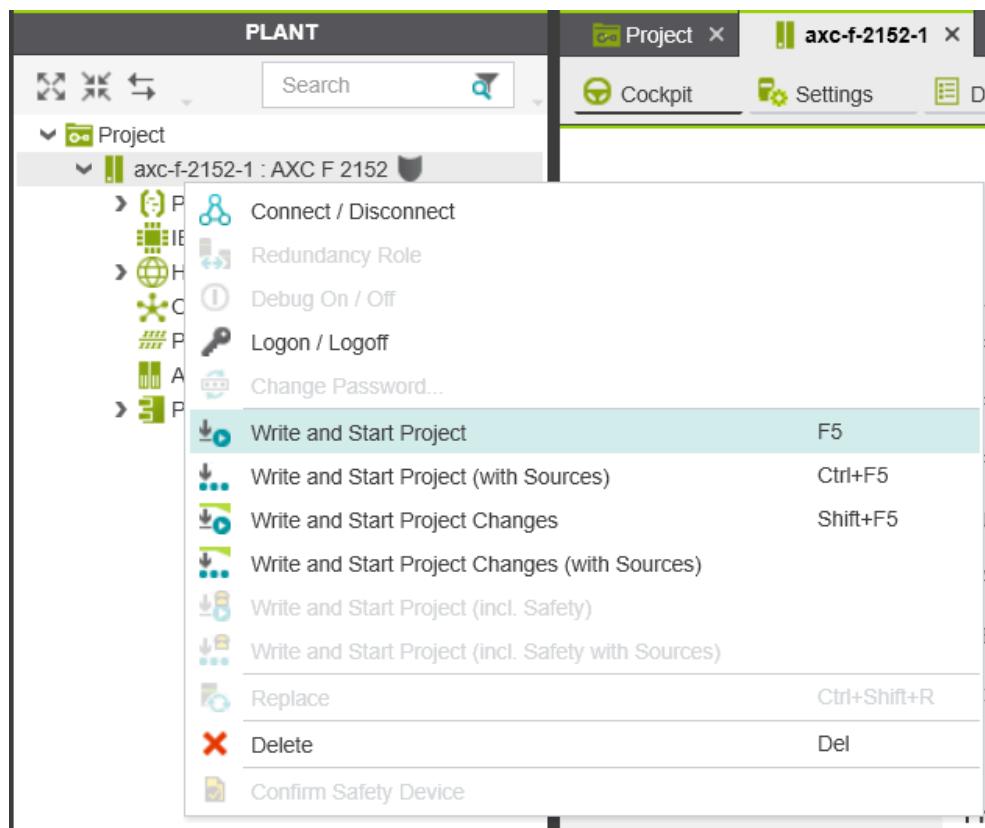
Posteriormente, damos clic en el botón “Scan the network”, y si todo funciona correctamente, debería detectarse automáticamente al PLC conectado a la red.



Una vez detectado el dispositivo, de ser necesario se coloca la IP del mismo en la pestaña “IP address”, luego vamos a la pestaña del PLC ubicada en el panel de la izquierda, nos dirigimos a la sección “Cockpit”, seleccionamos el protocolo de comunicación TCP/IP, y luego oprimimos el botón “Conectar/Desconectar”.



En la ventana emergente introducimos el usuario y la contraseña correspondientes, y finalmente para correr el programa oprimimos el botón “Write and start project”.



Hecho esto, el programa correrá en el PLC satisfactoriamente.

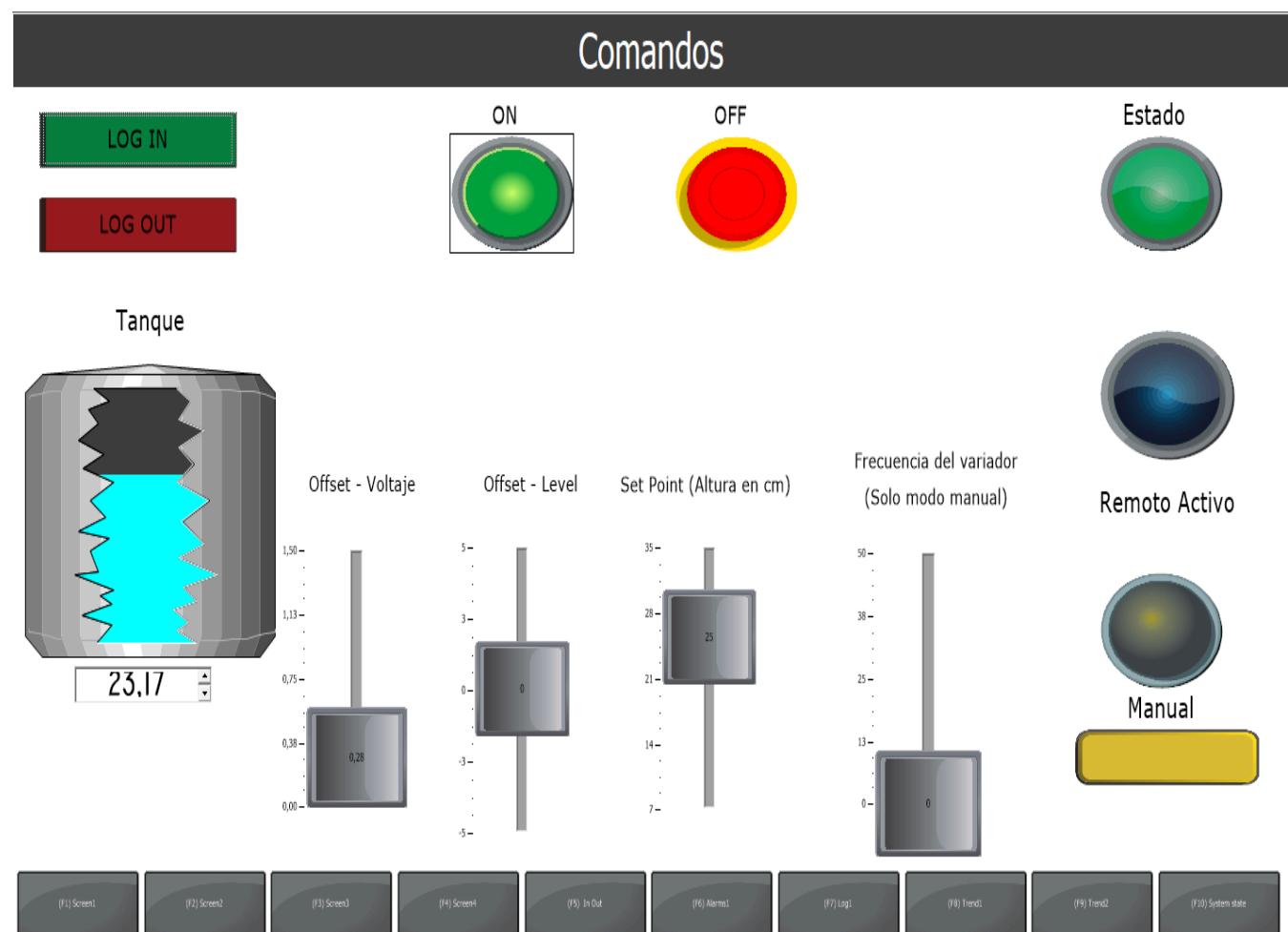
Interfaz SCADA

Ventanas

A continuación, se muestra una vista general de la interfaz SCADA, conformada en su totalidad por cuatro ventanas.

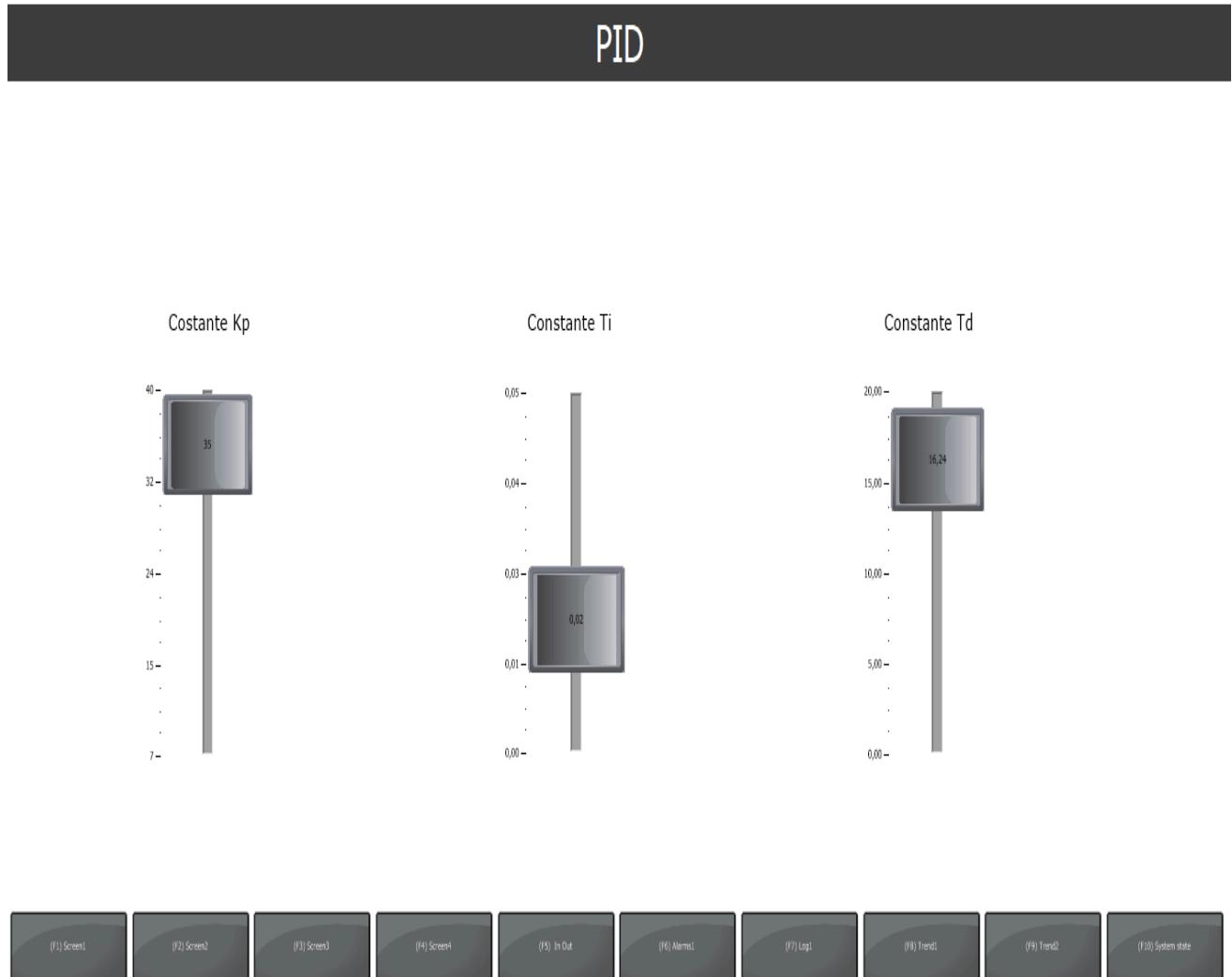
Comandos

Es la ventana principal. Contiene los botones de logueo y todos los elementos necesarios para controlar el proceso de manera remota: Botones de marcha y parada, botón para cambiar a modo manual, slides para elegir el SetPoint, la frecuencia manual, el offset del sensor de presión, offset de voltaje, luces indicadoras, y una imagen que indica a grandes rasgos el nivel de agua actual del tanque.



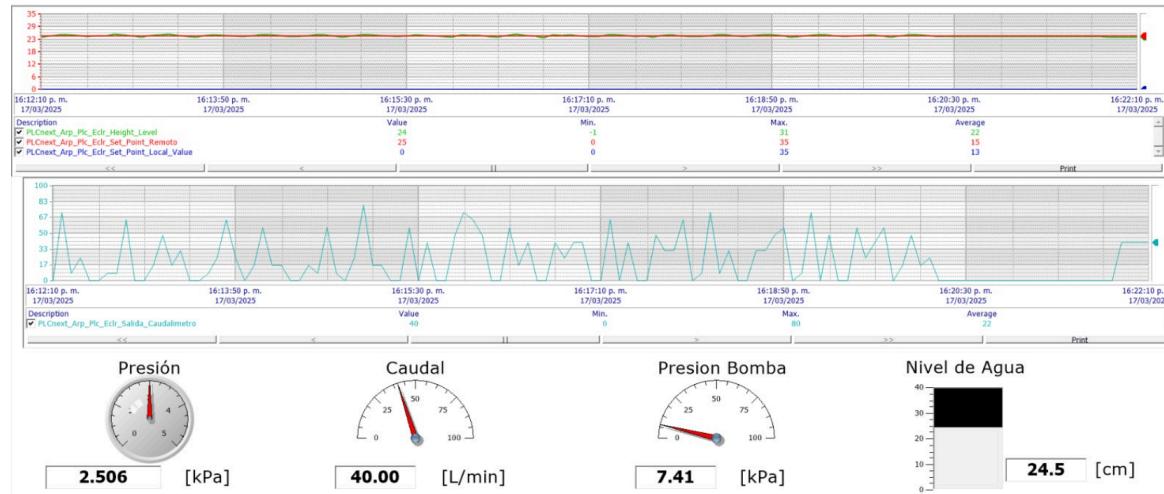
PID Controller

Desde esta ventana se podrán manipular los valores de las constantes de configuración del PID (KP, TI y TD) de modo remoto, como se muestra a continuación:

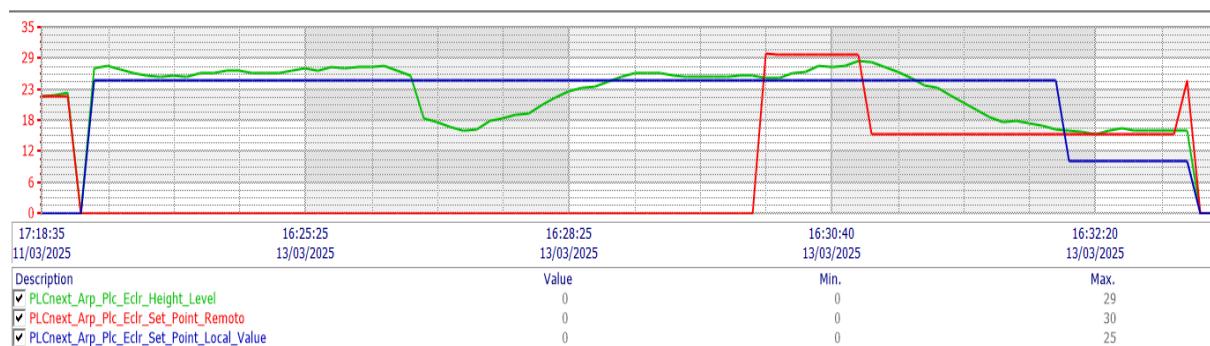


Lecturas

Es la ventana que contiene elementos de medición precisa de parámetros. Entre ellos podemos encontrar medidores de presión, de nivel, de caudal y gráficos que describen el proceso de llenado del tanque.

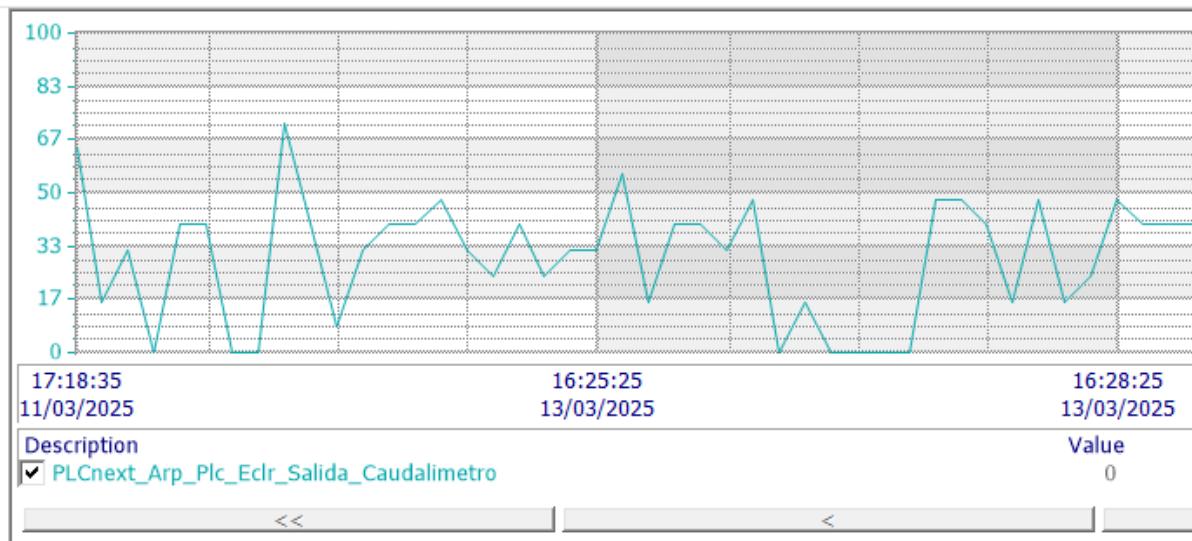


En el primer gráfico podemos ver en ROJO el SetPoint remoto, en AZUL el SetPoint local, y en VERDE el valor actual de la altura del tanque. Todo medido en centímetros. A continuación, se muestra un ejemplo de cómo interpretar este gráfico:



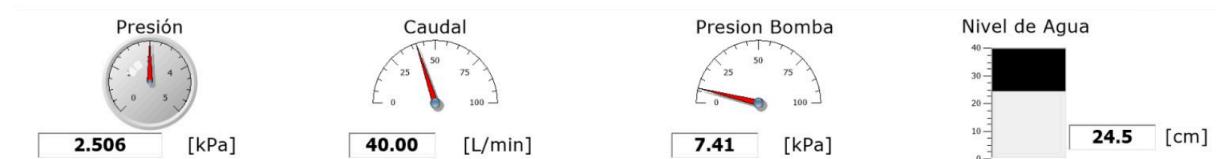
En este caso, comenzamos operando el PLC en modo local, por lo que la altura (VERDE) seguirá al SetPoint local (AZUL). Luego, en mitad de la gráfica, pasamos a manipular el sistema en modo remoto, por lo que la curva de altura seguirá a la curva del SetPoint remoto (ROJO), ignorando al SetPoint local.

El segundo gráfico indica el nivel de caudal a través de la tubería medido en litros/min.



Para valores altos de caudal, estas medidas pueden ser erróneas, debido a que el caudal máximo que puede proporcionar la bomba (130L/min) es mayor al rango de medición del sensor de caudal (30L/min).

Por último, debajo de esas gráficas tenemos medidores instantáneos de presión en tanque, altura en tanque y caudal.

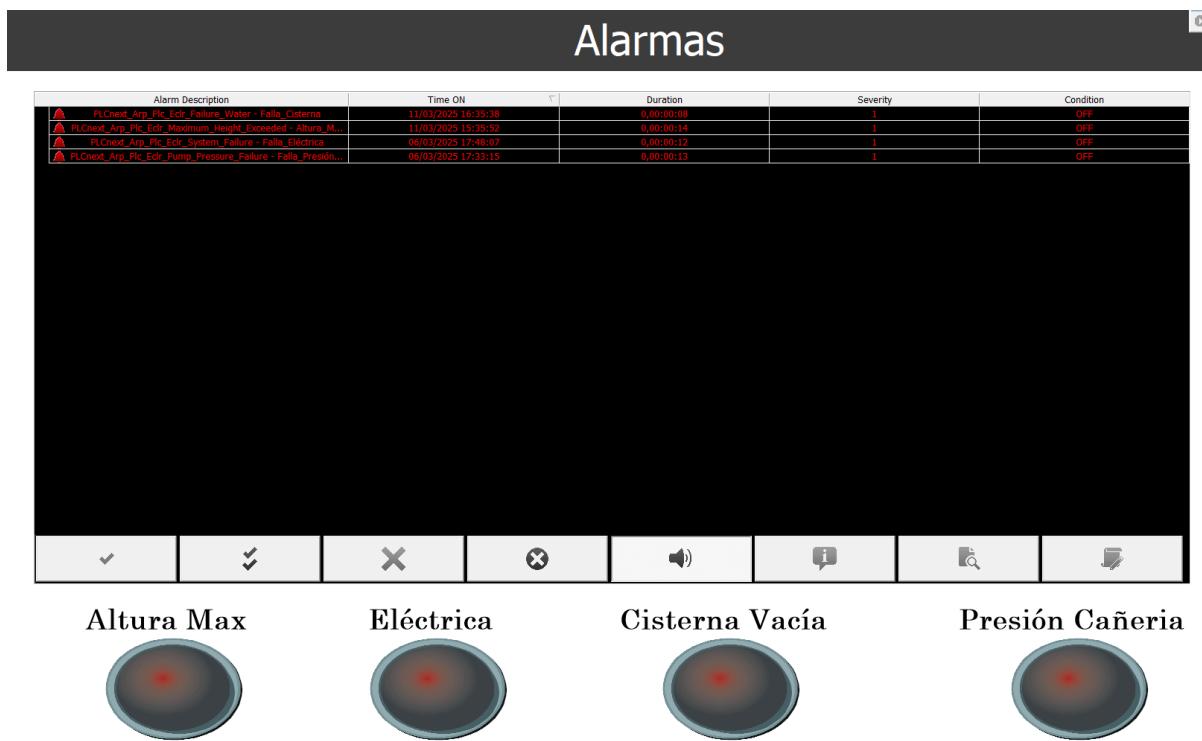


Alarmas

Como su nombre indica, esta ventana contendrá el registro de alarmas del sistema. En nuestro caso, como se explicó anteriormente, contamos con 4 alarmas:

- Alarma por exceso de altura (Maximum Height Exceeded)
- Alarma por ausencia de agua en cisterna (Failure Water)
- Alarma por falla eléctrica (System Failure)
- Alarma por obstrucción en tubería (Pump Pressure Failure)

Además de la pantalla de registro, contamos con cuatro luces intermitentes que nos indican cuál alarma se activó.



Haciendo zoom al registro:

Alarm Description	Time ON
PLCnext_Arp_Plz_Eclr_Failure_Water - Falla_Cisterna	11/03/2025 16:35:38
PLCnext_Arp_Plz_Eclr_Maximum_Height_Exceeded - Altura_M...	11/03/2025 15:35:52
PLCnext_Arp_Plz_Eclr_System_Failure - Falla_Eléctrica	06/03/2025 17:48:07
PLCnext_Arp_Plz_Eclr_Pump_Pressure_Failure - Falla_Presión...	06/03/2025 17:33:15

Registro de Eventos y Usuarios

Esta ventana tiene dos funciones: Por un lado, mantiene un registro de cada una de las alarmas anteriores, y por otro lado, indica al operador cuál usuario estaba logueado al momento de activarse la alarma. Debido a ciertas complicaciones, se tuvo que implementar esta función en dos ventanas separadas.

La ventana que mantiene el registro de los eventos es la siguiente:

Event Time	User	Failure_Water	Maximum_Height_Exceeded	Pump_Pressure_Failure	System_Failure
2025-03-11 15:36:06		false	false	false	false
2025-03-11 15:35:52		false	true	false	false
2025-03-11 15:35:46		false	false	false	false
2025-03-11 15:35:34		false	true	false	false
2025-03-11 15:31:01		false	false	false	false
2025-03-11 15:30:40		true	false	false	false
2025-03-11 15:30:39		false	false	false	false
2025-03-11 15:30:39		false	false	false	false
2025-03-11 15:30:39		false	false	false	false
2025-03-11 15:30:39		false	false	false	false
2025-03-11 15:30:39		false	false	false	false
2025-03-11 15:30:39		false	false	false	false
2025-03-11 15:30:39		false	false	false	false
2025-03-11 15:30:39		false	false	false	false
2025-03-11 15:30:39		false	false	false	false
2025-03-11 15:30:39		false	false	false	false
2025-03-11 15:28:17		false	false	false	false
2025-03-11 15:28:17		false	false	false	false
2025-03-11 15:28:17		false	false	false	false
2025-03-11 15:28:17		false	false	false	false
2025-03-06 17:48:19		false	false	false	false
2025-03-06 17:48:07		false	false	false	true
2025-03-06 17:47:50		false	false	false	false
2025-03-06 17:47:21		false	false	false	true
2025-03-06 17:47:16		false	false	false	false
2025-03-06 17:47:15		false	false	false	true

La ventana que mantiene el registro de los usuarios logueados es la siguiente:

Event Text	Event Time	User	Description
User 'UserGroup001\sup' has logged out	15/03/2025 15:22:16		
User 'UserGroup001\sup' has logged in	15/03/2025 15:20:19	UserGroup001\sup	
User 'UserGroup001\sup' has logged out	15/03/2025 15:18:21		
User 'UserGroup001\sup' has logged in	15/03/2025 15:17:18	UserGroup001\sup	
User 'UserGroup001\sup' has logged out	15/03/2025 15:15:09		
User 'UserGroup001\sup' has logged in	15/03/2025 15:13:57	UserGroup001\sup	
User 'UserGroup001\sup' has logged out	15/03/2025 15:13:12		
User 'UserGroup001\sup' has logged in	15/03/2025 15:11:41	UserGroup001\sup	
Booting up project : Last interaction was on 15/03/2025 15:...	15/03/2025 15:11:32		
User 'UserGroup001\sup' has logged out	15/03/2025 15:10:16		
User 'UserGroup001\sup' has logged in	15/03/2025 15:10:06	UserGroup001\sup	
Booting up project : Last interaction was on 13/03/2025 16:...	15/03/2025 15:09:56		
User 'UserGroup001\sup' has logged out	24/02/2025 9:55:58		
User 'UserGroup001\sup' has logged in	24/02/2025 9:55:43	UserGroup001\sup	
Booting up project : Last interaction was on 24/02/2025 9:4...	24/02/2025 9:55:30		
User 'UserGroup001\sup' has logged out	24/02/2025 9:44:57		
User 'UserGroup001\sup' has logged in	24/02/2025 9:44:52	UserGroup001\sup	
Booting up project : Last interaction was on 21/02/2025 12:...	24/02/2025 9:44:42		
User 'UserGroup001\sup' has logged out	21/02/2025 12:32:17		
User 'UserGroup001\sup' has logged in	21/02/2025 12:31:51	UserGroup001\sup	
Booting up project : Last interaction was on 21/02/2025 12:...	21/02/2025 12:31:42		
User 'UserGroup001\sup' has logged out	21/02/2025 12:31:17		
User 'UserGroup001\sup' has logged in	21/02/2025 12:31:02	UserGroup001\sup	
Booting up project : Last interaction was on 21/02/2025 12:...	21/02/2025 12:30:53		
User 'UserGroup001\sup' has logged out	21/02/2025 12:29:51		
User 'UserGroup001\sup' has logged in	21/02/2025 12:29:46	UserGroup001\sup	
Booting up project : Last interaction was on 21/02/2025 12:...	21/02/2025 12:29:30		
User 'UserGroup001\sup' has logged out	21/02/2025 12:29:11		
User 'UserGroup001\sup' has logged in	21/02/2025 12:28:31	UserGroup001\sup	
Booting up project : Last interaction was on 21/02/2025 12:...	21/02/2025 12:28:22		
User 'UserGroup001\sup' has logged out	21/02/2025 12:26:49		
User 'UserGroup001\sup' has logged in	21/02/2025 12:26:28	UserGroup001\sup	
Booting up project : Last interaction was on 21/02/2025 12:...	21/02/2025 12:26:17		
User 'UserGroup001\sup' has logged out	21/02/2025 12:22:03		
User 'UserGroup001\sup' has logged in	21/02/2025 12:21:31	UserGroup001\sup	
Booting up project : Last interaction was on 21/02/2025 12:...	21/02/2025 12:21:21		
Booting up project : Last interaction was on 21/02/2025 12:...	21/02/2025 12:19:31		
User 'UserGroup001\sup' has logged out	21/02/2025 12:15:50		
User 'UserGroup001\sup' has logged in	21/02/2025 12:15:28	UserGroup001\sup	
ller 'UserGroup001\sup' has logged out	21/02/2025 12:15:20		

Event Text	Event Time	User
User 'UserGroup0001\sup' has logged out	15/03/2025 15:22:16	
User 'UserGroup0001\sup' has logged in	15/03/2025 15:20:19	UserGroup0001\sup
User 'UserGroup0001\sup' has logged out	15/03/2025 15:18:21	
User 'UserGroup0001\sup' has logged in	15/03/2025 15:17:18	UserGroup0001\sup
User 'UserGroup0001\sup' has logged out	15/03/2025 15:15:09	
User 'UserGroup0001\sup' has logged in	15/03/2025 15:13:57	UserGroup0001\sup
User 'UserGroup0001\sup' has logged out	15/03/2025 15:13:12	
User 'UserGroup0001\sup' has logged in	15/03/2025 15:11:41	UserGroup0001\sup
Booting up project : Last interaction was on 15/03/2025 15:... User 'UserGroup0001\sup' has logged out	15/03/2025 15:11:32	

Usuarios

Por seguridad, para interactuar con la interfaz se han creado múltiples usuarios con distintos niveles de acceso para operar el sistema. En nuestro caso tenemos tres:

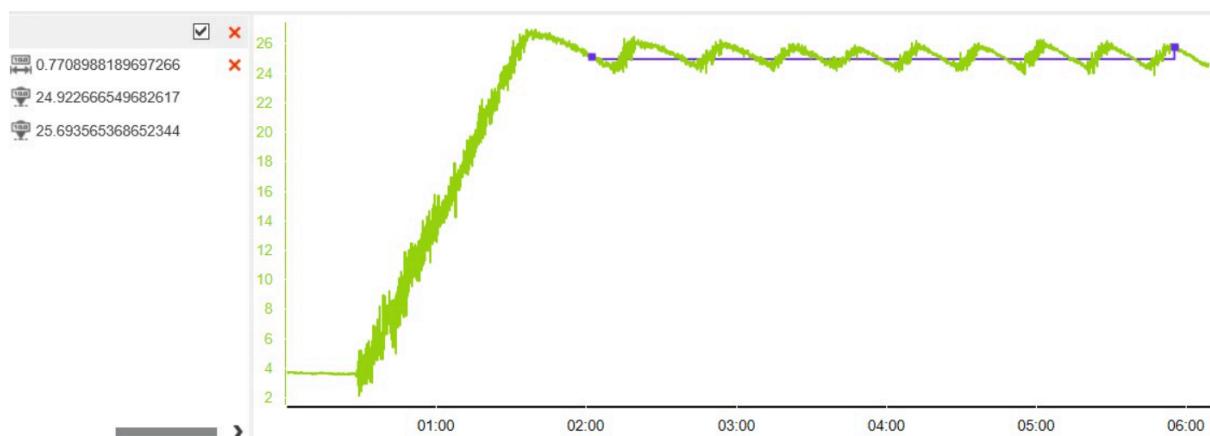
- “Supervisor”: Es el usuario con mayor nivel de acceso. Puede interactuar con todos los elementos de todas las ventanas, es decir, puede dar marcha y parada al proceso, cambiar entre modo manual y automático, elegir el SetPoint, velocidad del variador y los offset, ver todas las gráficas y registros, etc.
- “Operador”: Este usuario puede hacer casi todo lo que hace el supervisor, con la diferencia de que solo podrá operar en modo automático, sin posibilidades de utilizar el modo manual. También tendrá prohibido el acceso a cambiar los offset del sensor de presión y de voltaje.
- “Observador”: Este usuario podrá ver todos los elementos de todas las ventanas, pero no podrá interactuar con ninguno. Es el que menos nivel de acceso tiene.

Ensayos realizados

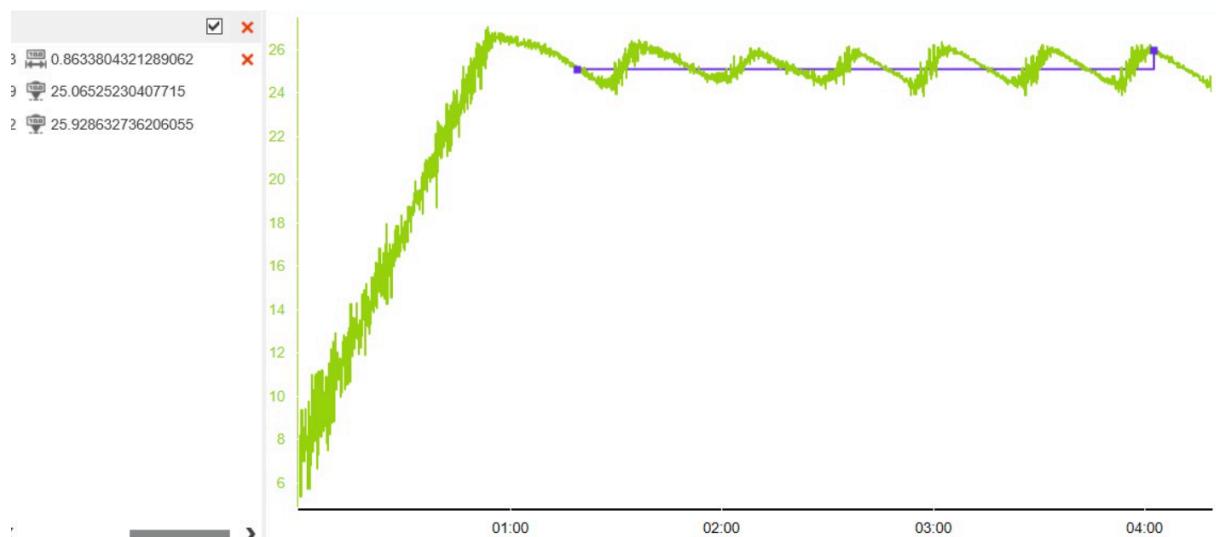
Se llevaron a cabo ensayos en laboratorio con el objetivo de ajustar los parámetros del controlador PID y minimizar las fluctuaciones en la respuesta del sistema. Estos ensayos se realizaron con un SetPoint de 25 centímetros y utilizando la herramienta “Logic Analyzer” provista por el software PLCNext Engineer, en donde podremos observar cómo varía el nivel del agua en función del tiempo.

Algunos de estos ensayos fueron los siguientes:

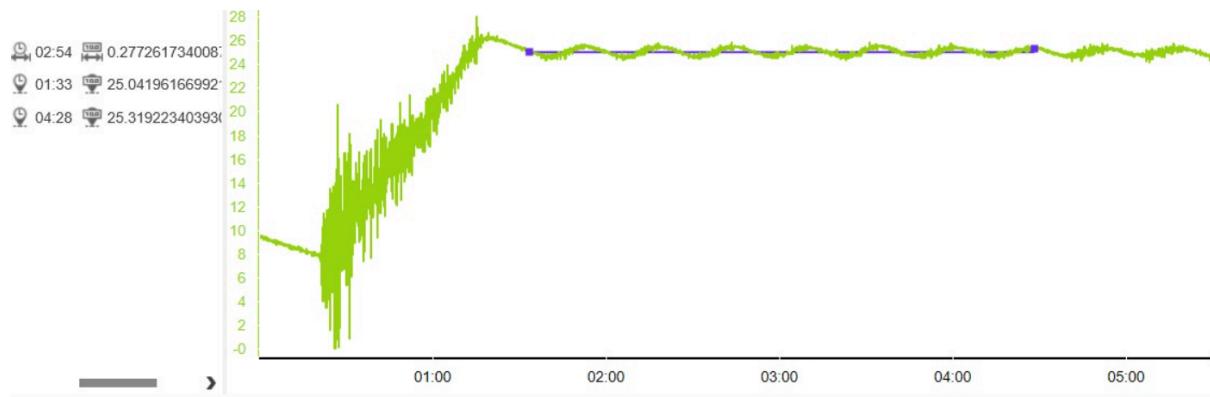
$KP = 35.0, TI = 0.01, TD = 3.0, \text{Offset} = -1.5$



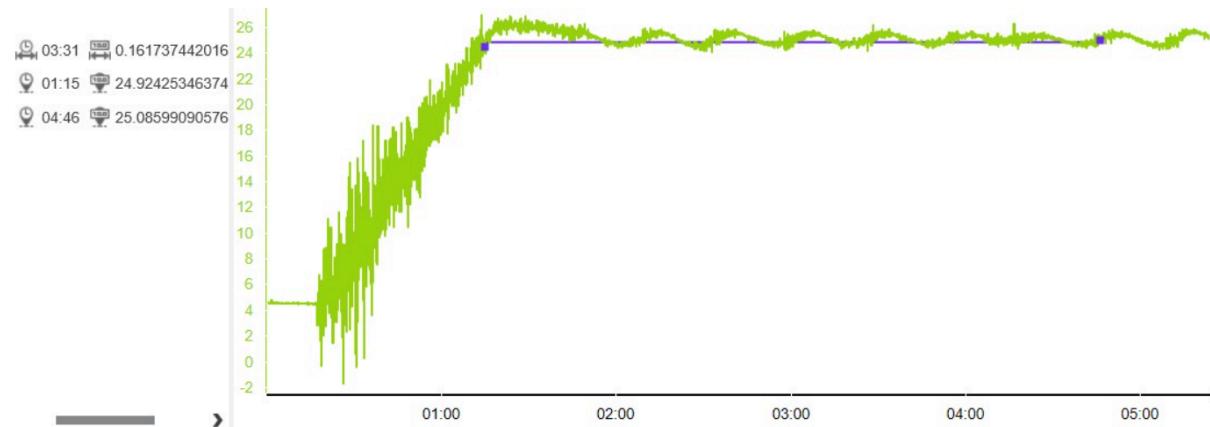
$KP = 35.0, TI = 0.01, TD = 5.0, \text{Offset} = -1.5$



KP = 35.0, TI = 0.02, TD = 10.0, Offset = -3.0



KP = 35.0, TI = 0.02, TD = 12.0, Offset = -3.0

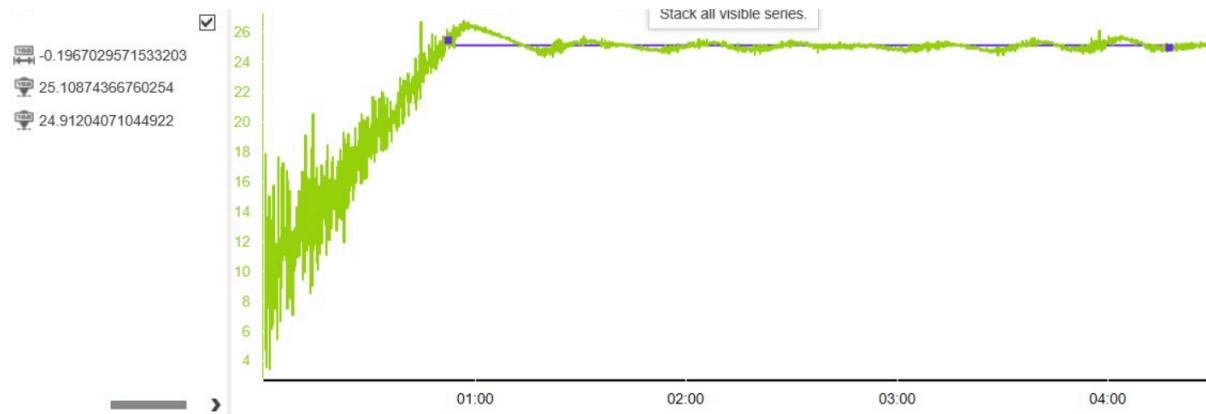


Mediante otros ensayos se determinó que para un TI superior a 1.0, la oscilación era superior al límite de altura máxima, por lo que el sistema se detenía. Esto nos indicó que el TI debía ser pequeño, del orden de las centésimas.

Como resultado final, se determinaron los siguientes valores óptimos para los parámetros del controlador:

- Ganancia proporcional (Kp): 35
- Tiempo integral (Ti): 0.02
- Tiempo derivativo (Td): 15.0
- Offset: -1.5

Estos valores fueron seleccionados por presentar el comportamiento más estable durante las pruebas, logrando reducir al mínimo las oscilaciones del sistema como se ve en la siguiente imagen.



Observaciones y Complicaciones

El problema más notorio que tuvimos durante este proyecto fue la dificultad para medir con precisión la altura del tanque, puesto que las fluctuaciones de presión eran considerables. Esto puede deberse a que el tanque que operamos es pequeño en comparación a los que se utilizan en las industrias, lo que implica que el sensor trabaja en la parte más baja de su escala, lo que disminuye la precisión de la medida.

Una posible solución que aplicamos fué determinar la lectura de la presión mediante el promedio de varias muestras. Esto redujo considerablemente el ruido, sin embargo, el error continuó presente.

A su vez, se notó que el sensor de presión presentaba problemas a la hora de medir cuando se comenzaba a utilizarlo con el tanque vacío. Esto no supuso un problema serio, ya que se solucionaba calibrando el sensor por unos minutos y esperando que la medida se estabilice.

Otro problema que se nos presentó fue la dificultad de sincronizar nuestro trabajo entre diferentes computadoras, teniendo que trabajar cada miembro del grupo con copias independientes. Esto derivó en un leve retraso en el desarrollo del proyecto.

Conclusiones

A lo largo del desarrollo de este proyecto aprendimos a manipular un PLC de la marca Phoenix Contact, muy utilizado en las industrias actuales. Adquirimos destrezas en el manejo de programas como PLCNext Engineer y Visu+, sintonización de controladores PID y creación de interfaz SCADA.

En términos generales, podemos decir que el proyecto funcionó correctamente, aunque con algunas imperfecciones debido a los problemas anteriormente mencionados.