



**UNIVERSIDAD NACIONAL
PEDRO RUIZ GALLO**



FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICA

**ESCUELA PROFESIONAL DE INGENIERÍA
ELECTRÓNICA**

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA SCADA
CON PLC SIEMENS S7-200 PARA EL CONTROL PID DE
LA PLANTA DE NIVEL DE LA ESCUELA PROFESIONAL
DE INGENIERÍA ELECTRÓNICA DE LA UNIVERSIDAD
NACIONAL PEDRO RUIZ GALLO, UTILIZANDO
SOFTWARE LIBRE”.**

TESIS

**PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO ELECTRÓNICO**

PRESENTADA POR:

**BACH. CARLOS ENRIQUE PADILLA JULCA
BACH. GERSON ENRIQUE TORO YDROGO**

ASESOR:

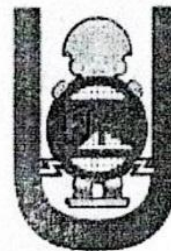
Ing. OSCAR ROMERO CORTEZ

**LAMBAYEQUE- PERÚ
2018**



UNIVERSIDAD NACIONAL

PEDRO RUIZ GALLO



FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICA

**ESCUELA PROFESIONAL DE INGENIERÍA
ELECTRÓNICA**

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA SCADA
CON PLC SIEMENS S7-200 PARA EL CONTROL PID DE
LA PLANTA DE NIVEL DE LA ESCUELA PROFESIONAL
DE INGENIERÍA ELECTRÓNICA DE LA UNIVERSIDAD
NACIONAL PEDRO RUIZ GALLO, UTILIZANDO
SOFTWARE LIBRE”.**

TESIS

**PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO ELECTRÓNICO**

PRESENTADA POR:

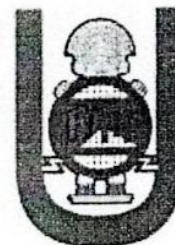
BACH. CARLOS ENRIQUE PADILLA JULCA
AUTOR

BACH. GERSON ENRIQUE TORO YDROGO
AUTOR

Ing. OSCAR ROMERO CORTEZ
ASESOR



**UNIVERSIDAD NACIONAL
PEDRO RUIZ GALLO**



FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICA

**ESCUELA PROFESIONAL DE INGENIERÍA
ELECTRÓNICA**

**“DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA SCADA
CON PLC SIEMENS S7-200 PARA EL CONTROL PID DE
LA PLANTA DE NIVEL DE LA ESCUELA PROFESIONAL
DE INGENIERÍA ELECTRÓNICA DE LA UNIVERSIDAD
NACIONAL PEDRO RUIZ GALLO, UTILIZANDO
SOFTWARE LIBRE”.**

TESIS

**PARA OPTAR EL TÍTULO PROFESIONAL DE
INGENIERO ELECTRÓNICO**

APROBADO POR LOS MIEMBROS DEL JURADO:

**Ing. MANUEL JAVIER RAMÍREZ CASTRO
PRESIDENTE**

**Ing. CARLOS LEONARDO OBLITAS VERA
SECRETARIO**

**Ing. MARTÍN NOMBERRA LOSSIO
VOCAL**

DEDICATORIA

A mis padres **Efrain y Orfelinda**, quienes con su esfuerzo están haciendo de mí una persona de bien y me motivan a ser mejor cada día.

CARLOS PADILLA

A mis padres Enrique y Maribel, por las arduas jornadas de trabajo y el esfuerzo incansable que realizan día a día con el objetivo de ver a sus hijos convertidos en profesionales.

GERSON TORO YDROGO

AGRADECIMIENTO

Agradecer a nuestros familiares quienes con su apoyo nos ayudan a ser mejor cada día.

Queremos expresar nuestro sincero agradecimiento a la memoria del **Ing. Víctor Jara Sandoval**, quien nos apoyó inicialmente con el planteamiento del proyecto de tesis.

Al ingeniero **Marlon Vega Zulueta** quien nos solucionaba algunas dudas con sus conocimientos de **PLC y SCADA**.

Al Ing. **Oscar Romero Cortez** quien aceptó ser nuestro asesor y nos apoyó arduamente para poder culminar el proyecto dejado por el **Ing. Víctor Jara**.

LOS AUTORES

RESUMEN

Se desarrolló un Sistema SCADA con software libre para la planta de nivel del laboratorio de ingeniería electrónica, la cual está gobernada por un PLC S7-200, el nivel del tanque se obtiene mediante un sensor ultrasónico.

El proyecto está desarrollado en dos partes la primera consiste en hacer un reconocimiento general de la planta de nivel, dispositivos de accionamiento y protección, sensores y actuadores, así como los elementos de control.

Luego se procede a realizar un programa en el PLC S7-200 para posteriormente conectarlo con LabView y obtener los datos del comportamiento de la planta, los datos obtenidos son procesados mediante MATLAB y obtenemos la función de transferencia, mediante un software de simulación le implementamos los controles P, PI, PID.

Luego en una segunda parte estos controladores son implementados en la planta real, mediante el software de programación de Siemens se pudo verificar el comportamiento mediante la librería PID.

Una vez que se logró sintonizar los controladores, se desarrolló un SCADA para poder visualizar las variables de proceso de forma más amigable con el operador, se probó varios software SCADA pero al final se optó por el IGSS ya que reunía los requisitos necesarios para ser implementados en aplicaciones de pequeña y media empresa por que cuenta con diversas formas de comunicación con PLC de diferentes marcas.

Se logró nuestro objetivo principal que fue el diseño de un sistema SCADA con software libre para la planta de nivel la cual podemos controlar y monitorear los diferentes parámetros como: nivel del tanque, velocidad de variador de frecuencia, así como también distintas alarmas implementadas en la planta.

ABSTRACT

A SCADA system with free software was developed for the level plant of the electronic engineering laboratory, which is governed by an S7-200 PLC, the level of the tank is obtained by means of an ultrasonic sensor.

The project is developed in two parts, the first consists of making a general recognition of the level plant, actuation and protection devices, sensors and actuators, control elements.

Then proceed to perform a program in the S7-200 PLC to later connect it with LabView and obtain the data of the plant behavior, the data obtained are processed by MATLAB and we obtain the transfer function, using simulation software we implement the controls P, PI, PID.

Then in a second part these controllers are implemented in the real plant, through the programming software of Siemens the behavior could be verified by the PID library.

Once the controllers were tuned, a SCADA was developed to be able to visualize the process variables in a more friendly way with the operator. Several SCADA software was tested by IGSS, since it met the necessary requirements to be implemented in small applications and half company because it has different forms of communication with PLC of different brands.

Our main objective was to design a SCADA system with free software for the level plant which we can control and monitor the different parameters such as tank level, frequency inverter speed, different alarms implemented in the plant.

INDICE

LISTA DE FIGURAS	XV
LISTA DE TABLAS	XXI
CAPÍTULO 1.	
INTRODUCCIÓN.....	1
1.1. Situación Problemática.....	1
1.2. Justificación del trabajo de tesis	1
1.3. Objetivos del trabajo de tesis	2
1.3.1.Objetivo principal	2
1.3.2.Objetivos secundarios.....	2
CAPÍTULO 2	
GENERALIDADES.....	3
2.1. Planta de nivel	3
2.2. Elementos que componen la planta de nivel	4
2.2.1. Sensor ultrasónico TOUGH SONIC 14 modelo: ts-30s1-IV..	4
2.2.2. Válvula Proporcional.....	5
2.2.3. Electrobomba.....	6
2.2.4. Válvulas de bola y tanque	7
2.2.5. Tablero de control	8
2.3. El controlador lógico programable S7-200	12
a) CPU S7200.....	12
b) Módulos de ampliación s7-200	14
2.4. Software de programación STEP 7-Micro/WIN	15
2.4.1. Requisitos del sistema.....	15
2.4.2. Opciones de comunicación	16
2.4.3. Configuración de STEP 7-Micro/WIN	16
a) Iniciar STEP 7-Micro/WIN	16
b) Verificar los parámetros de comunicación de STEP 7-Micro/WIN..	17
c) Establecer la comunicación con el S7-200	18
d) Crear programa	19
e) Cargar un programa.....	20
f) Poner el S7-200 en modo RUN	21
2.5. Software para el cálculo de la función de transferencia	22
2.5.1. NI LabView.....	22
2.5.2. Matlab	23
2.6. Software KEPServer v5	24

2.7. Comunicación entre KEPServer Y S7-200 CPU 224	25
2.7.1.Arquitectura del sistema.....	25
2.7.2.Descripción del sistema PLC	26
2.7.3 Conector RS-485 de la tarjeta moxa Uport 1150	27
2.7.4 Configuración de KEPServer para comunicación del PLC-S7 200 con HMI.....	28

CAPÍTULO 3

OBTENCIÓN DEL MODELO MATEMÁTICO DE LA PLANTA Y

SIMLUACIÓN.....30

3.1. Adquisición de datos	30
3.1.1. Programación en MICROWIN STEP-7	30
3.1.2. Programación en LABVIEW	32
3.2. Procesamiento de los datos obtenidos en LABVIEW	34
3.3. Estimación de la función de transferencia	35
3.3.1. Función de transferencia obtenido	37
3.4. Polos y ceros de la función de transferencia	38
3.5. Estimación del modelo de salida	39
3.6. Diseño del controlador	40
3.6.1.Comando PIDTOOL.....	40
▪ Controlador Proporcional (P)	41
▪ Control Proporcional Integral (PI)	42
▪ Control Proporcional Integral Derivativo (PID)	43
3.7. Simulación del modelo de la planta utilizando SIMULINK	44
3.7.1.Respuesta Control Proporcional	45
3.7.2.Respuesta Control Proporcional Integral (PI)	46
3.7.3. Respuesta al control Proporcional Integral Derivativo (PID)	47
3.8. Simulación del controlador utilizando LABVIEW	48

CAPÍTULO 4

DISEÑO DEL SISTEMA SCADA 49

4.1 SCADA IGSS FREE.....	49
4.1.1 Alcance del software	50
4.1.2 Conexión OPC	50
4.1.3 Entorno gráfico de SCADA IGSS	53

CAPÍTULO 5

IMPLEMENTACIÓN DE CONTROLADORES P, PI, PID 58

5.1. Programa final en PLC S7-20058

5.2. Creación de las variables en el OPC 63

5.3. Programación en LabView de los controladores 64

5.4. Creación de las variables en el SCADA libre 66

CAPÍTULO 6

RESULTADOS EXPERIMENTALES70

6.1 Respuesta al Control Proporcional (P)70

6.2 Respuesta al Control Proporcional Integral (PI) 72

6.3 Respuesta al Control Proporcional Integral Derivativo (PID)74

CONCLUSIONES 75

RECOMENDACIONES.....76

BIBLIOGRAFÍA..... 77

ANEXO A.....78

ANEXO B.....84

ANEXO C.....85

INDICE DE FIGURAS

2.1. Diagrama de bloques para el sistema de control de nivel de un tanque.....	3
2.2. Sensor de nivel tipo ultrasónico instalado en la planta de nivel.....	4
2.3. Electroválvula marca BELIMO.....	5
2.4. Imagen de electrobomba.....	6
2.5. Válvulas de bola y tanque.....	7
2.6. Dispositivos de accionamiento y protección.....	8
2.7. Imagen del variador de frecuencia	10
2.8. Imagen del PLC S7-200	11
2.9. Partes del CPU S7-200	12
2.10. Imagen de entorno de programación STEP7.....	15
2.11. Entorno de programación STEP7.....	16
2.12. Imagen de parámetros de comunicación de STEP 7-Micro/WIN	17
2.13. Establecer la comunicación con el S7-200	18
2.14. Subrutina en LADDER	19
2.15. Cargar un programa	20
2.16. Damos nombre a nuestro proyecto (MI PLC)	28
2.17. Seleccionamos los modelos de PLC al que deseamos conectar (S7-200).....	28
2.18. Seleccionamos el puerto COM	29
2.19. Ventana donde se verifica la comunicación PC-PLC	29
3.1. Variables empleadas en el PLC	30
3.2. Ladder utilizado para adquirir función de transferencia	31
3.3. Programación en LabView para adquirir función de transferencia	33
3.4. Diagrama de bloques de la programación en LabView.....	33
3.5. Comportamiento del sensor al manipular la frecuencia del variador	34
3.6. Datos de la planta cargados en Workspace	35
3.7. Ventana principal de System Identification	36
3.8. Datos cargados en System Identification	36
3.9. Imagen donde se muestra la aproximación de la señal	37
3.10. Polos y ceros de la función de transferencia de la planta de nivel	38
3.11. Aproximación del modelo de salida.....	39
3.12. Ventana donde se manipula <i>Response time</i> y <i>Transient Behavior</i>	40
3.13. Tabla de valores obtenidos para el controlador P	41
3.14. Polos y ceros del controlador Proporcional	41
3.15. Tabla de valores para el controlado Proporcional Integral	42
3.16. Polos y ceros de la función de transferencia al aplicar control PI.....	42

3.17. Tabla de valores obtenidos al aplicar control PID a la función de transferencia de la planta de nivel.....	43
3.18. Polos y ceros de función de transferencia al aplicar control PID.....	43
3.19. Diagrama de bloques para control P, PI, PID de la planta de nivel.....	44
3.20. Respuesta de la planta a un control Proporcional.....	45
3.21. Respuesta de la planta de nivel al control Proporcional Integral.....	46
3.22. Respuesta de la planta de nivel al controlador PID.....	47
3.23. Panel principal de la simulación del sistema en LabView.....	48
3.24. Panel de la programación en diagrama de bloques del sistema	48
4.1. Configuración de nuevo proyecto en IGSS	50
4.2. Ventana donde podemos seleccionar el driver para comunicación con cualquier PLC	51
4.3. Driver seleccionado para la comunicación OPC	51
4.4. Ventana donde seleccionamos el servidor OPC para leer y escribir los datos al PLC	52
4.5. Ventana principal de software IGSS	53
4.6. Muestra que la comunicación con PLC será mediante driver OPC	53
4.7. Muestra la ventana principal del panel de instrumentos, donde se configura los tags y animaciones	54
4.8. Muestra los servidores OPC instalados en la PC que están disponibles para utilizar con el software IGSS SCADA	54
4.9. Se creó un tag en el panel de diseño, al cambiar IGSS a modo Run ya puede ser monitoreado.....	55
4.10. Muestra el diagrama de tuberías para diseñar procesos industriales.....	55
4.11. Se muestra panel de instrumentos la siguiente animación se utilizara para mostrar el estado del variador de frecuencia	56
4.12. Se muestra los distintos elementos que se puede seleccionar para nuestro diseño tanques, válvulas, motores, etc.....	56
4.1. Se muestra algunos elementos que se utilizaran para el proyecto de PID para control de nivel.....	57
5.1. Variables utilizadas para la implementación de los controladores	58
5.2. Programación final en Ladder.....	62
5.3. Se muestra los tags que fueron configurados para leer y escribir en el PLC.....	63
5.4. Panel frontal de programación final en LabView del SCADA	65
5.5. Diagrama de bloques del SCADA en LabView	65
5.6. Conexión del variable sensor entre OPC Y IGSS	66

5.7. Conexión de la variable Variador	66
5.8. Conexión de los tags SET_POINT, KP, KI, KD.....	67
5.9. Visualizador circular	67
5.10. Gráfico de históricos	68
5.11. Tags de marcha y paro	68
5.12. Imagen final del SCADA libre desarrollado en IGSS.....	69
6.1. Se puede observar un control proporcional.....	70
6.2. Respuesta de la planta de nivel a $K_p=0.6$	71
6.3. Control PI con $K_p=0.6$ y $K_i=0.1$	72
6.4. Control PI con $K_p=1$ y $K_i=0.4$	73
6.5. Respuesta de la planta al control PID.....	74

.LISTA DE TABLAS

1.1. Características de módulos CPU S7-200.....	13
1.2. Módulos de ampliación del S7-200.....	14
2.1. Configuración de los pines para comunicación RS-485.....	26
2.2. Conexión entre moxa RS-485 y RS-485 profibus de siemen	27
3.1. Variables utilizadas en el PLC S7-200	32

CAPÍTULO 1

INTRODUCCIÓN

1.1. Situación problemática

En la actualidad laboratorio de la Escuela Profesional de Ingeniería Electrónica cuenta con diferentes módulos y equipos para realizar distintas prácticas de control y automatización.

Sin embargo, estos no son aprovechados al máximo debido a que el software instalado en las plantas tanto de nivel y presión fueron dañados durante su uso y reponer estos tendría un costo muy elevado, la escuela no cuenta con los recursos suficientes para reponerlo, esto limita a los alumnos para desarrollar aplicaciones complejas de control, por el momento solo se trabaja en pequeñas aplicaciones poco utilizadas en plantas industriales.

Es por esto que se propone realizar una aplicación con software libre, que cumpla las funciones de un SCADA que realice control y supervisión de procesos, en este caso de nivel, para que los estudiantes puedan adquirir destreza en sistemas SCADA, manipular las variables de proceso en la planta de nivel, con esto ahorramos el costo de adquirir un software de National Instrument para reponer la pantalla MHI o utilizar NI Labview que es software con licencia.

Es así como surge la propuesta del diseño e implementación de un sistema SCADA con software libre que trabaja con el PLC S7-200 para la supervisión y monitoreo de la planta de nivel de nuestro laboratorio.

1.2. Justificación del trabajo de tesis.

En el control de los procesos industriales, no podemos conocer exactamente el valor de las múltiples variables que intervienen ya sea temperatura, presión, nivel, caudal, etc. debido a que los sensores y actuadores se encuentra dispersos en los diferentes lazos de control que existen en los procesos y paneles de control.

Entonces es necesario la integración del proceso a lo cual llamamos SCADA, pero estos sistemas son muy costosos para la pequeña y mediana empresa los cuales no cuentan con los recursos suficientes para poder adquirirlos.

Existen sistemas para controlar y monitorear los procesos que son desarrollados con lenguajes de código abierto que están disponibles para ser acondicionados a los procesos industriales, y beneficiar en costos y tener un control más preciso, mejorando la calidad del producto final.

Es así como nace la idea de implementar un sistema SCADA con software libre en nuestro caso para ser implementado en la planta de nivel de laboratorio de ingeniería electrónica.

1.3. Objetivo del trabajo de tesis.

1.3.1. Objetivo principal.

Implementar un sistema SCADA para el control y supervisión de la planta de nivel, desarrollando interface gráfica con software libre.

1.3.2. Objetivos secundarios.

Los objetivos secundarios son los siguientes:

- Diseñar controladores P, PI, PID para la planta de nivel con el fin de ver la respuesta en el tiempo y la frecuencia de diferentes variables.
- Interpretar y comparar los diferentes tipos de control P, PI, PID.
- Utilizar servidor OPC estándar en comunicaciones industriales.
- Implementar un sistema SCADA con software libre para controlar y monitorear la planta de nivel.

CAPÍTULO 2

GENERALIDADES

2.1 Planta de nivel:

La planta de nivel del laboratorio de Ingeniería Electrónica de la Universidad Nacional Pedro Ruiz Gallo está compuesta por diferentes dispositivos electrónicos-electromecánicos interconectados entre sí.

Está compuesto por un PLC s7-200 el cual gobierna el proceso, un sensor de nivel tipo ultrasónico que está ubicado en la parte superior de un tanque, este envía señales de corriente 4-20mA al controlador el cual de acuerdo a las instrucciones del programador puede activar una salida analógica conectada a un variador de frecuencia que controla a una electrobomba de 0.5Kw, de esta manera podemos realizar algunas estrategias de control en lazo abierto y lazo cerrado.

En la **figura 1** se puede observar un diagrama de bloques para un sistema de control en lazo cerrado.

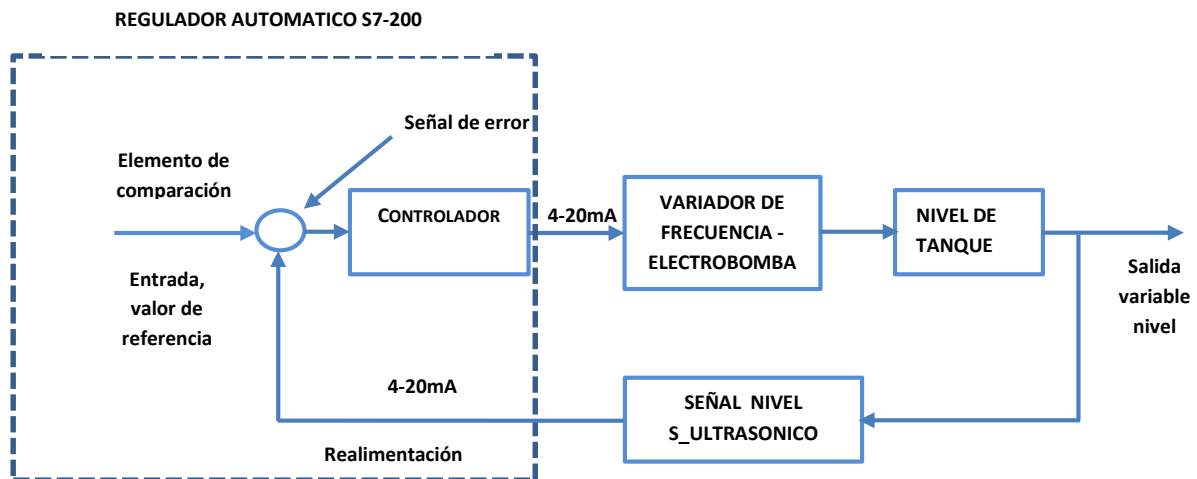


Figura 2.1. Diagrama de bloques para el sistema de control de nivel de un tanque.

2.2 Elementos que componen la planta de nivel.

2.2.1 Sensor ultrasónico TOUGH SONIC 14 modelo: ts-30s1-IV

El sensor ultrasónico ToughSonic 14 (antes TSPC-30S1) tiene un alcance máximo de 14 pies (4.3 metros). Está construido para una duración prolongada y fácil de usar en entornos industriales difíciles. Es totalmente sumergible y resistente a la corrosión.

Este sensor compacto se utiliza en todo, como por ejemplo en sistemas de medición nivel de líquidos, control de movimiento, aplicaciones de automatización de fábrica y más.

Cuenta con protección IP68 para aplicaciones al aire libre.

Su principio de funcionamiento es medir la distancia mediante una onda ultrasónica, el cabezal emite una onda ultrasónica y recibe la onda reflejada desde la base del tanque ó donde se encuentra el nivel del líquido, mide la distancia al objeto (nivel del líquido) contando el tiempo entre la emisión y la recepción.



Figura 2.2. Sensor de nivel tipo ultrasónico instalado en la planta de nivel.

2.2.2 Válvula Proporcional

Marca *Belimo*, modelo LRB24-SR, válvula accionada por solenoide de 2 vías, mediante la regulación de corriente en la bobina abre y cierra la válvula, esta se abre por completo cuando la corriente de la bobina es máxima 20mA y se cierra por completo cuando no existe corriente en la bobina 4mA. Es una electroválvula alimentada por 24Vdc.

En la figura 2.3 se observa la electroválvula instalada en la parte inferior del tanque de la planta de nivel, nos sirve para realizar distorsión a los controladores P, PI, PID.



Figura 2.3. Electroválvula marca BELIMO.

2.2.3 Electrobomba

Cuenta con una electrobomba de Jaula de ardilla marca Pentax, puede funcionar con 220 VAC o 380 VAC.

Este actuador será el encargado de llevar el agua hasta el proceso (tanque) de acuerdo a las señales emitidas por el PLC s7-200.



Figura 2.4. Imagen de la Electrobomba.

2.2.4 Válvulas de bola y tanque

Las válvulas de bola son de $\frac{1}{4}$ de vuelta en las cuales la bola taladrada gira entre asientos elásticos, lo cual permite la circulación directa en la posición abierta y corta el paso cuando se gira la bola 90° y cierra el conducto.

Estas se encuentran distribuidas en la planta para simular señales de perturbación y mantenimiento de la misma, un tanque de altura 60 cm que simula el proceso a controlar.



Figura 2.5. Vista panorámica del tanque y llaves de bola.

2.2.5 Tablero de control

❖ Parte Externa:

El tablero de control en su exterior tiene 4 Lámparas indicadoras de colores ámbar, verde y 02 rojos. También cuenta con 02 botones, uno verde con la etiqueta START y uno rojo con la etiqueta STOP, un potenciómetro, un selector de tres posiciones y una botón de paro de emergencia, una pantalla HMI modelo TPC-2106 de National Instruments.

❖ Parte Interna

En el interior encontramos:

Dispositivos de accionamiento y protección.

- a) **Llave termo magnético bipolar:** Para accionar el circuito de mando.
- b) **Llave termo magnético trifásico:** Para energizar el circuito de fuerza.
- c) **Interruptor termo magnético diferencial:** Dispositivo electromecánico que protege a las personas de contacto directo con partes activas de la planta.
- d) **Guarda motor magnético:** Los guarda motores magnéticos cumplen la función de protección contra cortocircuitos, cumpliendo adicionalmente la función de seccionamiento. Los requisitos para que cumplan con la función de protección contra cortocircuito son básicamente una pronta detección de la corriente de defecto y una rápida apertura de los contactos.



Figura 2.6. Dispositivos de protección.

e) Variador de frecuencia ALTIVAR 31

APLICACIONES:

El variador Altivar 31 es un convertidor de frecuencia para motores asíncronos trifásicos de jaula. Es resistente, de dimensiones reducidas, fácil de instalar.

Incluye funciones que se ajustan a las aplicaciones más usuales, en particular:

- ✓ Manutención (cintas transportadoras pequeñas, polipastos...).
- ✓ Máquinas de envase y embalaje.
- ✓ Máquinas especiales (mezcladores, trituradores, maquinaria textil...).
- ✓ Bombas, compresores, ventiladores.

Los variadores Altivar 31 se comunican con los buses industriales Modbus y CANopen. Estos dos protocolos se integran de fábrica en el variador.

Los variadores Altivar 31 se suministran con un radiador para entornos normales y envolventes ventilados. El montaje yuxtapuesto permite ahorrar gran cantidad de espacio.

Los variadores están diseñados para potencias de motor comprendidas entre 0,18 kW y 15 kW con cuatro tipos de alimentación:

- ✓ De 200 V a 240 V monofásica, de 0,18 kW a 2,2 kW.
- ✓ De 200 V a 240 V trifásica, de 0,18 kW a 15 kW.
- ✓ De 380 V a 500 V trifásica, de 0,37 kW a 15 kW.
- ✓ De 525 V a 600 V trifásica, de 0,75 kW a 15 kW.

Los variadores Altivar 31 están referenciados con dos interfaces hombre-máquina diferentes:

b1 ATV 31Hpppp con visualizador y teclas de navegación para los menús.

b2 ATV 31HppppA con visualizador, teclas de navegación en los menús y control local (Marcha/Parada y potenciómetro para ajustar la consigna de velocidad).

FUNCIONES:

El variador Altivar 31 dispone de seis entradas lógicas, tres entradas analógicas, una salida lógica/analógica y dos salidas de relé. Las principales funciones integradas son las siguientes:

- ✓ Protecciones para motor y variador.
- ✓ Rampas de aceleración y deceleración, lineales, en S, en U y personalizadas.
- ✓ Más/menos velocidad. b 16 velocidades preseleccionadas.
- ✓ Consignas y regulador PI.
- ✓ Mando 2 hilos/3 hilos. b Lógica de freno.
- ✓ Recuperación automática con búsqueda de velocidad y re-arranque automático.
- ✓ Configuración de fallos y de tipos de paradas.



Figura 2.7 Imagen del variador ATV31.

f) Controlador lógico programable.

El tablero cuenta con un PLC de la marca siemens S7-200 con fuente de alimentación, CPU, módulo de entradas y salidas digitales, módulo de entradas analógicas y salidas analógicas.

El CPU es el 224 y su módulo de entradas es EM231 con 8 entradas analógicas, módulo de salida es EM 232 con 4 salidas analógicas.

- **Módulo de entrada EM231:**

Rango de entrada: Canales de 0 a 5:	+10V, +5V, +/-2.5V.
Canales de 6 a 7:	+10V, +5V, +/-5V, +/-2.5V y de 0 a 20mA.

- **Módulo de salida EM232:**

Rango de señales:	Tensión +/-10V
	Corriente de 0 a 20mA



Figura 2.8. Imagen del PLC S7-200

2.3 El Controlador Lógico Programable S7-200

La gama S7-200 comprende diversos sistemas de automatización pequeños (Micro-PLCs) que se pueden utilizar para numerosas tareas.

El S7-200 vigila las entradas y cambia el estado de las salidas conforme al programa de usuario que puede incluir operaciones de lógica booleana, operaciones con contadores y temporizadores, operaciones aritméticas complejas, así como comunicación con otros aparatos inteligentes. Gracias a su diseño compacto, su configuración flexible y su amplio juego de operaciones, el S7-200 es especialmente apropiado para solucionar numerosas tareas de automatización.

a) CPU S7-200

La CPU S7-200 incorpora en una carcasa compacta un microprocesador, una fuente de alimentación integrada, así como circuitos de entrada y de salida que conforman un potente Micro-PLC (figura 3.8). Tras haber cargado el programa en el S7-200, éste contendrá la lógica necesaria para supervisar y controlar los aparatos de entrada y salida de la aplicación.

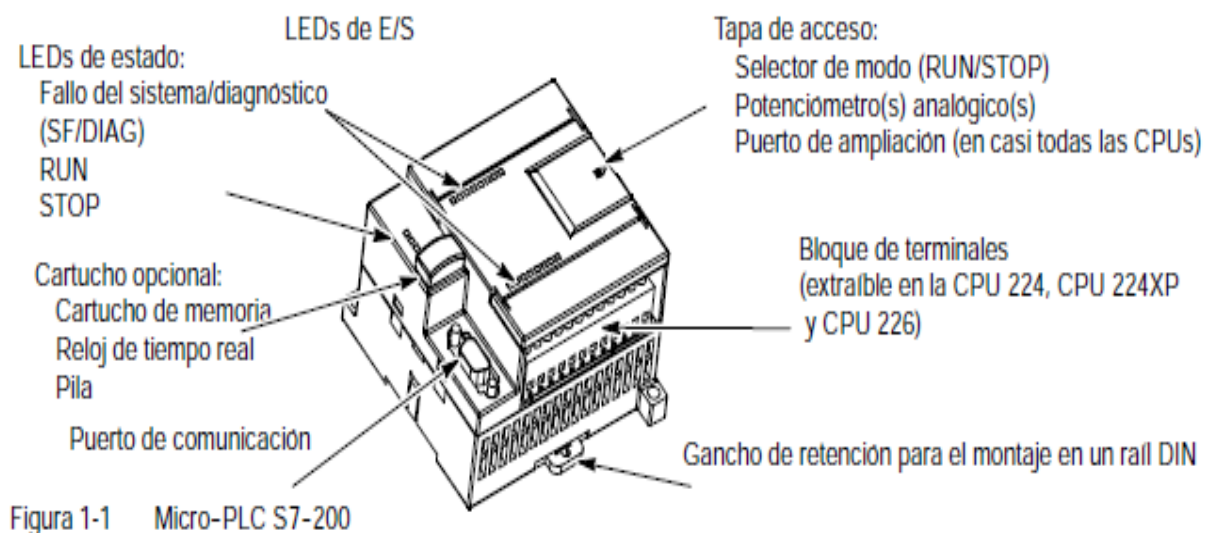


Figura 2.9. Partes de un PLC S7-200

Siemens ofrece diferentes modelos de CPUs S7-200 que ofrecen una gran variedad de funciones y prestaciones para crear soluciones efectivas de automatización destinadas a numerosas aplicaciones. En la tabla 1-1 se comparan de forma resumida algunas de las funciones de la CPU.

FUNCIÓN	CPU 221	CPU 222	CPU 224	CPU 224XP CPU 224XPSi	CPU 226
Dimensiones físicas (mm)	90 x 80 x 62	90 x 80 x 62	120.5 x 80 x 62	140 x 80 x 62	190 x 80 x 62
Memoria del programa:					
con edición en runtime	4 096 bytes	4096 bytes	8 192 bytes	12 288 bytes	16 384 bytes
sin edición en runtime	4 096 bytes	4096 bytes	12 288 bytes	16 384 bytes	24 576 bytes
Memoria de datos	2 048 bytes	2 048 bytes	8 192 bytes	10 240 bytes	10 240 bytes
Memoria de backup	50 horas (tip)	50 horas (tip)	100 horas (tip)	100 horas (tip)	100 horas (tip)
E/S integradas					
Digitales	6E/4S	8E/6S	14E/10S	14E/10S	24E/16S
Analógicos				2E/1S	
Módulos de ampliación	0 módulos	2 módulos	7 módulos	7 módulos	7 módulos
Contadores rápidos					
fase simple	4 a 30 KHz	4 a 30 KHz	6 a 30 KHz	4 a 30/ 2 a 200 KHz	6 a 30 KHz
dos fases	2 a 20 KHz	2 a 20 KHz	4 a 20 KHz	2 a 20/ 1 a 100 KHz	4 a 20 KHz
Salidas de Impulsos (DC)	2 a 20 KHz	2 a 20 KHz	2 a 20 KHz	2 a 100 KHz	2 a 20 KHz
Potenciómetros analógicos	1	1	2	2	2
Reloj de tiempo real	Cartucho	Incorporado	Incorporado	Incorporado	Incorporado
Puertos de comunicación	1 RS-485	1 RS-485	2 RS-485	2 RS-485	2 RS-485
Aritmética en coma flotante	Sí				
Tamaño de la imagen de E/S digitales	256 (128 E/ 128 S)				
Velocidad de ejecución Boleana	0.22 microsegundos / operación				

Tabla 1-1. Módulos de Ampliación.

b) Módulos de ampliación S7-200

La gama S7-200 incluye una gran variedad de módulos de ampliación para poder satisfacer aún mejor los requisitos de la aplicación. Estos módulos se pueden utilizar para agregar funciones a la CPU S7-200. En la tabla 1-2 figura una lista de los módulos de ampliación disponibles en la actualidad.

Tabla 1-2 Módulos de ampliación S7-200

Módulos de ampliación	Tipo de datos			
Módulos digitales				
Entrada	8 entradas DC	8 entradas AC	16 entradas DC	
Salida	4 salidas DC	4 salidas de relé	8 salidas de relé	
	8 salidas DC	8 salidas AC		
Combinacion	4 entradas DC / 4 salidas DC	8 entradas DC / 8 salidas DC	16 entradas DC / 16 salidas DC	32 entradas DC / 32 salidas DC
	4 entradas DC / 4 salidas de relé	8 entradas DC / 8 salidas de relé	16 entradas DC / 16 salidas de relé	32 entradas DC / 32 salidas de relé
Módulos analógicos				
Entrada	4 entradas analógicas	8 entradas analógicas	4 entradas termopar	8 entradas termopar
	2 entradas RTD	4 entradas RTD		
Salida	2 salidas analógicas	4 salidas analógicas		
Combinacion	4 entradas analógicas 4 salidas analógicas			
Módulos inteligentes				
	Posicion	Modem	PROFIBUS-DP	
	Ethernet	Ethernet IT		
Otros módulos				
	ASInterface	SIWAREX MS ¹		
¹ Información detallada no incluida en el anexo A. Consulte la documentación del módulo.				

¹ Información detallada no incluida en el anexo A. Consulte la documentación del módulo.

Tabla 1-2. Módulos de ampliación en S7-200

2.4 Software de programación STEP 7-Micro/WIN

El paquete de programación STEP 7-Micro/WIN constituye un entorno de fácil manejo para desarrollar, editar y observar el programa necesario con objeto de controlar la aplicación.

STEP 7-Micro/WIN provee tres editores que permiten desarrollar de forma cómoda y eficiente el programa de control. Para encontrar fácilmente la información necesaria, así mismo se puede monitorear el estado del programa desde el mismo software además se tiene un asistente PID donde se puede observar gráficamente las variables que intervienen.

2.4.1 Requisitos del sistema

STEP 7-Micro/WIN se puede ejecutar en un ordenador (PC).

El PC deben cumplir los siguientes requisitos mínimos:

- Sistema operativo: Windows 2000, Windows XP, Vista
- 350 Mbytes libres en el disco duro (como mínimo)
- Ratón (recomendado)

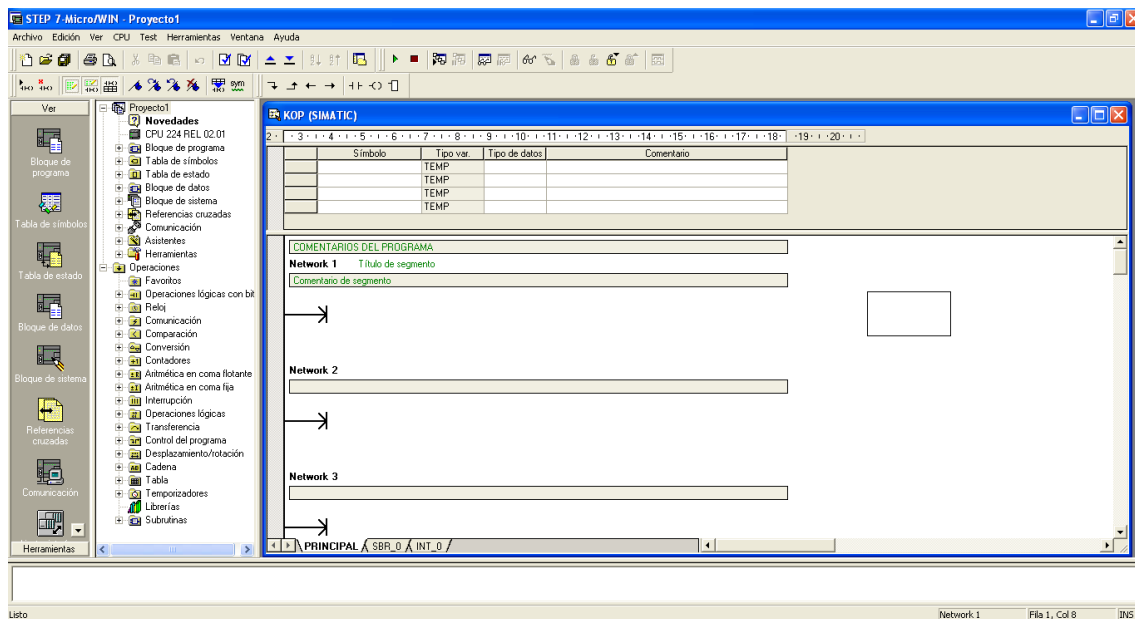


Figura 2.10. Vista del entorno de programación.

2.4.2 Opciones de comunicación

Siemens ofrece dos opciones de programación para conectar el PC al S7-200: una conexión directa vía un cable PPI multimaestro, o bien un procesador de comunicaciones (CP) con un cable MPI.

El cable de programación PPI multimaestro es el método más usual y más económico de conectar el PC al S7-200. Este cable une el puerto de comunicación del S7-200 con el puerto serie del PC. El cable de programación PPI multimaestro también se puede utilizar para conectar otros dispositivos de comunicación al S7-200.

2.4.3 Configuración del STEP 7-Micro/WIN

a) Iniciar STEP 7-Micro/WIN

Haga clic en el icono de STEP 7-Micro/WIN para abrir un nuevo proyecto.

Puede utilizar los iconos de la barra de navegación para abrir los elementos de proyecto de STEP 7-Micro/WIN. En la barra de navegación, haga clic en el icono “Comunicación” para abrir el cuadro de diálogo correspondiente. Utilice este cuadro de diálogo para configurar la comunicación de STEP 7-Micro/WIN.

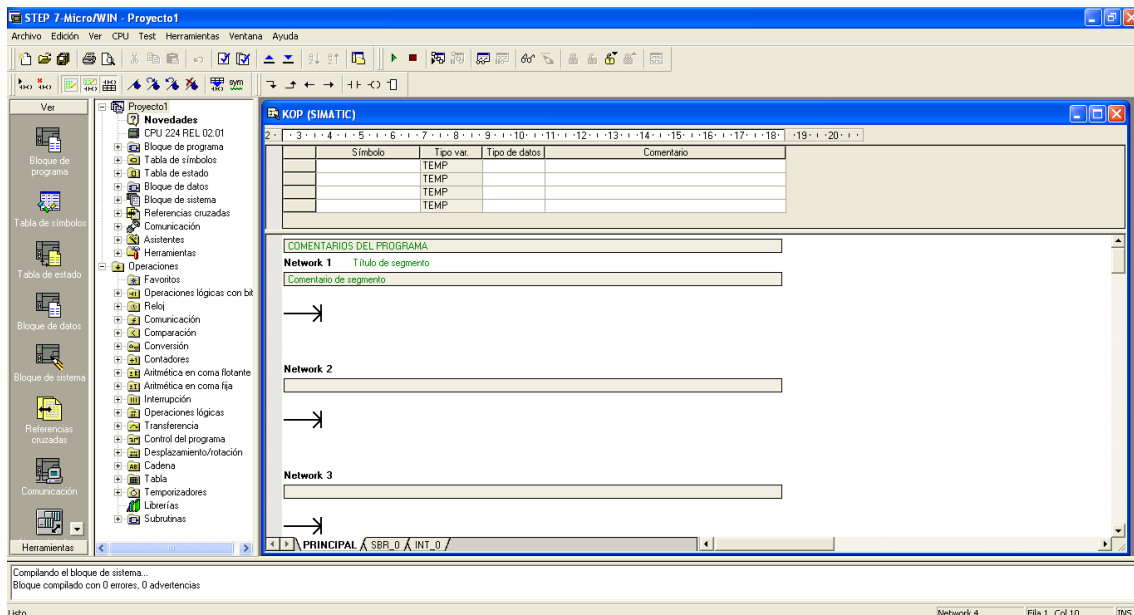


Figura 2.11. Entorno de programación S7-200

b) Verificar los parámetros de comunicación de STEP 7-Micro/WIN

En el proyecto de ejemplo se utilizan los ajustes estándar de STEP 7-Micro/WIN y del cable multimaestro RS-232/PPI.

Para verificar los ajustes:

1. Vigile que la dirección del cable PC/PPI esté ajustada a 0 en el cuadro de diálogo “Comunicación”.
2. Vigile que la interfaz del parámetro de red esté configurada para el cable PC/PPI (COM1).
3. Vigile que la velocidad de transferencia esté ajustada a 9,6 kbit/s.

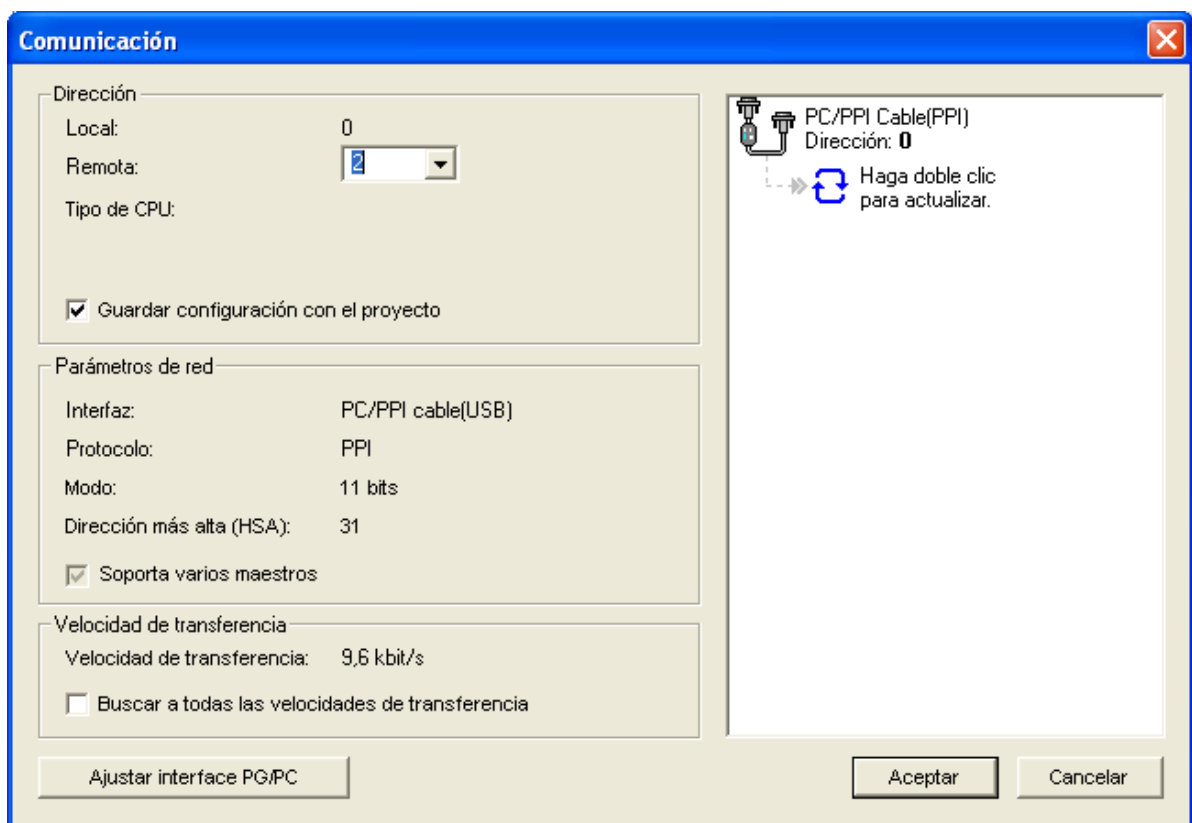


Figura 2.12. Ajuste de parámetros de comunicación

c) Establecer la comunicación con el S7-200

Utilice el cuadro de diálogo “Comunicación” para establecer la comunicación con el S7-200:

1. En el cuadro de diálogo “Comunicación”, haga doble clic en el ícono “Actualizar”. STEP 7-Micro/WIN buscará el S7- 200 y visualizará un ícono “CPU” correspondiente a la CPU S7-200 conectada.
2. Seleccione el S7-200 y haga clic en “Aceptar”. Si STEP 7-Micro/WIN no encuentra el S7-200, compruebe los parámetros de Comunicación y repita los pasos descritos arriba. Tras haber establecido la comunicación con el S7-200 podrá crear el programa de ejemplo y cargarlo.

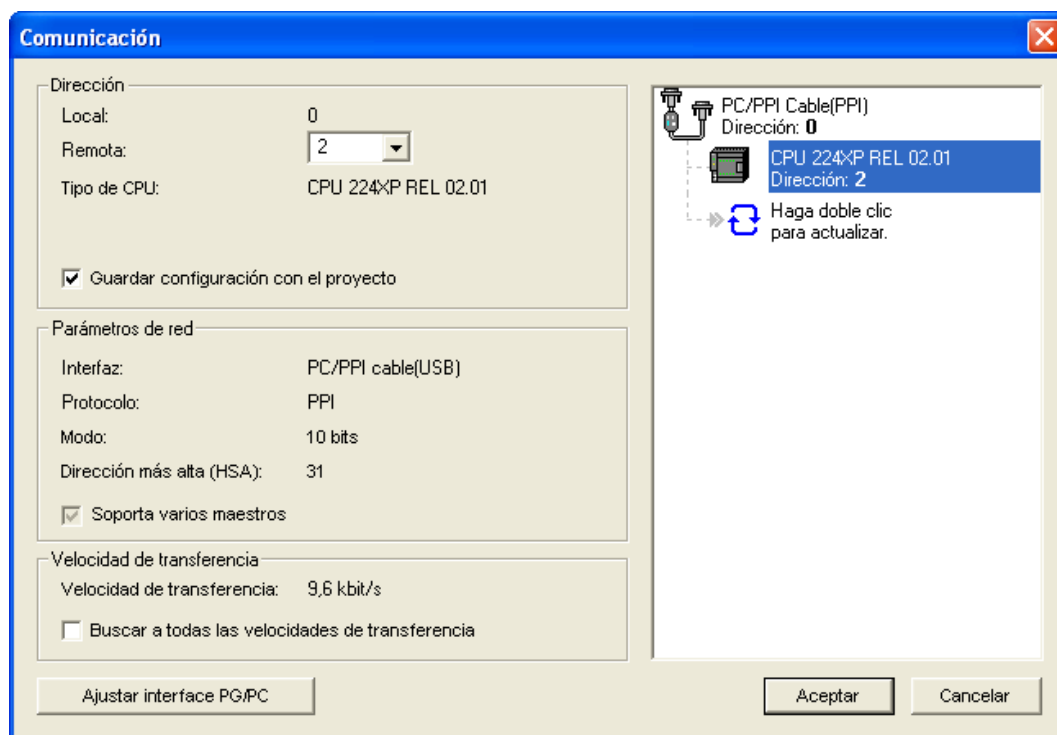


Figura 2.13. Imagen de la comunicación entre PC-PLC.

d) Crear programa

Insertamos las instrucciones de control al autómata con sus direcciones de entrada y salida, analógicas y digitales según lo que se quiere controlar.

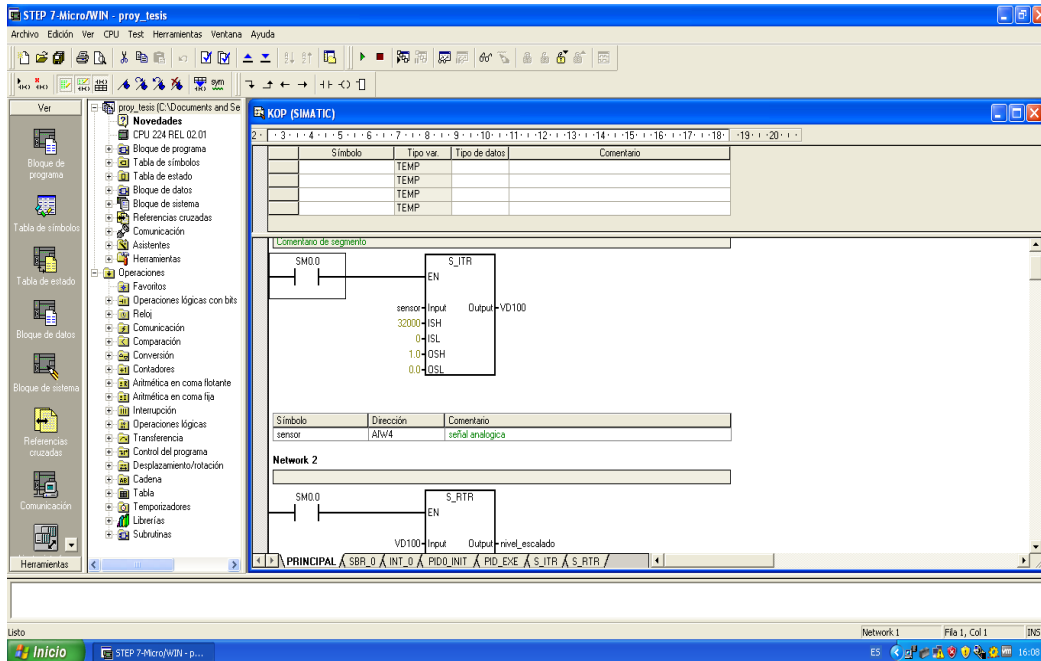


Figura 2.14. Subrutina en ladder.

e) Cargar un programa

1. En la barra de herramientas, haga clic en el botón “Cargar” o elija el comando de menú **Archivo > Cargar** para cargar el programa en la CPU (v. fig. 2-11).
2. Haga clic en “Aceptar” para cargar los elementos de programa en el S7-200.
3. Si el S7-200 está en modo RUN, aparecerá un mensaje indicando que debe cambiar el S7-200 a modo STOP.
4. Haga clic en “Sí” para poner el S7-200 en modo STOP.

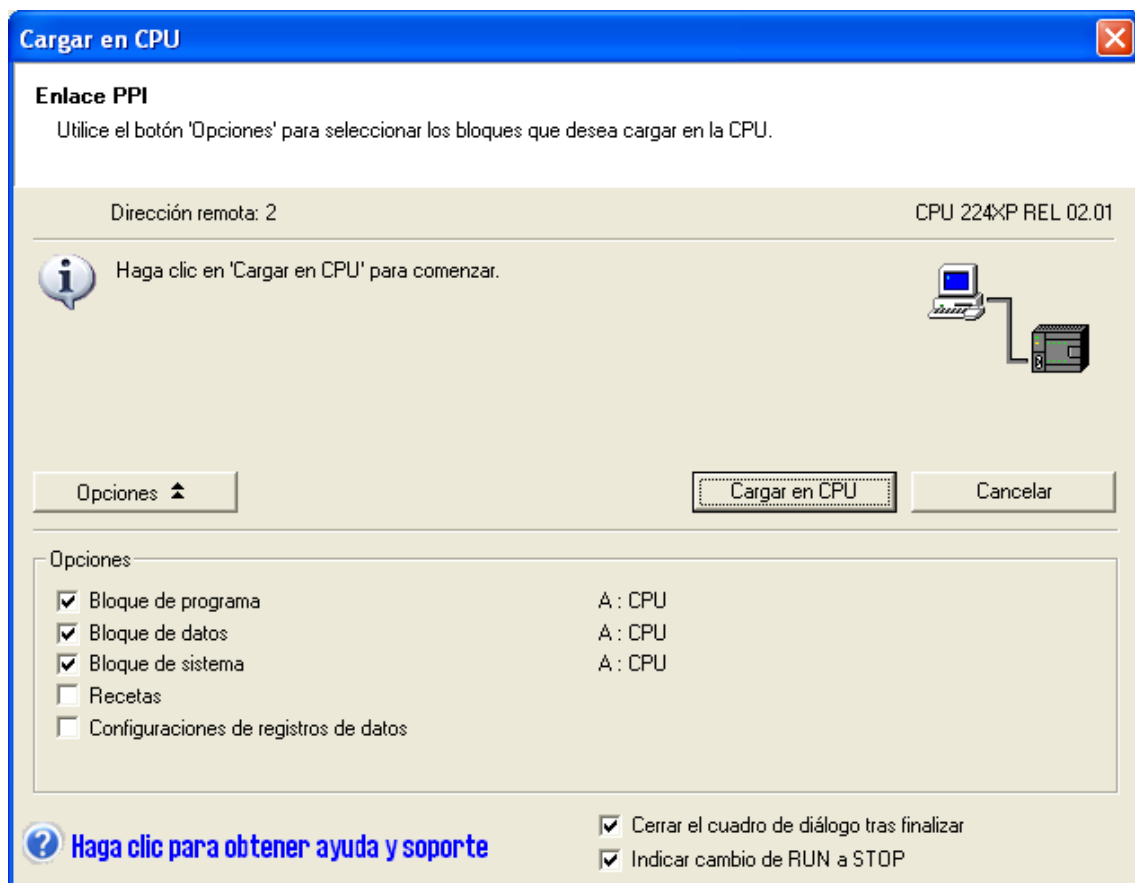
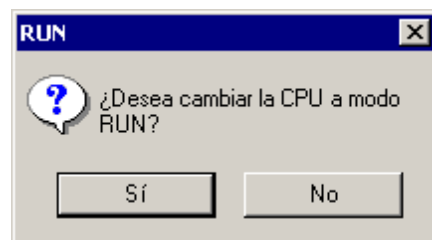


Figura 2.15. Cargar Programas al CPU del PLC.

f) Poner el S7-200 en modo RUN

Para que STEP 7-Micro/WIN pueda poner el S7-200 en modo RUN, el selector de modo de la CPU debe estar en posición TERM o RUN. El programa se ejecuta cuando el S7-200 cambia a modo RUN:

1. En la barra de herramientas, haga clic en el botón "RUN" o elija el comando de menú **CPU > RUN**.
2. Haga clic en "Aceptar" para cambiar el modo de operación del S7-200.
3. Cuando el S7-200 cambia a modo RUN, el LED correspondiente a la salida Q0.0 se enciende y se apaga a medida que el S7-200 ejecuta el programa.
4. Poner el S7-200 en modo RUN



2.5 Software para el cálculo de la función de transferencia

Para el cálculo de la función de transferencia se usaron dos software que requieren licencia para ser utilizados, LabView y Matlab.

2.5.1 NI LabVIEW

LabVIEW fue creado en 1976 por la empresa National Instruments (NI) con el propósito de funcionar sobre máquinas Apple Macintosh (MAC). Actualmente está disponible en los principales sistemas operativos utilizados en el mundo, como es Windows y Linux, entre otros.

“LabVIEW es un revolucionario entorno de desarrollo gráfico con funciones integradas para realizar adquisición de datos, control de instrumentos, análisis de medida y presentaciones de datos. LabVIEW le da la flexibilidad de un potente ambiente de programación, pero mucho más sencillo que los entornos tradicionales”.

Esta definición es bastante clara, el software LabVIEW tiene funciones específicas para acelerar el desarrollo de aplicaciones de medida, control y automatización, nos proporciona herramientas poderosas para que el usuario pueda crear aplicaciones sin líneas de código (lenguaje G) y nos permite colocar objetos ya contruidos para lograr crear interfaces de usuario rápidamente. Después, es uno mismo el que especifica las funciones del sistema construyendo diagramas de bloques.

No podemos dejar de mencionar que LabVIEW se puede conectar con todo tipo de hardware, incluyendo instrumentos de escritorio, tarjetas insertables, controladores de movimiento y controladores lógicos programables (PLCs).

Otra de las razones que explican la importancia y le dan un punto a favor a este software es que como las necesidades de las aplicaciones van cambiando con el tiempo, los sistemas definidos y creados por el usuario de LabVIEW tienen la movilidad y la flexibilidad necesaria para adecuarse sin la necesidad de incorporar equipos nuevos.

Por lo descrito anteriormente, un sistema basado en LabVIEW, simplifica el desarrollo de sistemas y tiene la habilidad de reutilizar su código, tiene acceso a sistemas de instrumentación completos con un coste mucho más bajo que un único instrumento comercial.

Finalmente, una de las ventajas que posee este software y que nos llamó la atención para nuestro trabajo, fue que LabVIEW es ideal para la adquisición de datos y monitorización en la industria.

2.5.2 Matlab

MATLAB es un entorno de cálculo técnico de altas prestaciones para cálculo numérico y visualización. Integra:

- Análisis numérico
- Cálculo matricial
- Procesamiento de señales
- Gráficos

Es un entorno fácil de usar, donde los problemas y las soluciones son expresados como se escriben matemáticamente, sin la programación tradicional. El nombre *MATLAB* proviene de "MATrix LABoratory" (Laboratorio de Matrices). *MATLAB* fue escrito originalmente para proporcionar un acceso sencillo al software matricial desarrollado por los proyectos *LINPACK* y *EISPACK*, que juntos representan lo más avanzado en programas de cálculo matricial. *MATLAB* es un sistema interactivo cuyo elemento básico de datos es una matriz que no requiere dimensionamiento. Esto permite resolver muchos problemas numéricos en una fracción del tiempo que llevaría hacerlo en lenguajes como *C*, *BASIC* o *FORTRAN*. *MATLAB* ha evolucionado en los últimos años a partir de la colaboración de muchos usuarios. En entornos universitarios se ha convertido en la herramienta de enseñanza estándar para cursos de introducción en álgebra lineal aplicada, así como cursos avanzados en otras áreas.

En la industria, *MATLAB* se utiliza para investigación y para resolver problemas prácticos de ingeniería y matemáticas, con un gran énfasis en aplicaciones de control y procesamiento de señales. *MATLAB* también proporciona una serie de soluciones específicas denominadas *TOOLBOXES*. Estas son muy importantes para la mayoría de los usuarios de *MATLAB* y son conjuntos de funciones *MATLAB* que extienden el entorno *MATLAB* para resolver clases particulares de problemas como:

- Procesamiento de señales
- Diseño de sistemas de control
- Simulación de sistemas dinámicos
- Identificación de sistemas
- Redes neuronales y otros.

Probablemente la característica más importante de *MATLAB* es su capacidad de crecimiento. Esto permite convertir al usuario en un autor contribuyente, creando sus propias aplicaciones. En resumen, las prestaciones más importantes de *MATLAB* son:

- Escritura del programa en lenguaje matemático.
- Implementación de las matrices como elemento básico del lenguaje, lo que permite una gran reducción del código, al no necesitar implementar el cálculo matricial.
- Implementación de aritmética compleja.
- Un gran contenido de órdenes específicas, agrupadas en *TOOLBOXES*.
- Posibilidad de ampliar y adaptar el lenguaje, mediante ficheros de script y funciones .m.

2.6 Software KEPServer v5

KEPServer es la plataforma de comunicaciones líder en la industria que proporciona una única fuente de datos de automatización industrial para todas las aplicaciones. El diseño de la plataforma permite a los usuarios conectar, administrar, supervisar y controlar los dispositivos de automatización y aplicaciones de software diverso a través de una interfaz de usuario intuitiva, además no requiere licencia para ser instalado en un ordenador.

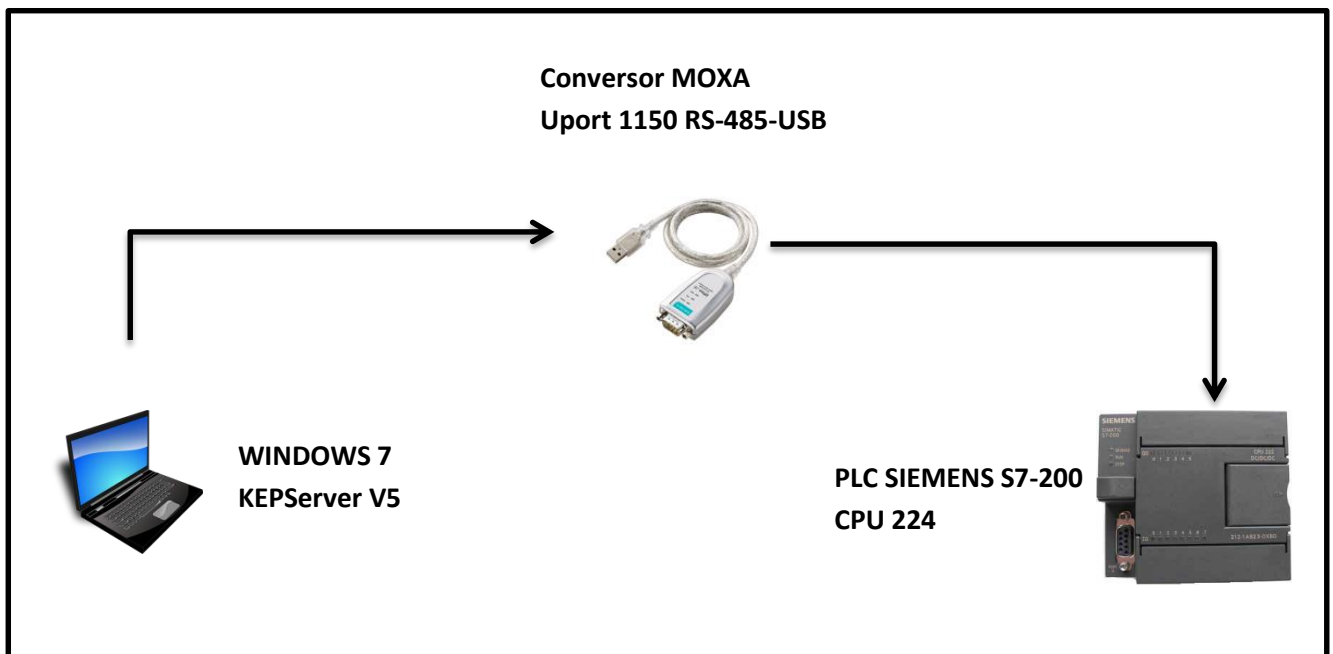
2.7 Comunicación entre KEPServer Y S7-200 CPU 224.

La comunicación entre un PLC y un PC-HMI es típica en el área industrial, aquí se describirá los requerimientos para conectar un s7-200 con un OPC instalado en un computador.

2.7.1 Arquitectura del sistema:

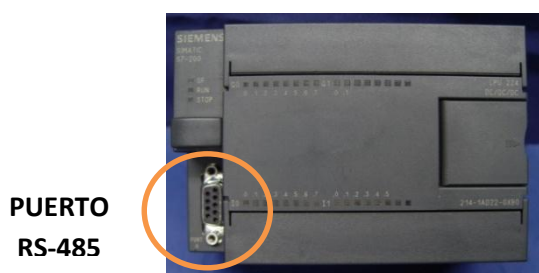
El PLC s7-200 cuenta con un puerto de comunicación RS-485 por lo cual se requiere instalar este tipo puerto en el pc.

Actualmente disponemos de un conversor de comunicaciones RS-485 a USB de la marca MOXA.



2.7.2 Descripción del sistema PLC:

El PLC Siemens S7-200 tiene un puerto de comunicaciones RS-485. Las CPU 224 tiene un puerto RS-485 el cual es configurable desde el software del PLC (STEP-7).



**PUERTO
RS-485**



CABLE PPI

Este cable de programación se utiliza solamente para configurar el PLC y descargar el programa. Para la comunicación HMI OPC no se utiliza ese cable, sino un puerto de comunicaciones seriales RS-485.

Los terminales de comunicación del puerto del PLC son los siguientes. Como puede observarse, los pines 6 y 7 tienen altos voltajes que podrían dañar un interfaz RS-485 estándar, por lo que es necesario construir un cable que interconecte solamente las señales necesarias.

Connector Pin Assignments

The communications ports on the S7-200 CPU are RS-485 compatible on a nine-pin subminiature D connector in accordance with the PROFIBUS standard as defined in the European Standard EN 50170. Table 7-7 shows the connector that provides the physical connection for the communications port and describes the communications port pin assignments.

Table 7-7 Pin Assignments for the S7-200 Communications Port

Connector	Pin Number	PROFIBUS Signal	Port 0/Port 1
	1	Shield	Chassis ground
	2	24 V Return	Logic common
	3	RS-485 Signal B	RS-485 Signal B
	4	Request-to-Send	RTS (TTL)
	5	5 V Return	Logic common
	6	+5 V	+5 V, 100 Ω series resistor
	7	+24 V	+24 V
	8	RS-485 Signal A	RS-485 Signal A
	9	Not applicable	10-bit protocol select (input)
	Connector shell	Shield	Chassis ground

**CUIDADO CON
PINES #6 y #7**

Tabla 2.1. Configuración de los pines para comunicación RS-485.

2.7.3 CONECTOR RS-485 DE LA TARJETA MOXA UPORT 1150

Disposición de pines de Uport 1150



Pin	422/485
1	TX+
2	TX-
3	RX+
4	RX-
5	GND

PLC: RS-485 PROFIBUS		PC: RS-485 GENÉRICO	
Pin#	Descripción	Pin#	Descripción
1	Tierra (Shield)	1	Gnd
2	Hilo de Retorno		No conectado
3	Señal B	4,8	RXD+, TXD+
4			
5			
6	+5V - ATENCIÓN		No conectado

CONEXIÓN ENTRE MOXA RS-485 Y RS-485 PROFIBUS DE SIEMENS

PLC : RS-485 PROFIBUS		MOXA Uport rs-485	
Pin#	Descripción	Pin#	Descripción
1	GND	1	TX+
2	Hilo de retorno	2	TX-
3	Señal B	3	RX+
4		4	RX-
5		5	GND
6	+5V		
7	+24V		
8	Señal A		
9			

Tabla 2.2. Conexión entre moxa RS-485 y RS-485 profibus de siemens.

Es necesario construir un cable para adaptar RS-485 PROFIBUS-RS 485 MOXA.

2.7.4 Configuración de KEPServer para comunicación del PLC-S7 200 con HMI.

Como se comentó líneas arriba el PLC S7-200 instalado en la planta de novel no cuenta con módulo de comunicación Ethernet para ello se utiliza un cable de comunicación serial RS-485 el cual es adaptado en el puerto DB-9 de PLC y el puerto USB conectado a la computadora luego abrimos nuestro servidor OPC KepServer y lo configuramos de la siguiente manera:

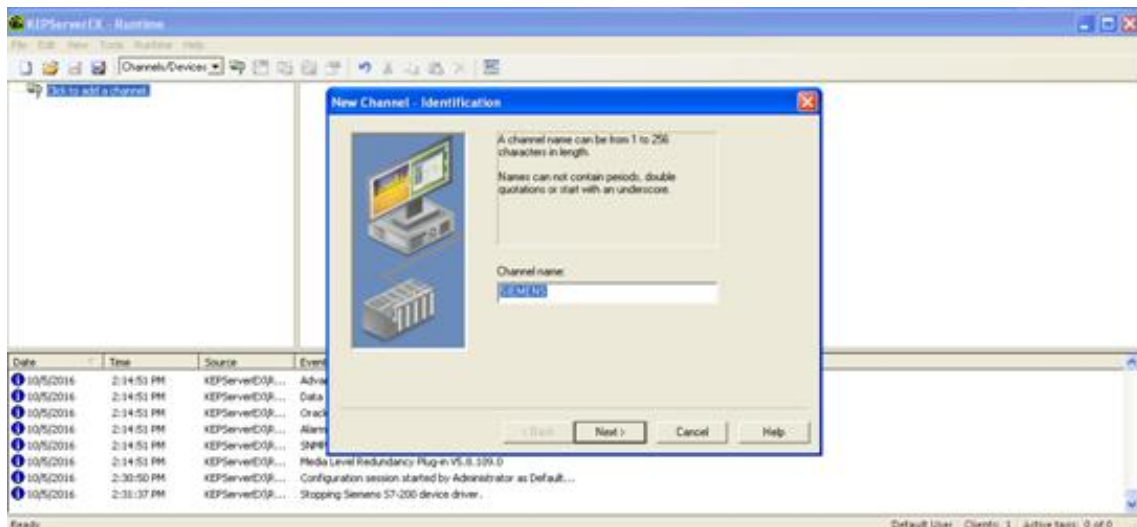


Figura 2.16. Damos nombre a nuestro proyecto (MI PLC)

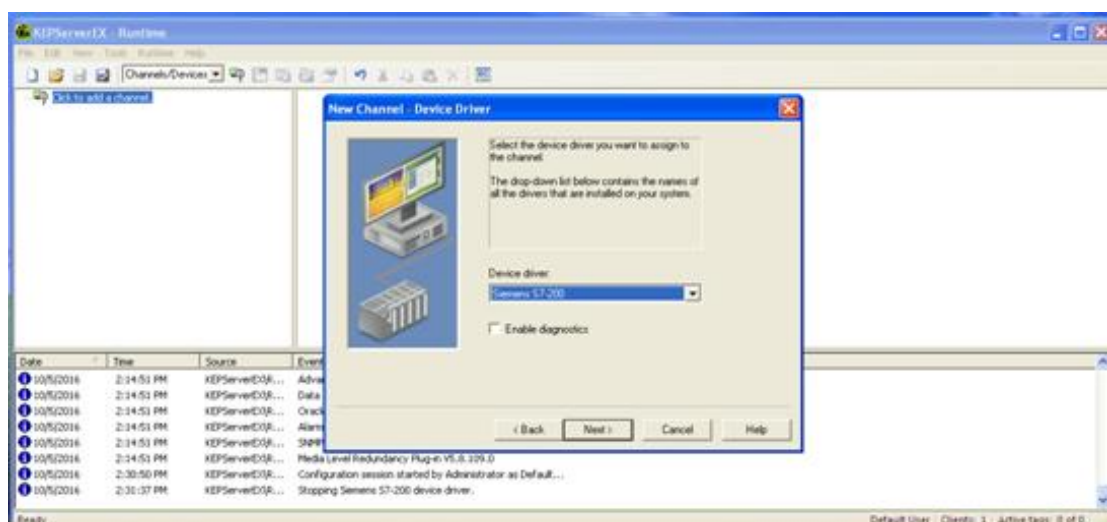


Figura 2.17. Seleccionamos los modelos de PLC al que deseamos conectar (S7-200).

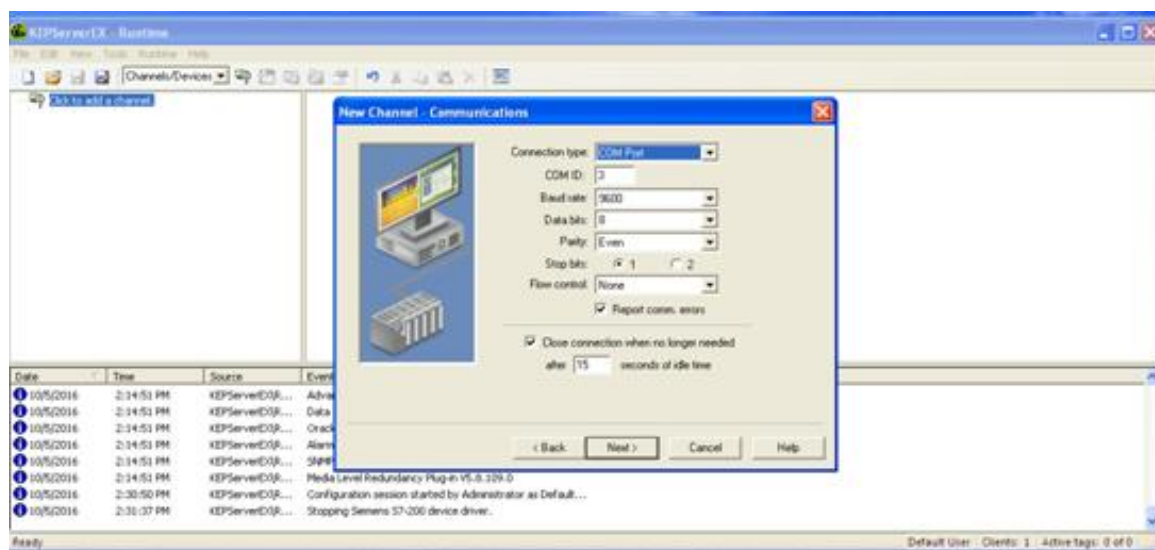


Figura 2.18. Seleccionamos el puerto COM en el cual está conectado el cable serial RS-485.

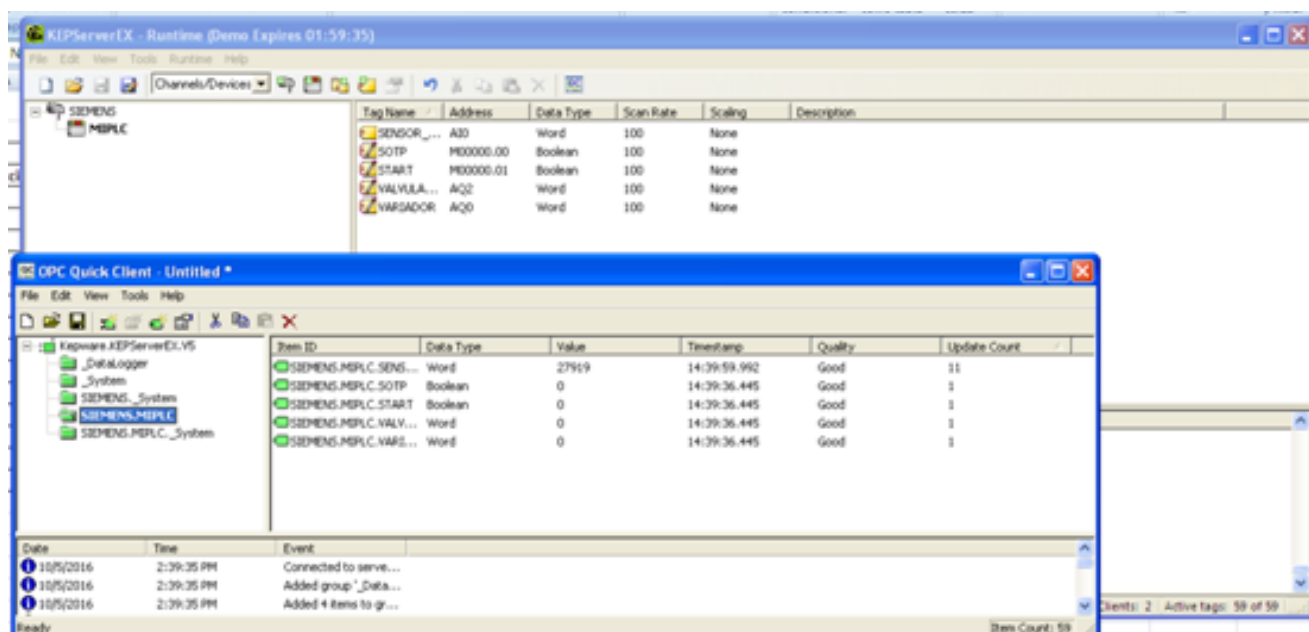


Figura 2.19. Luego de haber configurado correctamente KepServer y S7-200 procedemos a agregar los tags, para verificar que la comunicación es correcta se hace clic en el icono **QC** “OPC Quick Client” donde se observa los valores de los tag son los mismo que MicroWin step7 PLC.

CAPÍTULO 3

OBTENCIÓN DEL MODELO MATEMÁTICO DE LA PLANTA Y SIMULACIÓN

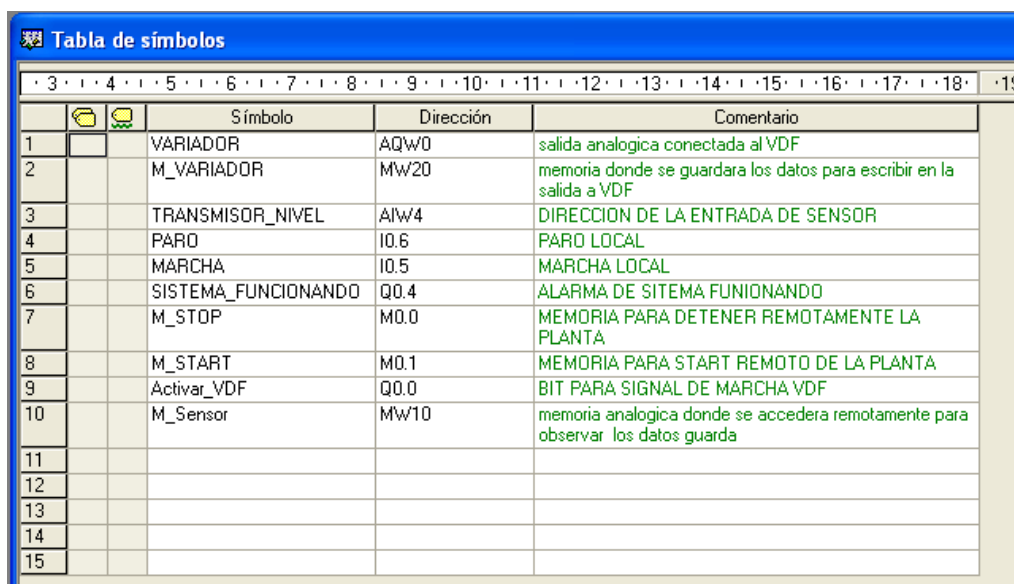
3.1 Adquisición de datos.

Para la adquisición de datos se utiliza el software NI-Labview de National Instrument, así como el software Microwin Step 7 de Siemens para programar las instrucciones del controlador y el OPC KepServer v5 para hacer el intercambio de datos entre el PLC y la computadora.

3.1.1 Programación en MICROWIN STEP-7

Esta lógica de control se muestra en la figura 3.2, la secuencia de control consiste en utilizar las variables de entrada y realizar el control de la planta en base a lo que se procesa de la señal que se lee del sensor para activar las salidas tanto analógica como digital.

Esta rutina realizada en lenguaje Ladder nos permite poner en marcha la planta de nivel, lee los datos del sensor ultrasónico y controla un variador de velocidad (conectado a un motor), aquí podemos ver cómo se comporta la variable de salida (nivel del tanque) al aplicar a la entrada una señal escalón (señal de entrada aplicada al variador de frecuencia).



		Símbolo	Dirección	Comentario
1		VARIADOR	AQW0	salida analogica conectada al VDF
2		M_VARIADOR	MW20	memoria donde se guardara los datos para escribir en la salida a VDF
3		TRANSMISOR_NIVEL	AIW4	DIRECCION DE LA ENTRADA DE SENSOR
4		PARO	I0.6	PARO LOCAL
5		MARCHA	I0.5	MARCHA LOCAL
6		SISTEMA_FUNCIONANDO	Q0.4	ALARMA DE SITEMA FUNIONANDO
7		M_STOP	M0.0	MEMORIA PARA DETENER REMOTAMENTE LA PLANTA
8		M_START	M0.1	MEMORIA PARA START REMOTO DE LA PLANTA
9		Activar_VDF	Q0.0	BIT PARA SIGNAL DE MARCHA VDF
10		M_Sensor	MW10	memoria analogica donde se accedera remotamente para observar los datos guarda
11				
12				
13				
14				
15				

Figura 3.1. Variables utilizadas en el PLC S7-200

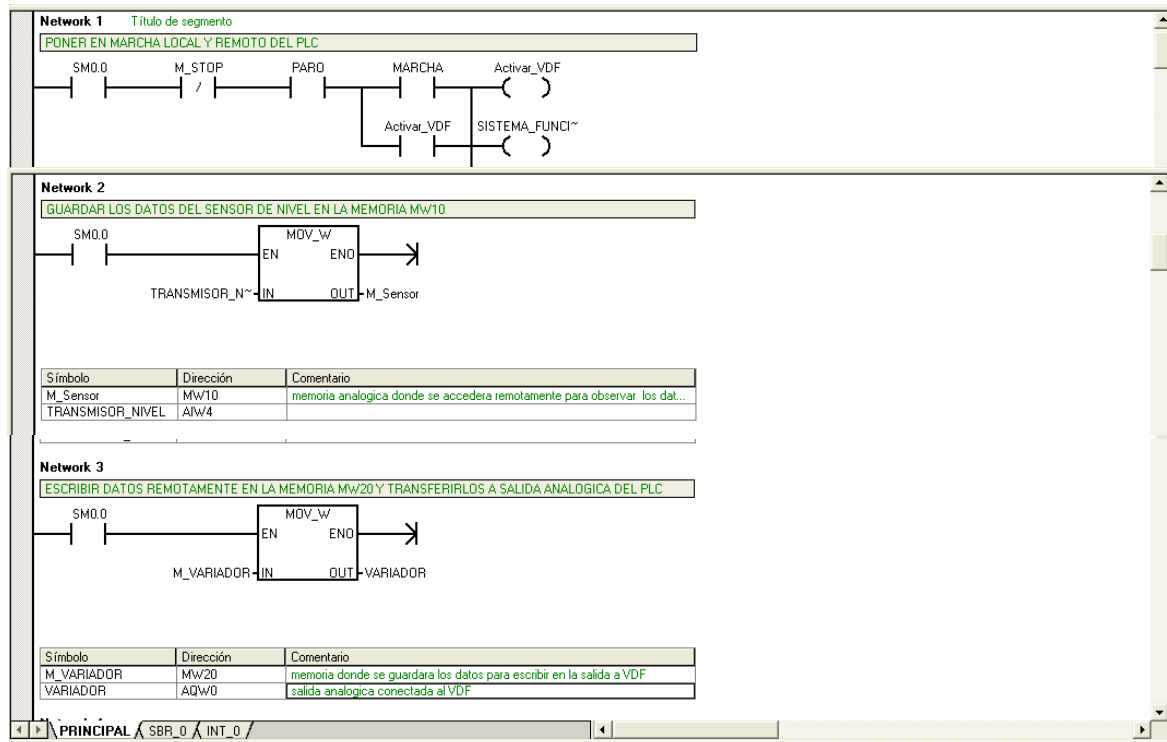


Figura 3.2. Programación Ladder en S7-200.

3.1.2 Programación en LABVIEW

Se desarrolla una interface gráfica para poner en START y STOP la planta de nivel, manipular la frecuencia del variador para controlar la velocidad de giro del motor esta variable es guardada en un documento Excel con valores que van desde 0 hasta 3200 donde 0 es el mínimo valor y 3200 es la máxima frecuencia del motor “60Hz”, así mismo se monitorea el estado del sensor ultrasónico y estos son exportados a un documento Excel para posteriormente calcular la función de transferencia de la planta utilizando el programa Matlab.

También visualizamos que el sistema esté funcionando a través de la lámpara RUN.

Variable LabView	Variable PLC	Variable Física
MARCHA	M_START	M0.1
PARO	M_STOP	M0.0
RUN	SISTEMA_FUNCIONANDO	Q0.0
SENSOR	M_Sensor	MW10
VARIADOR	M_VARIADOR	MW20

Tabla 3.1. Variables utilizadas en el PLC S7-200.

En la **figura 3.3**, se muestra la aplicación desarrollada para adquirir los datos de la planta de nivel y calcular la función de transferencia.

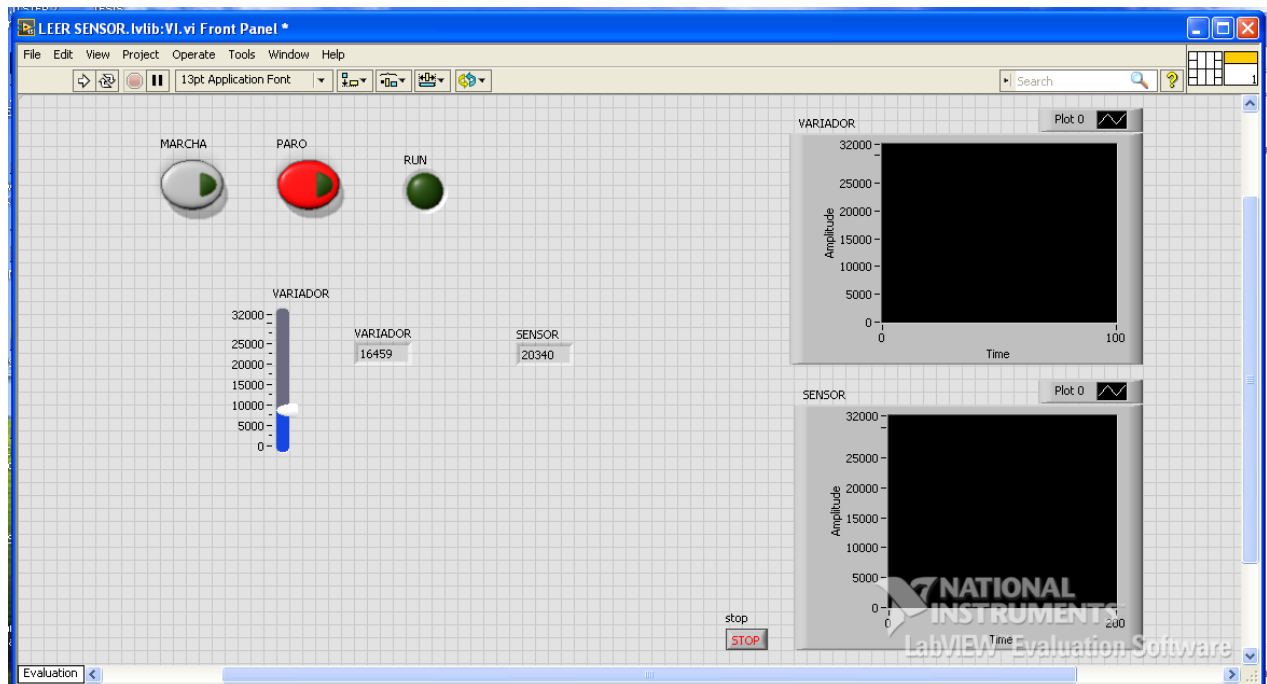


Figura 3.3. Panel frontal de la programación en LabView para adquisición de datos.

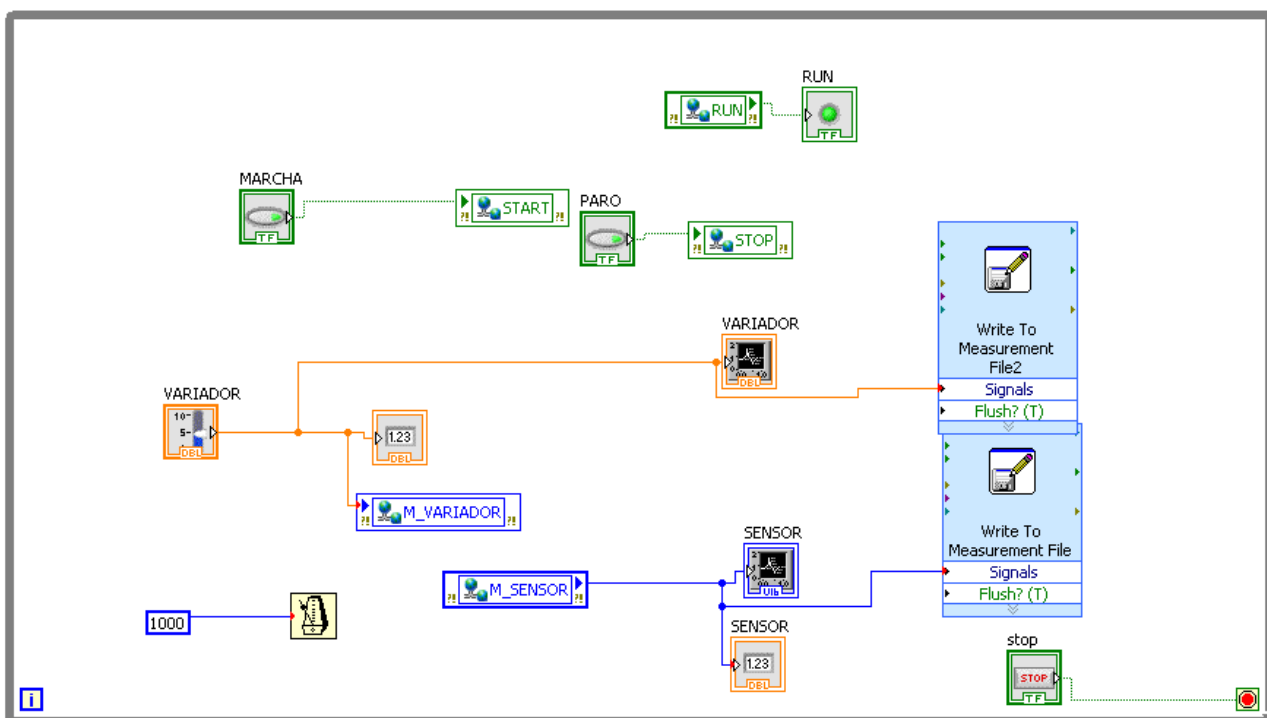


Figura 3.4. Diagrama de bloques de la Programación en LabView.

3.2 Procesamiento de los datos obtenidos en LabView.

Utilizando la ventana del editor de Matlab se cargó los datos que estaban en formato Excel obtenidos de la planta de nivel, se realizó la siguiente rutina para graficar las funciones obtenidas:

Programa para graficar datos obtenidos de la planta de nivel.

```
A=xlsread('VARIADOR.xlsx');%señal de entrada
B=xlsread('SENSOR.xlsx');%señal de salida
figure(1)
subplot(2,1,1)
plot(A)
title('Señal PRBS')
xlabel('time (ms)')
ylabel('U1')
subplot(2,1,2)
plot(B)
title('Respuesta')
xlabel('time (ms)')
ylabel('Y1')
```

Gráfica de las señales de entrada y salida en el dominio del tiempo, estos datos fueron adquiridos de la planta de nivel utilizando Labview y exportados a Excel.

Donde “y1” es la señal adquirida del sensor al manipular la frecuencia del variador, “u1” representa la frecuencia del variador en valores de 0 a 32000.

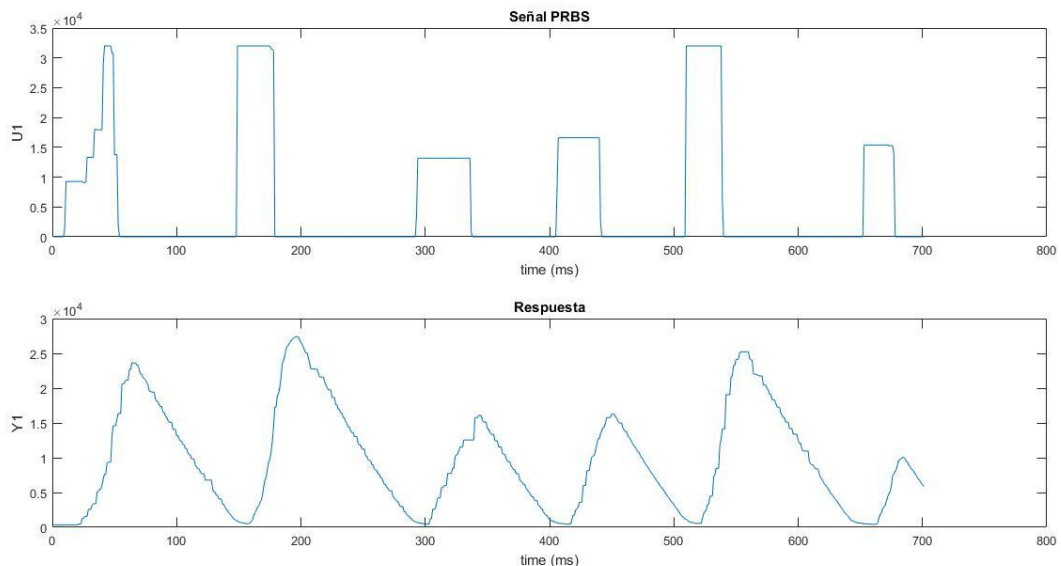


Figura 3.5. U1 señal de entrada, Y1 señal de salida.

3.3 Estimación de la función de transferencia

Utilizando el comando **System_Identification** de Matlab se cargan las variables de entrada y salida obtenidas de la planta de nivel utilizando Labview.

A continuación se describe paso a paso como obtener la función de transferencia.

Paso 1: cargar los archivos obtenidos en Excel de entrada y salida de la planta en la ventana **'workspace'** así como se muestra en la **figura 3.6**.

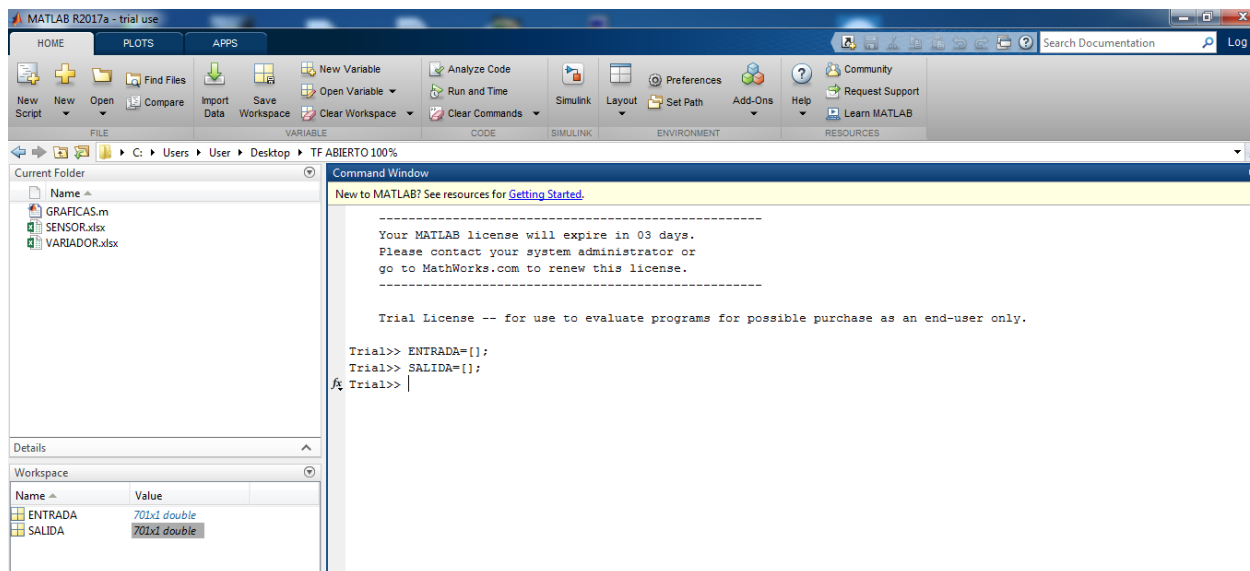


Figura 3.6. Datos de la planta cargados en **Workspace**

Paso 2: Desplegamos en el menú **'import data'** seleccionamos la opción **'time domain data'** la cual permite ingresar nuestros datos obtenidos de la planta de nivel el tiempo de muestreo es de **'0.8 seg'**, la **figura 3.7** muestra una ventana donde se cargan los datos.

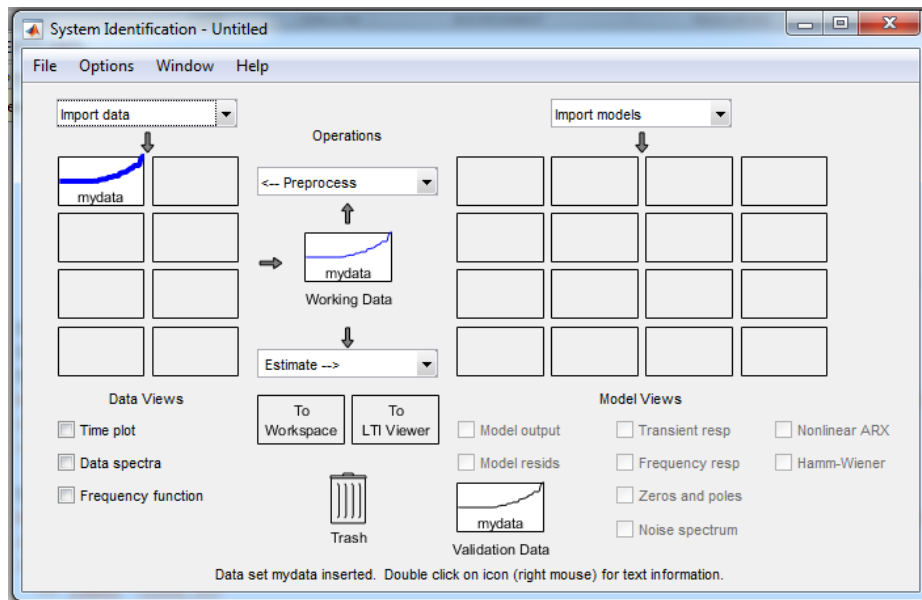


Figura 3.7. Ventana principal de **System Identification**

Paso 3: Desplegamos el menú **preprocess** seleccionamos las opciones **'remove meands'** y **'remove trends'**.

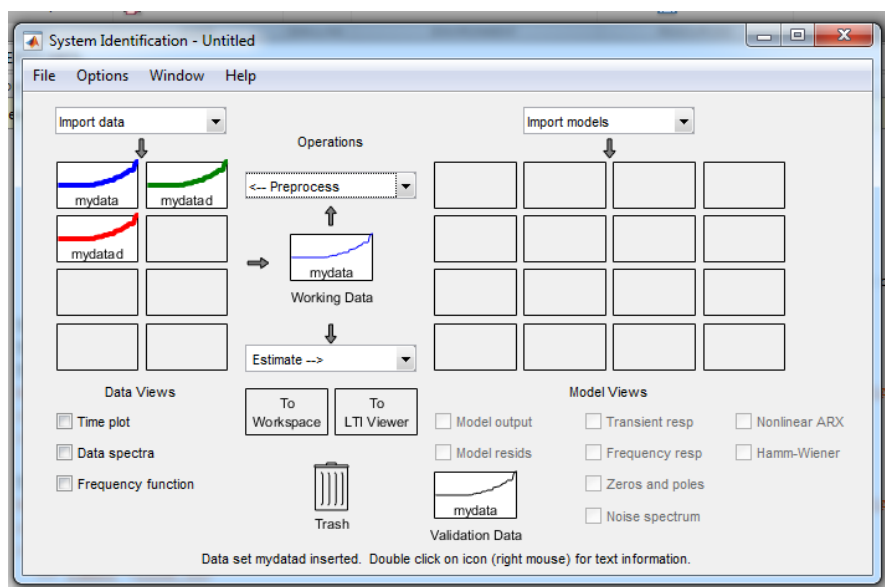


Figura 3.8. Remover extremos y medios.

Paso 4: Desplegamos el menú **'estimate'** y escogemos la **opción 'transfer functions'** elegimos 2 polos y 1 cero para nuestra función,

Se dice que la función se aproxima más a la señal real cuando se acerca al 75%, en nuestro caso estimando la función a 72% es confiable y se aproxima mucho a la señal real, si elegimos "0" ceros y 1 polo nuestra función resulta ser de primer orden y tiene una aproximación del 68% la cual no es muy confiable con nuestra señal real.

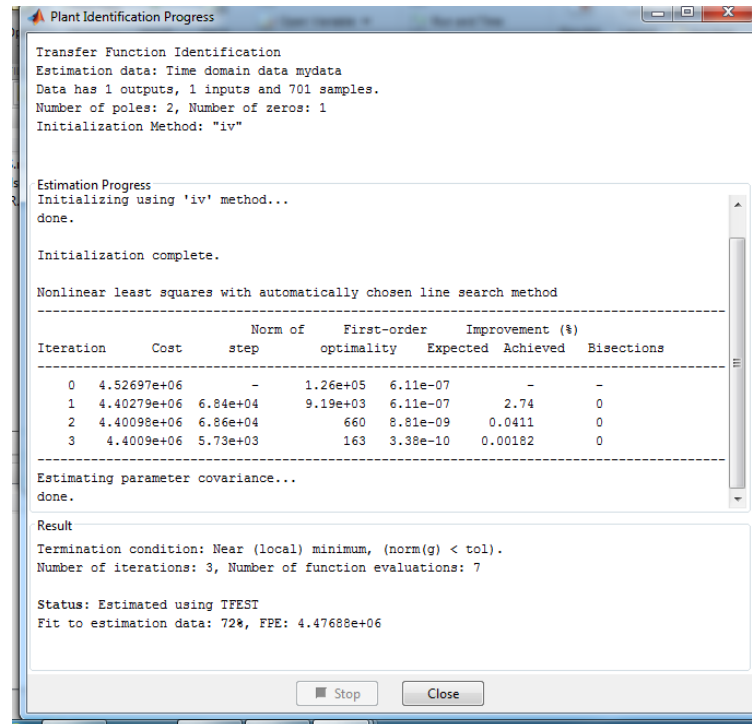


Figura 3.9. Imagen donde se muestra la aproximación de la señal.

3.3.1 Función de transferencia obtenida.

$$H = \frac{-0.02441s + 0.003256}{s^2 + 0.08484s + 0.0019}$$

3.4. Polos y ceros de la función de transferencia.

Nota:

- Si todos los polos de la función de transferencia están en el lado izquierdo del plano s entonces el sistema es estable.
- Un sistema es críticamente estable si uno o más polos están en el eje imaginario del plano s .
- En el estudio de estabilidad sólo los polos de la función de transferencia son importantes, los zeros son irrelevantes.
- Los polos de un sistema son las raíces obtenidas del denominador de la función de transferencia cuando es igualado a cero. Polinomio característico.
- El concepto de estabilidad es aplicado a sistemas a lazo cerrado o a lazo abierto.
- Para nuestro sistema se puede observar en la **figura 3.10** que los polos se encuentran en la parte izquierda del plano s , entonces el sistema es estable.

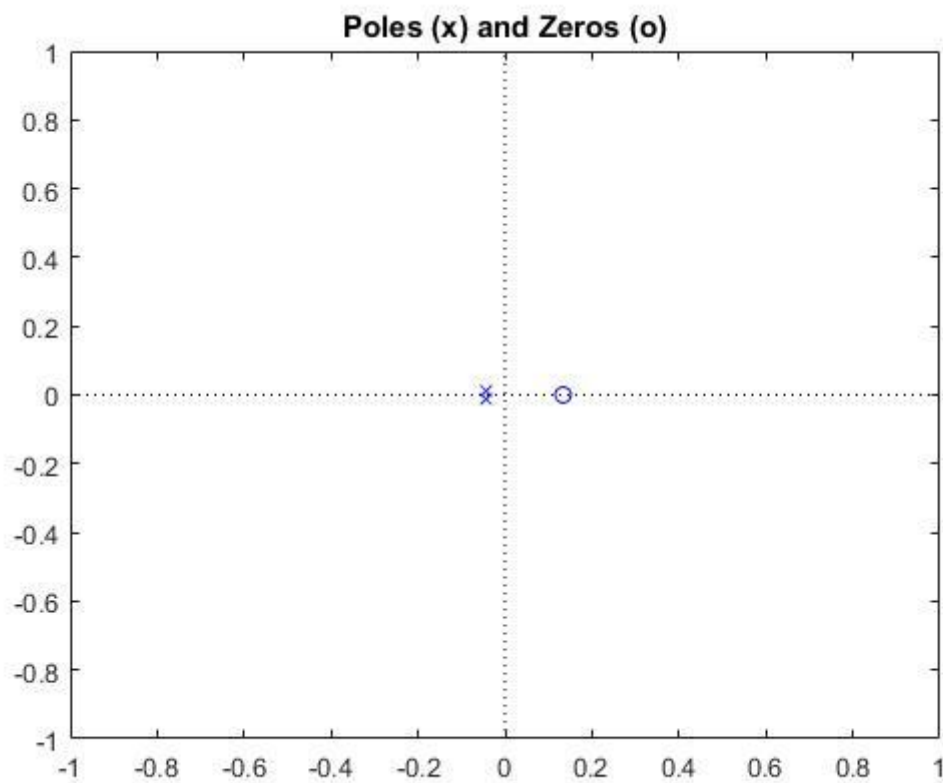


Figura 3.10. Polos y ceros de la función de transferencia de la planta de nivel.

3.5 Estimación del modelo de salida.

En la **figura 3.11** se puede observar que la función estimada por Matlab se asemeja mucho al obtenido en la planta de nivel, el programa Matlab lo calculó una aproximación al 88.89% de la señal real.

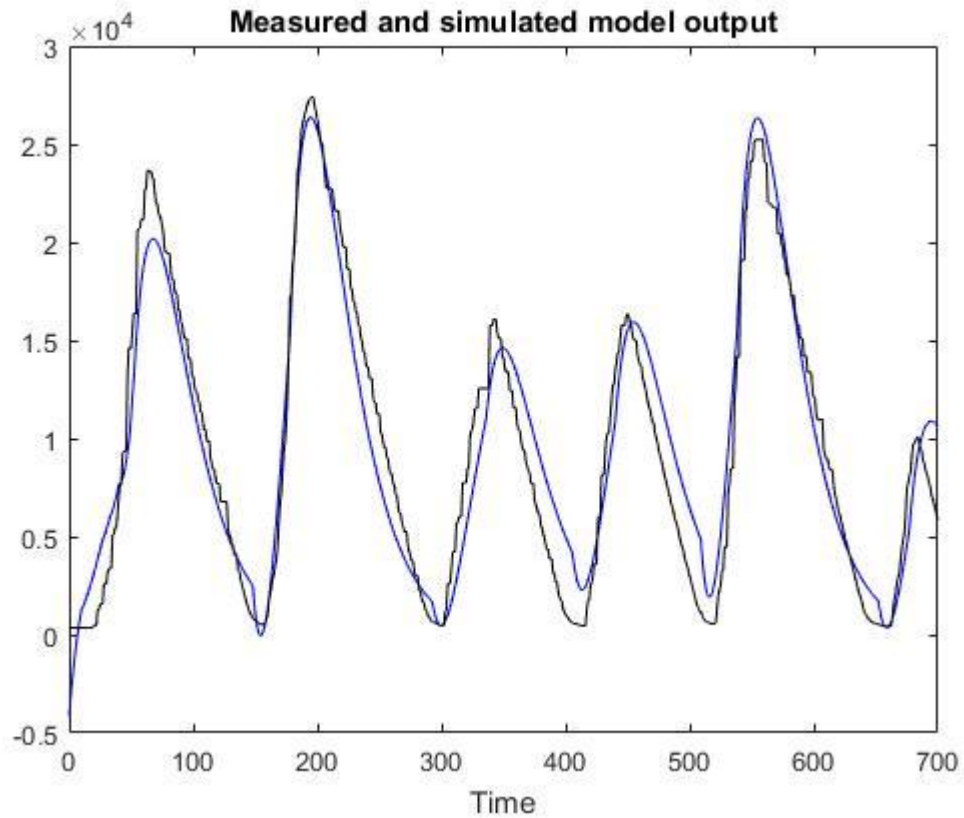


Figura 3.11. Imagen de estimación del modelo de salida.

_____ **Señal real** (obtenida de la planta de nivel)

_____ **Señal estimada** (obtenida del programa Matlab)

3.6 Diseño del controlador.

3.6.1 Comando PIDTOOL

Para diseñar los controladores P, PI, PID cargamos la función de transferencia en la ventana o editor de comandos del Matlab.

```
%ALGORITMO PARA GRAFICAR FUNCION DE TRANSFERENCIA  
num=[ -0.02441 0.003256];  
den=[1 0.08484 0.0019];  
H=tf(num,den);  
pidtool(H)
```

Seleccionamos el tipo de controlador que deseamos haciendo click en el menú **“type”**, luego manipulamos **‘response time’** y **‘transient Behavior’** hasta conseguir el menor tiempo de asentamiento de la señal, teniendo en cuenta que la **‘amplitud pico’** no exceda el 25% del valor fijado en la función escalón.

Para los valores de Kp, Ki, Kd y otros parámetros del controlador haciendo click en el menú **‘show parameters’**.

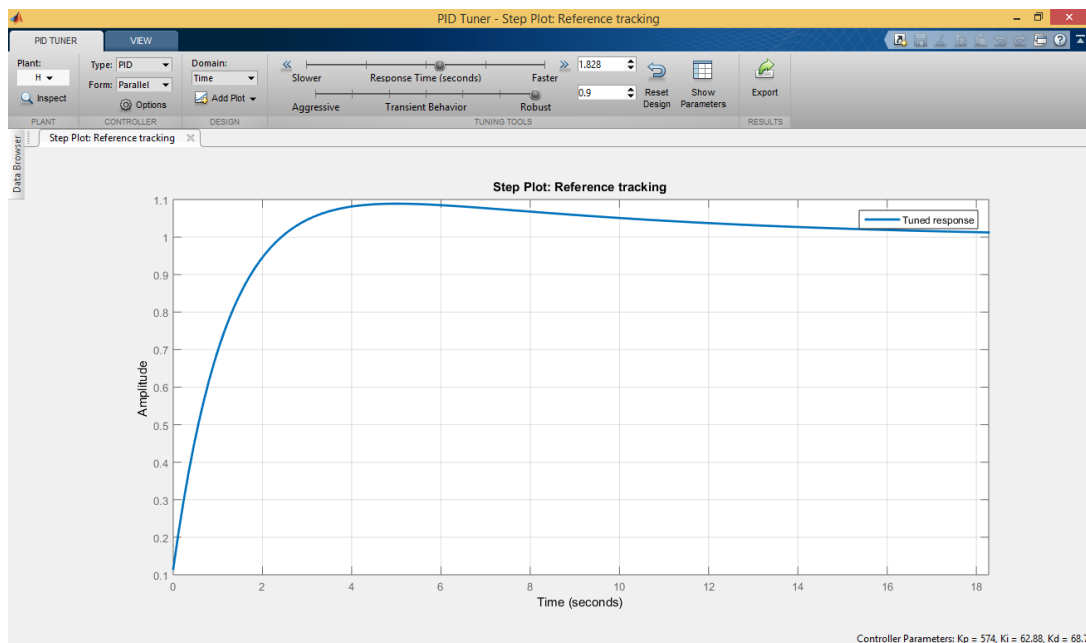


Figura 3.12. Ventana donde se manipula **response time** y **transient Behavior**.

a) Controlador Proporcional

Para el diseño del controlador proporcional se manipuló los valores de **response time y transient Behavior** en la ventana **PID Tuner**, hasta conseguir el menor tiempo de asentamiento de la señal.

Los valores obtenidos de K_p se puestran en la figura 3.13, el tiempo mínimo de asentamiento de la señal es del 101 segundos.

Controller Parameters	
	Tuned
K_p	0.58989
K_i	n/a
K_d	n/a
T_f	n/a
Performance and Robustness	
	Tuned
Rise time	26.6 seconds
Settling time	101 seconds
Overshoot	12.2 %
Peak	0.564
Gain margin	15.4 dB @ 0.115 rad/s
Phase margin	165 deg @ 0.00494 rad/s
Closed-loop stability	Stable

Figura 3.13. Tabla de valores obtenidos para el controlador Proporcional.

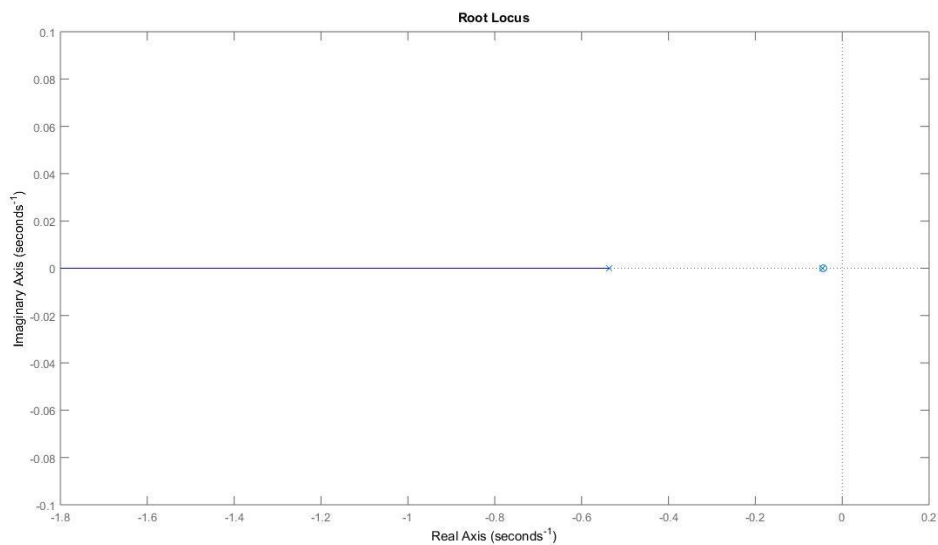


Figura 3.14. Polos y ceros del controlador Proporcional.

b) Control Proporcional Integral (PI)

Para obtener los valores de K_p y K_i del controlador Proporcional Integral se manipuló los valores de **response time y transient Behavior** en la ventana **PID Tuner**, hasta conseguir el menor tiempo de asentamiento de la señal.

En la figura 3.15 se muestra los valores obtenidos de K_p , K_i el tiempo mínimo de asentamiento de la señal es de 41.9 segundos.

Controller Parameters	
	Tuned
K_p	0.59352
K_i	0.013635
K_d	n/a
T_f	n/a
Performance and Robustness	
	Tuned
Rise time	41.9 seconds
Settling time	185 seconds
Overshoot	1.61 %
Peak	1.02
Gain margin	12.8 dB @ 0.0941 rad/s
Phase margin	65 deg @ 0.0272 rad/s
Closed-loop stability	Stable

Figura 3.15. Tabla de valores para el controlado Proporcional Integral.

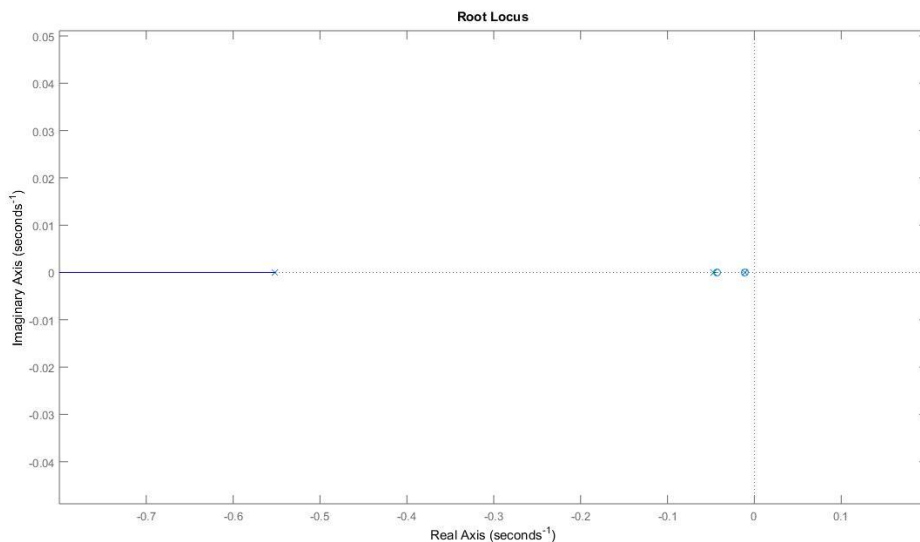


Figura 3.16. Polos y ceros de la función de transferencia al aplicar control PI.

c) Control Proporcional Integral Derivativo (PID)

Para el diseño del controlador Proporcional Integral Derivativo, se manipuló los valores de **response time y transient Behavior** en la ventana **PID Tuner**, hasta conseguir el menor tiempo de asentamiento de la señal.

Para la función de transferencia de la planta de nivel se aplicó el controlador PID donde se obtiene los siguientes valores de K_p , K_i , K_d , mostrados en la **figura 3.17** en un tiempo mínimo de asentamiento de 21.4 segundos.

Controller Parameters	
	Tuned
K_p	1.7092
K_i	0.032694
K_d	16.5901
T_f	n/a
Performance and Robustness	
	Tuned
Rise time	19.2 seconds
Settling time	28.2 seconds
Overshoot	0.807 %
Peak	1.01
Gain margin	7.85 dB @ Inf rad/s
Phase margin	57 deg @ 0.0715 rad/s
Closed-loop stability	Stable

Figura 3.17. Tabla de valores obtenidos al aplicar control PID a la función de transferencia de la planta de nivel.

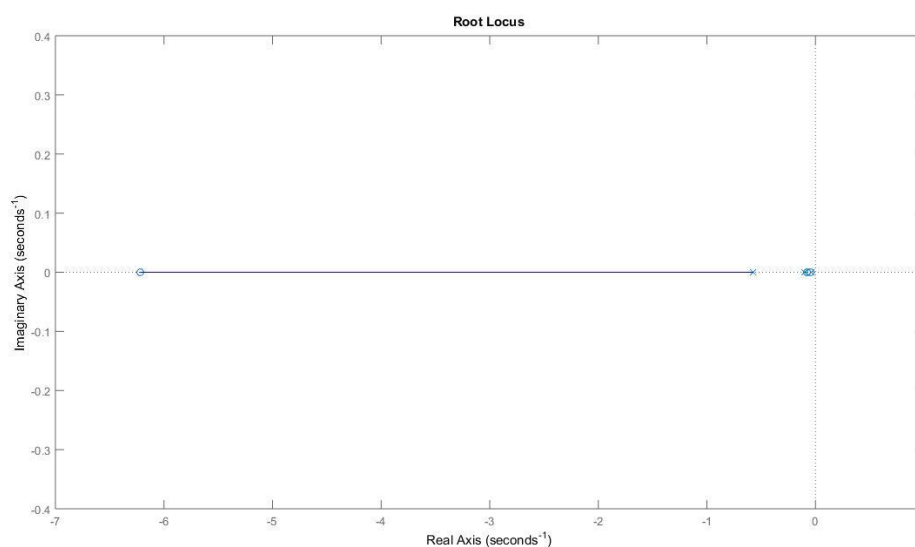


Figura 3.18. Polos y ceros de función de transferencia al aplicar control PID.

3.7. Simulación del modelo de la planta utilizando SIMULINK

Luego de adquirir los datos mediante el comando PIDTOOL de Matlab se pasó a realizar la simulación de la planta mediante diagrama de bloques en Simulink implementando los controladores P, PI, PID.

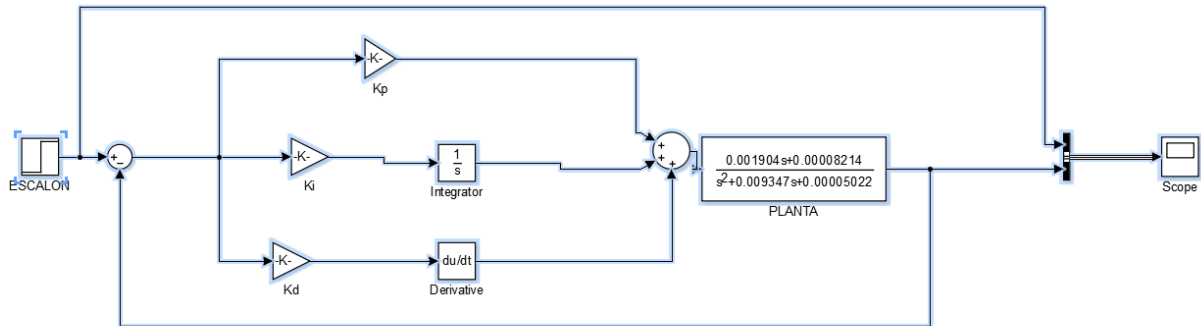


Figura 3.19. Diagrama de bloques para control P, PI, PID de la planta de nivel.

3.7.1 Respuesta Control Proporcional.

Para un control netamente proporcional las constantes K_i , K_d , deben estar en valor cero.

Al ingresar los valores obtenidos en la **figura 3.13**, $K_p=0.58$, $K_i=0$, $K_d=0$, se observa que la respuesta de la planta a una entrada escalón logra estabilizarse en 101 segundos, cada que aumentamos el valor de K_p nuestra planta responde más rápido.

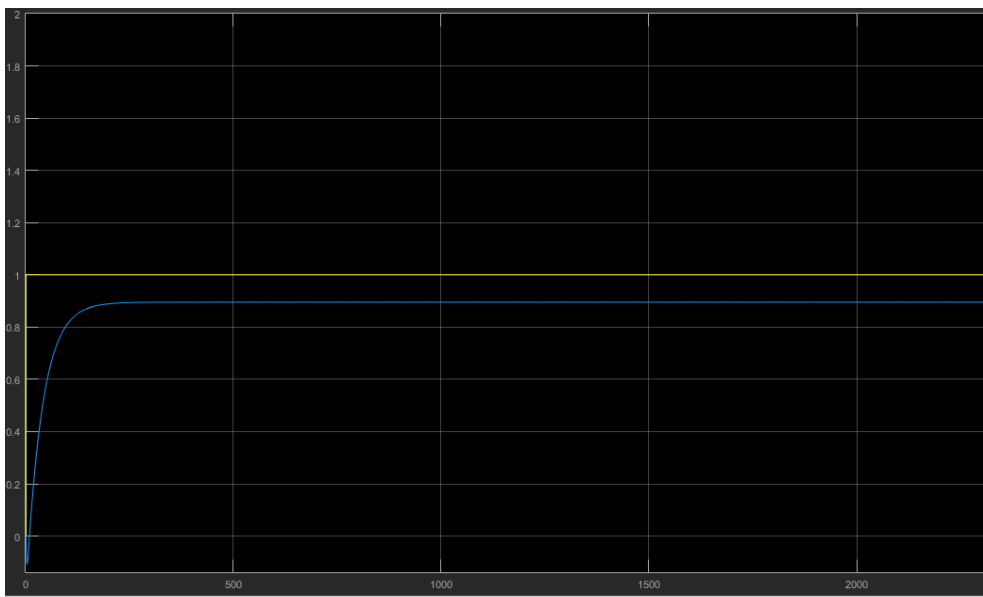
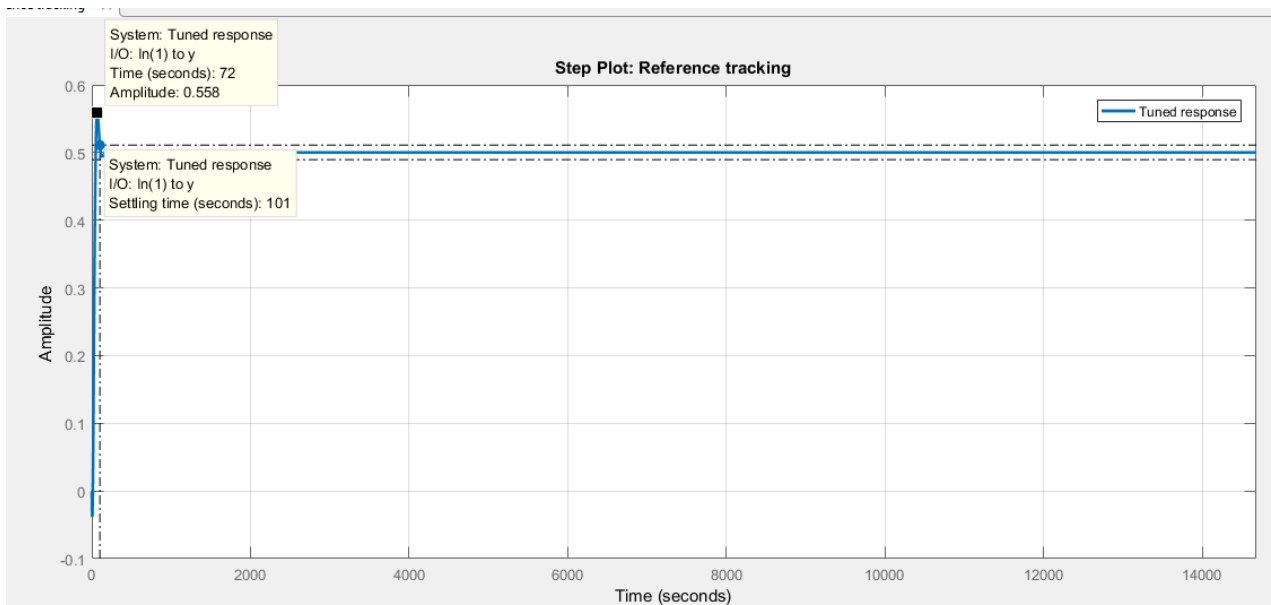


Figura 3.20. Respuesta de la planta a un control Proporcional.

3.7.2 Respuesta Control Proporcional Integral (PI).

Para la simulación de nuestro control PI con los valores obtenidos mediante PIDTOOL, se ingresó los siguientes valores mostrados en la **figura 3.15**, de sintonía a la planta $K_p=0.58$, $K_i=0.01$, $K_d=0$.

Donde se observa que el tiempo de asentamiento de la señal es 183 segundos aproximadamente y el valor pico es de 1.03, y es alcanzado en 9.09 segundos.

En este diseño podemos deducir que cumple con los requisitos planteamos.

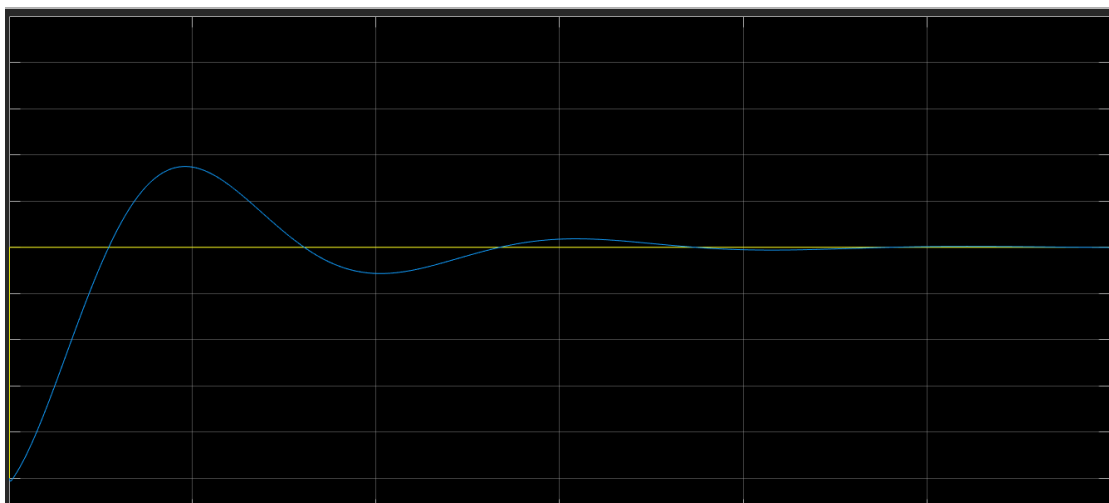
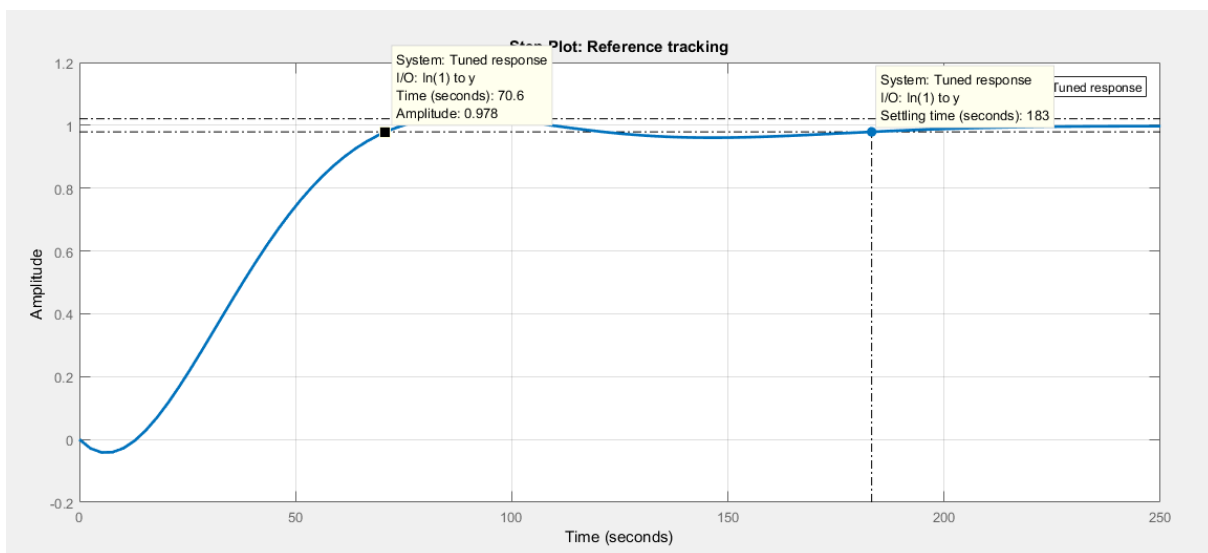


Figura 3.21. Respuesta de la planta de nivel al control Proporcional Integral.

3.7.3 Respuesta al control Proporcional Integral Derivativo (PID).

Al ingresar los valores de la **figura 3.17** en el diagrama de bloques de simulink se muestra los resultados de la **figura 3.22**, donde podemos observar que el tiempo de asentamiento de la señal es aproximadamente 29.1 segundos.

El error en estado estacionario es cero en teoría cumple con los requisitos de diseño.

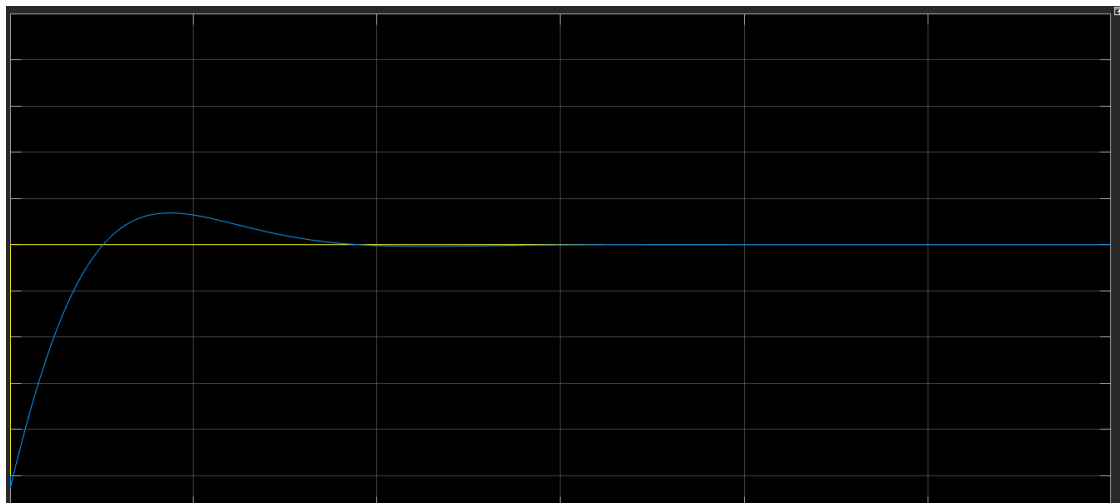
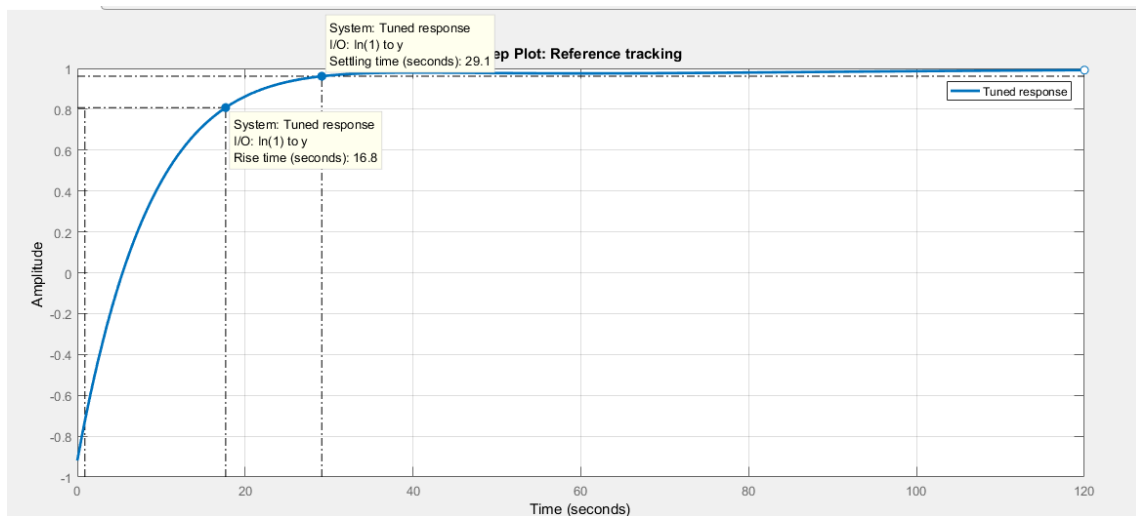


Figura 3.22. Respuesta de la planta de nivel al controlador PID.

3.8. Simulación del controlador utilizando LABVIEW

Luego de obtener las constantes de sintonía con el PIDTOOL de Matlab se procedió a simular los controladores implementados en la planta utilizando **'Labview'** y el módulo **'Control desing and simulation'** de National Instrument.

En la pantalla de animación **Figura 3.23** se puede observar de forma más real, el comportamiento del nivel del tanque al manipular las variables SET_POINT, Kp, Ki, Kd.

En las gráficas se puede observar el comportamiento de La señal escalón (color rojo) y la respuesta a la señal escalón (color azul),

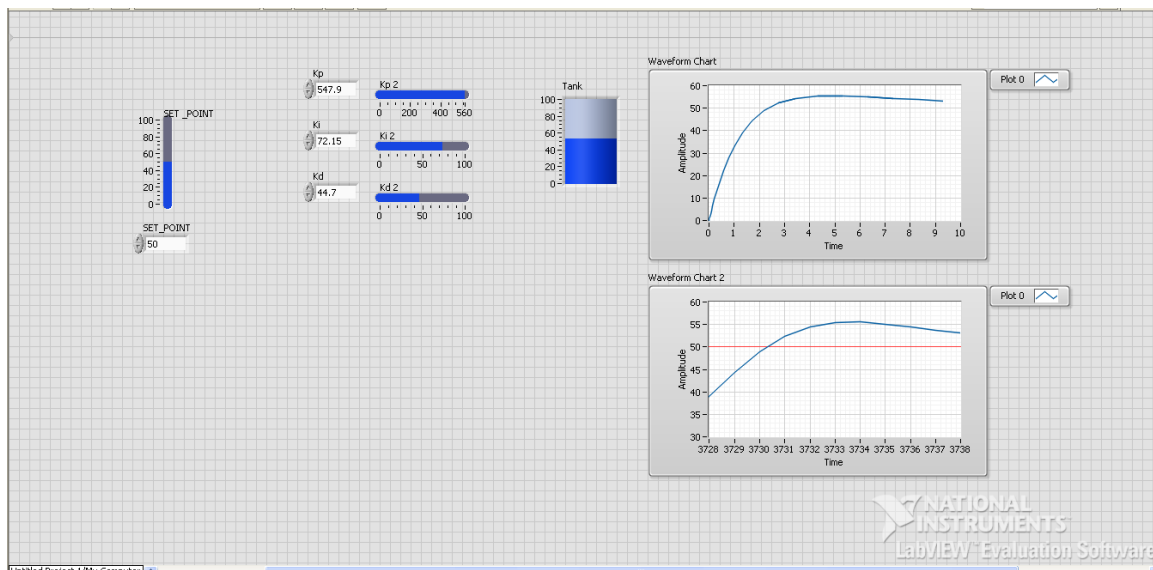


Figura 3.23. Panel principal de la simulación del sistema en LabView

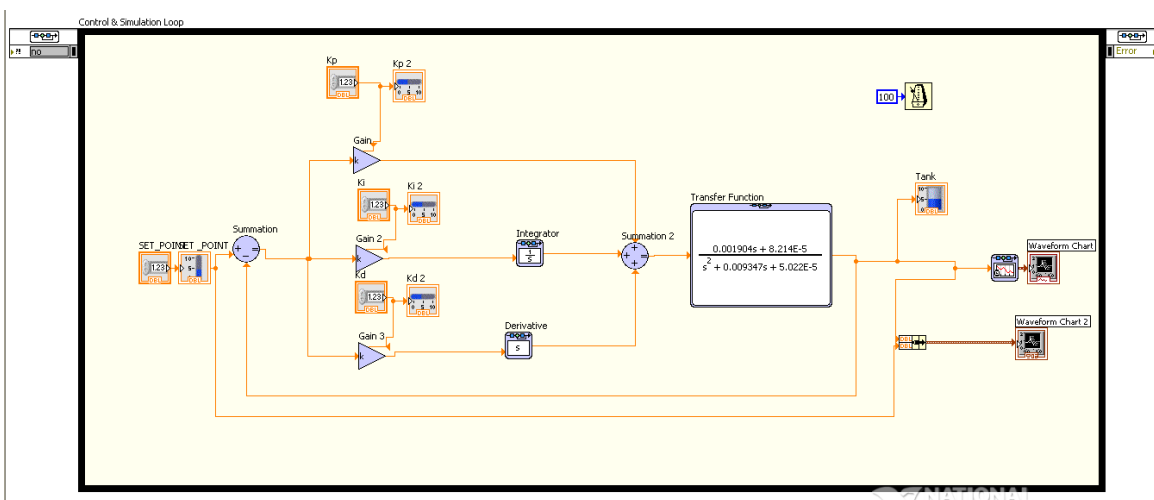


Figura 3.24: Diagrama de bloques para simulación de la función de transferencia.

CAPÍTULO 4

DISEÑO DEL SISTEMA SCADA

4.1. SCADA IGSS FREE

Para realizar un SCADA sin adquirir una licencia se utilizó el software IGSS V11 en forma free.

IGSS FREE50 es la versión libre del software SCADA IGSS V11 creado por Schneider Electric.

Esta versión libre dispone de todas las funcionalidades de la versión comercial pero está limitado al uso máximo de 50 objetos.

Este software SCADA cumple los requerimientos para aplicaciones en pequeñas y medianas aplicaciones, donde se puede leer y escribir un máximo de 50 variables de una planta industrial.

Con IGSS FREE50 se pueden crear proyectos con total funcionalidad

Algunas características que presenta son:

- ✓ Definir proyectos con un máximo de 50 objetos
- ✓ Recoger información de PLC sin límites temporales
- ✓ Permite usar más de 70 drivers para PLC
- ✓ También soporta OPC UA

Es posible actualizarlo a la versión comercial en cualquier momento.

Esta versión libre del SCADA IGSS también incluye proyectos pre-configurados que pueden ser utilizados en el modo Demo. Esta opción permite mostrar las capacidades del software sobre diversos entornos simulados.

IGSS FREE50 funciona bajo entornos Windows.

4.1.1 Alcance del Software.

El software se encargara de supervisar el proceso dentro del alcance de la red RS-485, con una animación muy amigable con el operador.

También se puede controlar y modificar los parámetros de la planta, almacenar los datos y presentarlos en reportes históricos, el cliente OPC y el servidor OPC estarán instalados en la misma PC, adicionalmente se configurará el control de acceso al software SCADA.

4.1.2 Conexión OPC

El cliente SCADA IGSS FREE50 cuenta con servidor OPC integrado al cual podemos acceder y configurar para recoger datos de cualquier PLC.

IGSS SCADA FREE50, soporta conexión con múltiples servidores OPC para leer y escribir variables en controladores industriales, sin límites de tiempo, como se mencionó anteriormente se puede leer y escribir hasta un máximo de 50 variables.

A continuación, se detalla paso a paso la configuración de IGSS FREE50 para comunicación por OPC, ya que en internet no podemos encontrar información detallada.

- a) Ejecutamos IGSS V11.0
- b) Clic en nuevo proyecto y nos aparece una ventana como la que se muestra en la figura 4.1, asignamos un nombre al proyecto y seleccionamos **siguiente**.

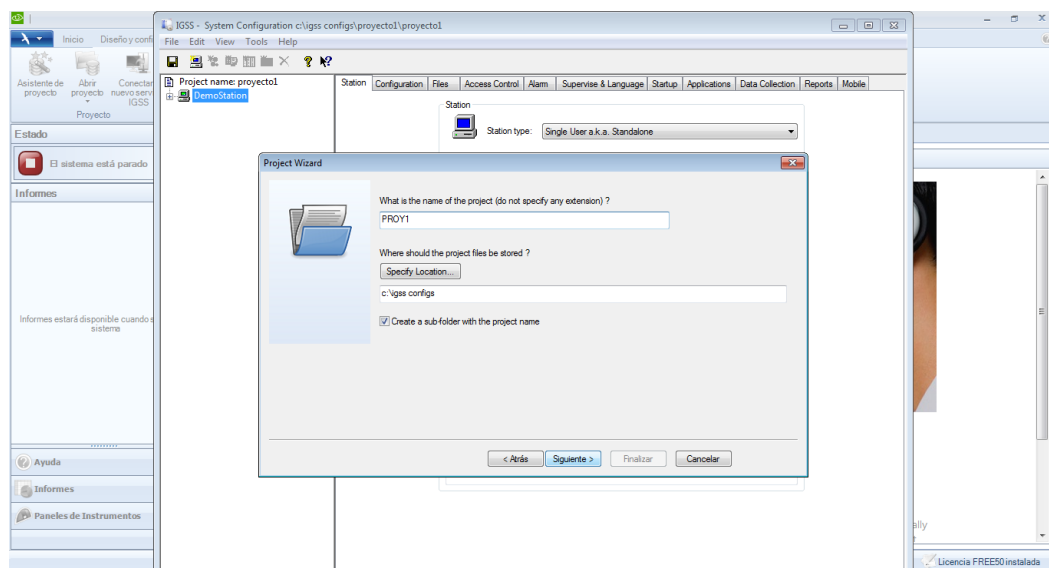


Figura 4.1. Ventana donde se muestra la configuración de nuevo proyecto en IGSS.

- c) Seleccionamos el driver a utilizar, para nuestro caso driver con ID 42 **OPC Client side driver.**

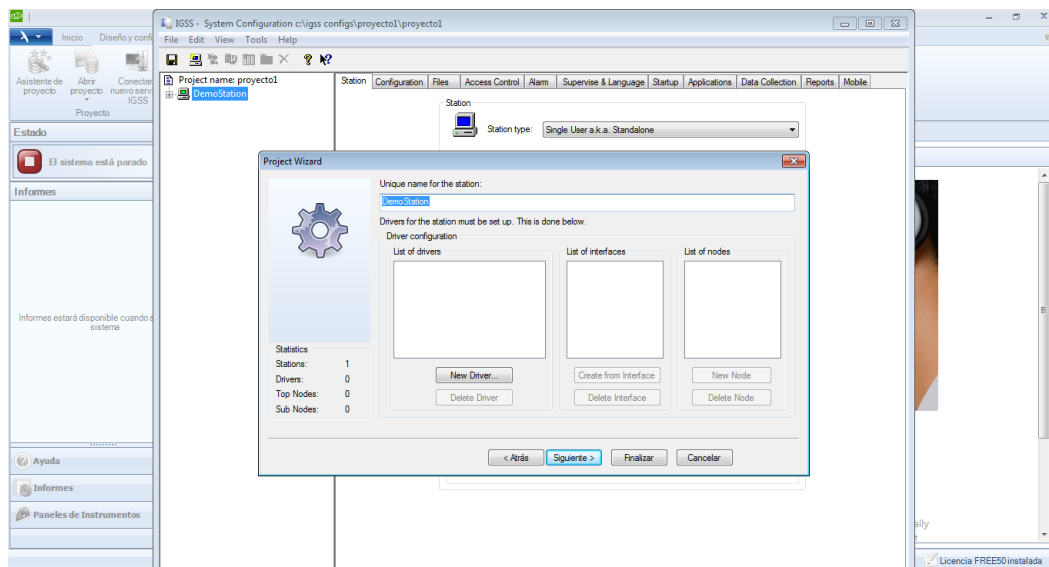


Figura 4.2. Ventana donde podemos seleccionar el driver para comunicación con cualquier PLC.

- d) En la figura 4.3 se muestra una ventana donde seleccionamos driver OPC Lista de interfaces y lista de nodos, luego de seleccionar el Driver, lista de interfaces y lista de nodos damos click en siguiente y luego finalizar.

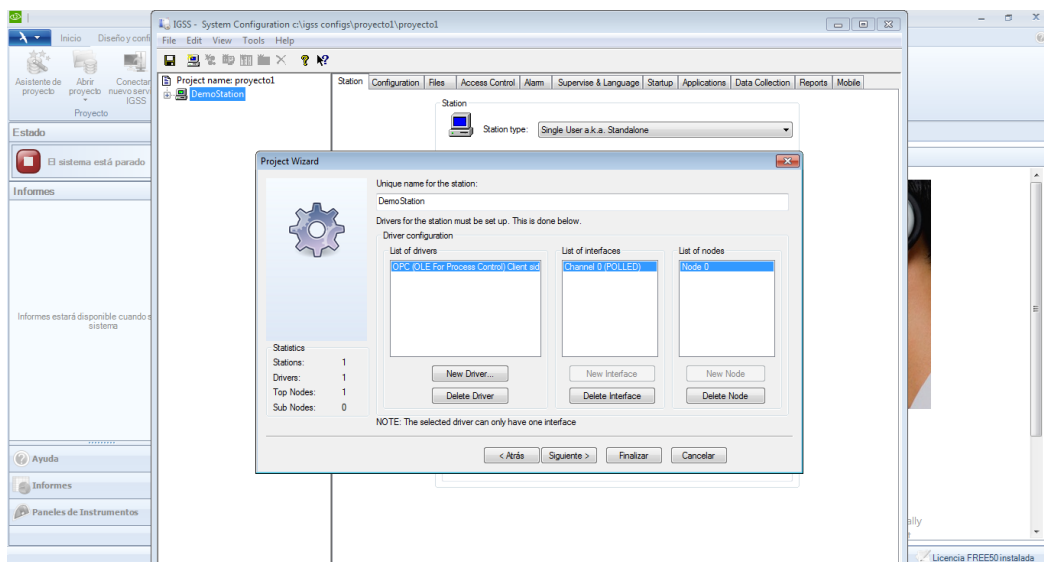


Figura 4.3. Driver seleccionado para la comunicación OPC.

- e) Ya en la pantalla de diseño podemos agregar el servidor OPC que se configuró para intercambiar datos con un PLC, en nuestro caso estamos utilizando KEPServerEX v5.

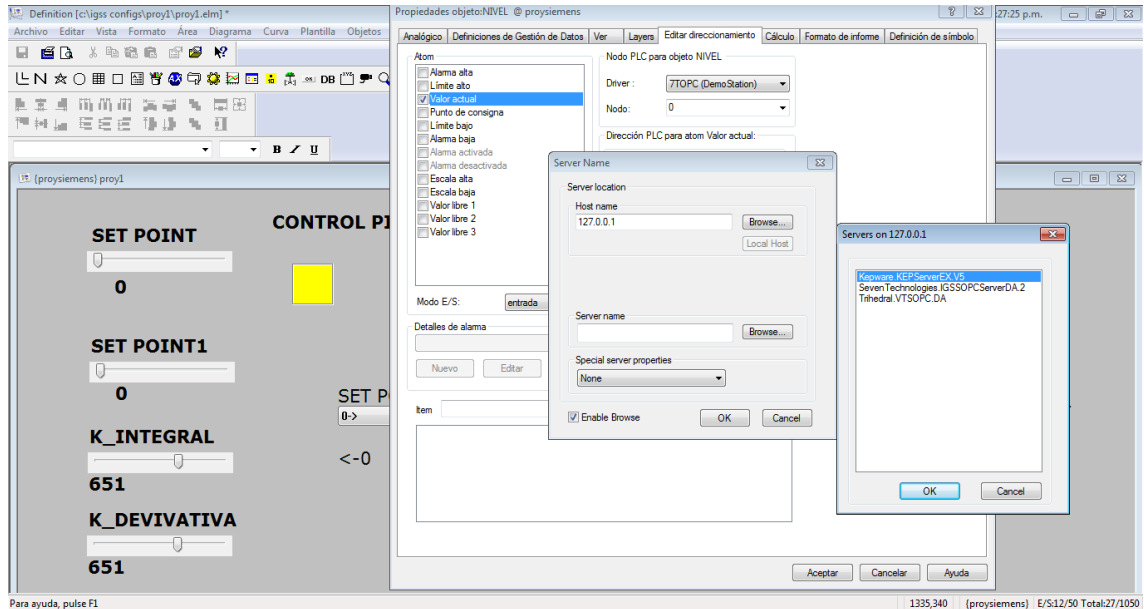


Figura 4.4. Ventana donde seleccionamos el servidor OPC para leer y escribir los datos al PLC.

4.1.3 Entorno gráfico del SCADA IGSS

- ✓ La figura 4.5 muestra la ventana principal de software IGSS.



Figura 4.5. Ventana principal del software IGSS.

- ✓ La Figura 4.6 muestra que la comunicación con PLC será mediante driver OPC.

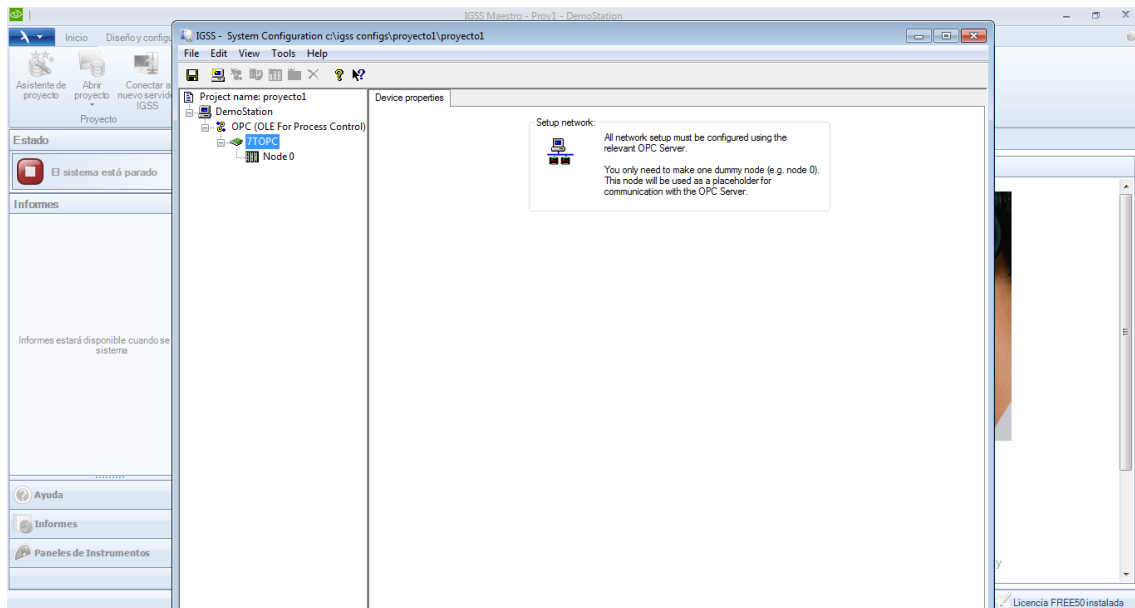


Figura 4.6. Configuración del Driver OPC para comunicación con PLC S7-200.

- ✓ La Figura 4.7 muestra la ventana principal del panel de instrumentos, donde se configura los tags y animaciones.

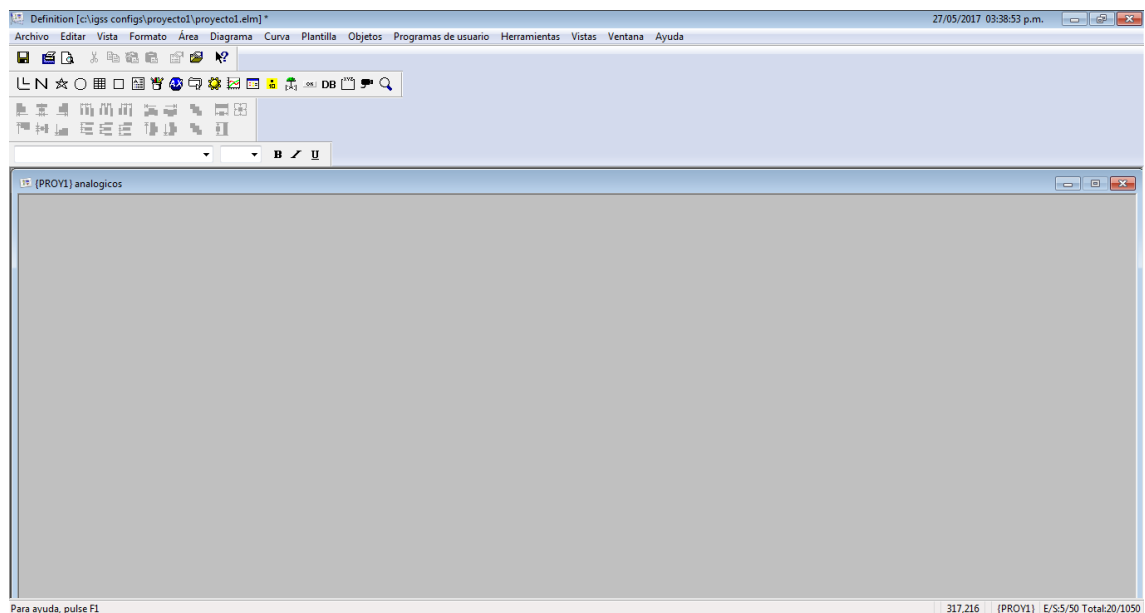


Figura 4.7. Ventana principal de del panel de instrumentos.

- ✓ La Figura 4.8 muestra los servidores OPC instalados en la PC que están disponibles para utilizar con el software IGSS SCADA.

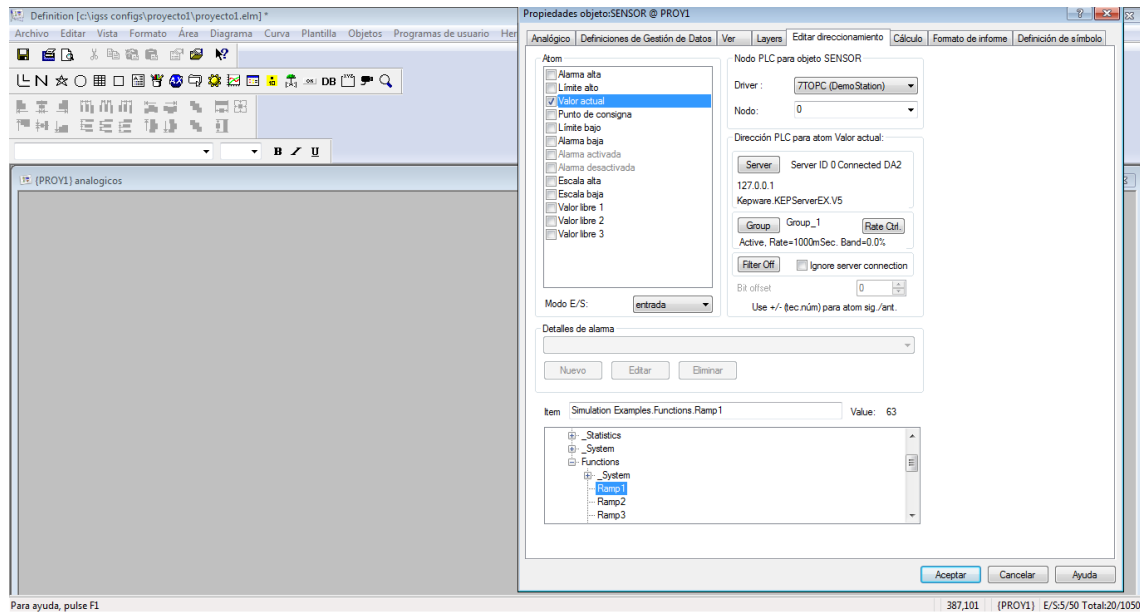


Figura 4.8. Servidores OPC instalados.

- ✓ En la figura 4.9 se creó un tag en el panel de diseño, al cambiar IGSS a modo Run ya puede ser monitoreado.

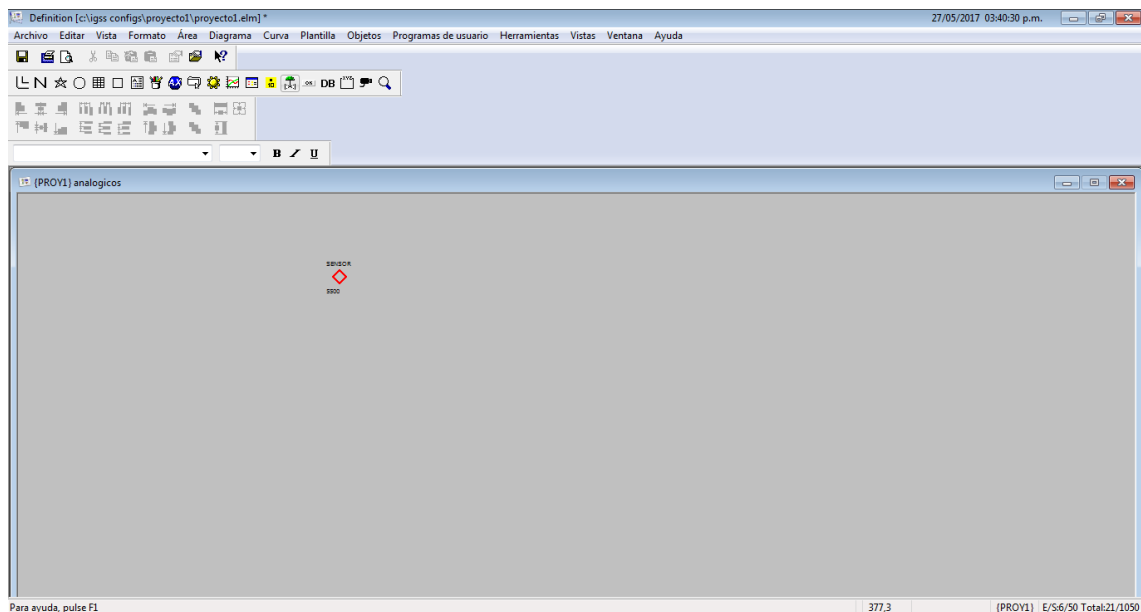


Figura 4.9. Comunicación entre SCADA y OPC.

- ✓ Figura 4.10 muestra el diagrama de tuberías para diseñar procesos industriales

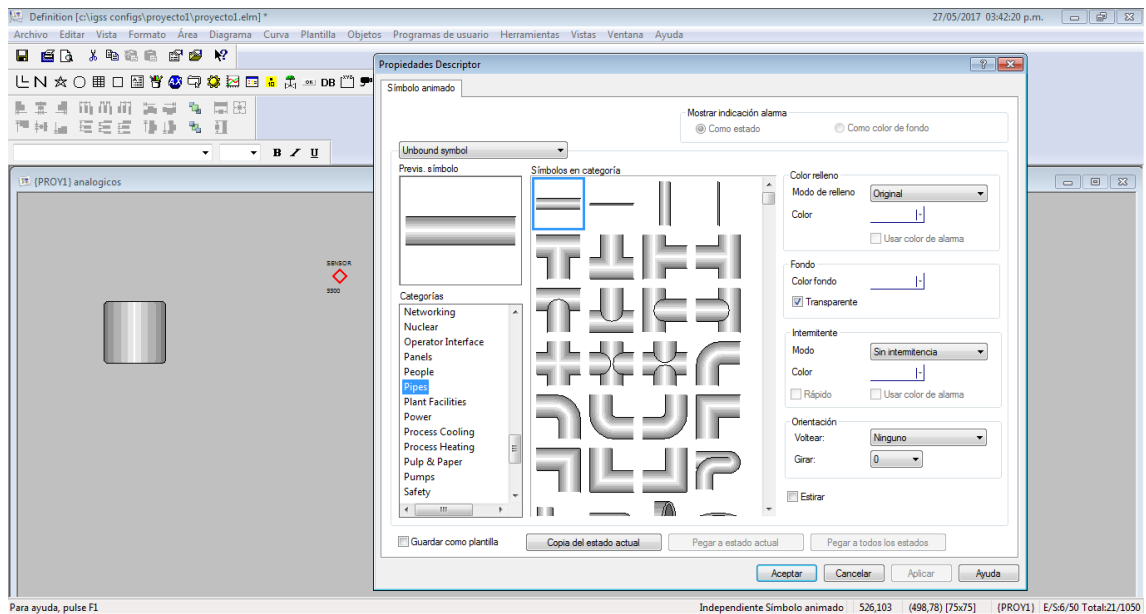


Figura 4.10. Diagrama de tuberías.

- ✓ Figura 4.11 se muestra panel de instrumentos la siguiente animación se utilizara para mostrar el estado del variador de frecuencia.

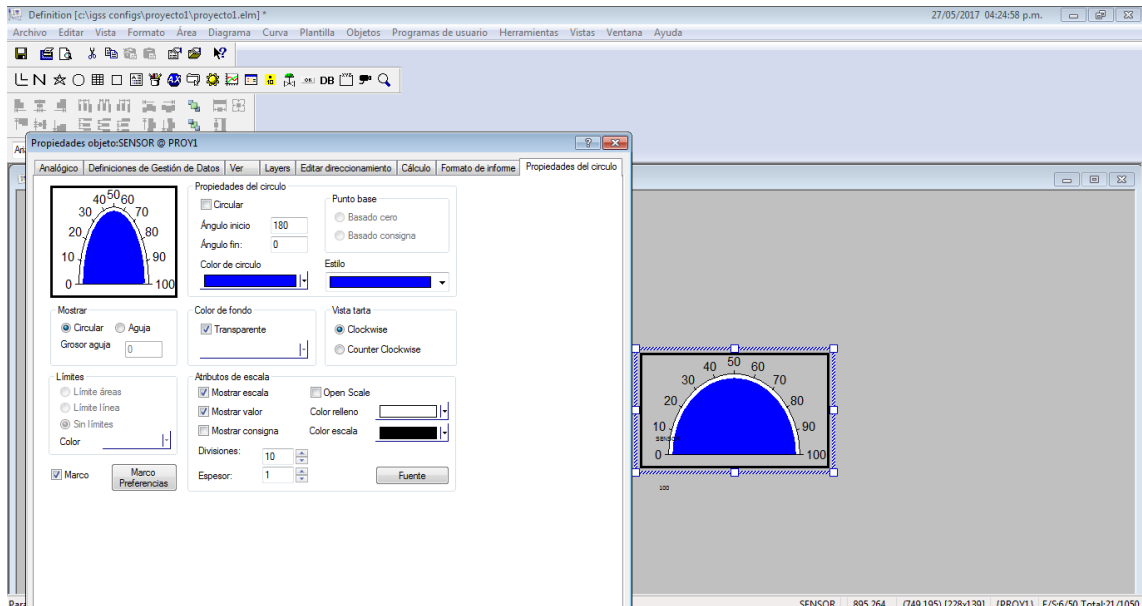


Figura 4.11. Visualizador Circular.

- ✓ En la Figura 4.12 se muestra los distintos elementos que se puede seleccionar para nuestro diseño tanques, válvulas, motores, etc.

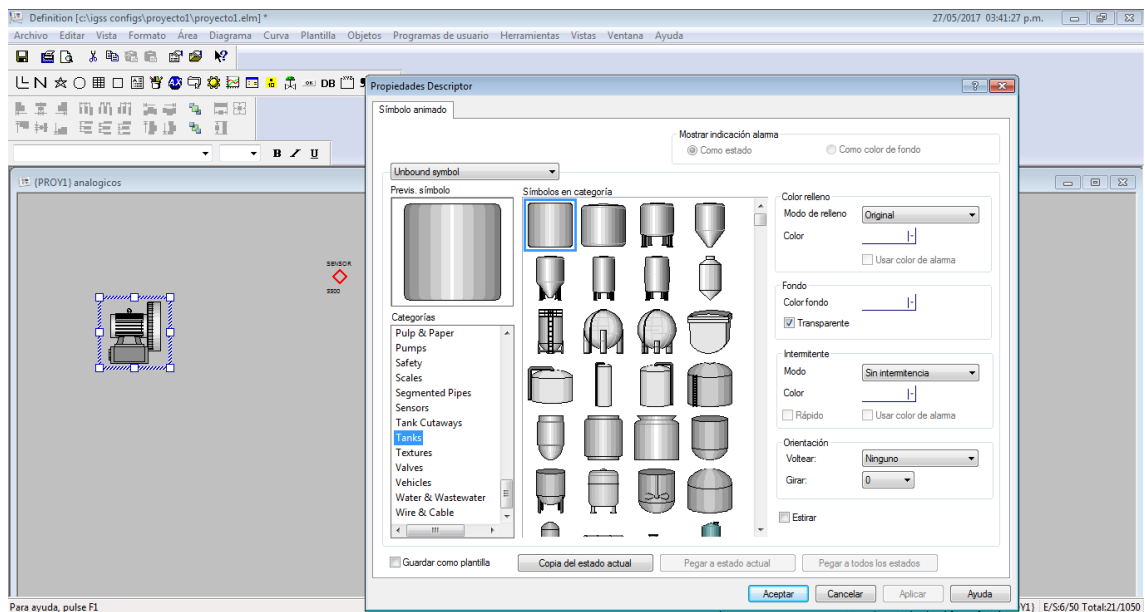


Figura 4.12. Imágenes de tanques para utilizar en animaciones.

- ✓ En la figura 4.13 se muestra algunos elementos que se utilizaran para el proyecto de PID para control de nivel.

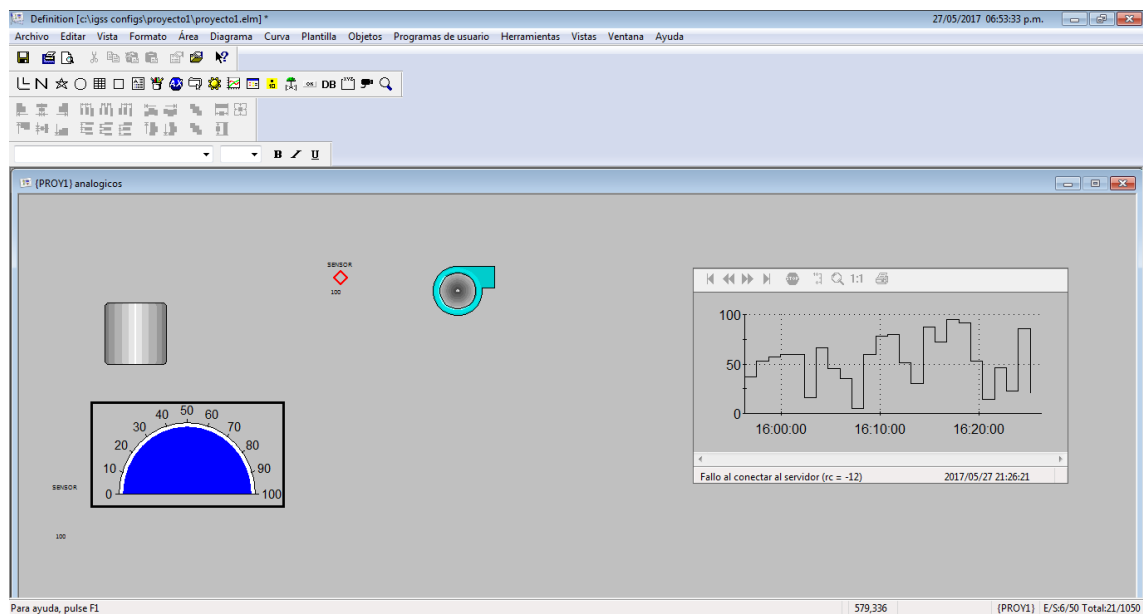


Figura 4.13. Elementos utilizados en el PID.

CAPÍTULO 5

IMPLEMENTACIÓN DE CONTROLADORES P, PI, PID

5.1 Programa final en el PLC S7-200.

Utilizando el software MICROWIN STEP7 se programó el PLC S7-200 para que realice control P, PI, PID en la planta de nivel.

A continuación se muestra la secuencia realizada en lenguaje Ladder, que permite controlar las constantes Kp, Ki, Kd, set point, leer variables como nivel del tanque, velocidad de la bomba.

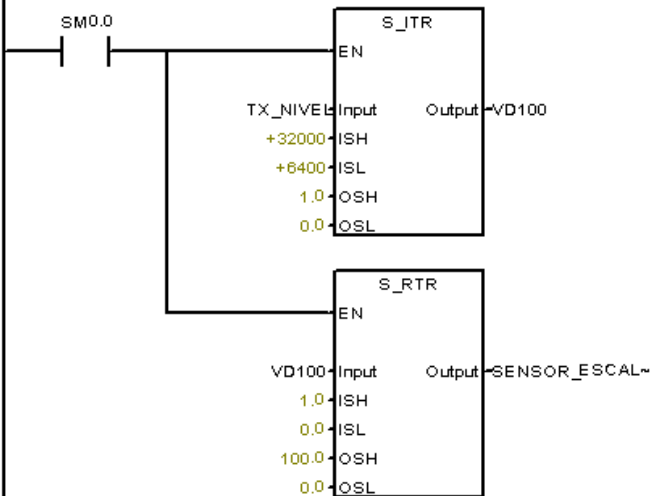
Tabla de símbolos				
· 3 · · 4 · · 5 · · 6 · · 7 · · 8 · · 9 · · 10 · · 11 · · 12 · · 13 · · 14 · · 15 · · 16 · · 17 · · 18 · ·				
		Símbolo	Dirección	Comentario
1		M_VARIADOR	VD501	MEMORIA PARA LEER VARIADOR EN HMI
2		SALIDA_VARIADOR	MW20	MEMORIA PARA TRASFERENCIA DE PIDN HACIA EL VARIADOR
3		POTENCIOMETRO_ESCALADO	VD190	MEMORIA SET_POINT LOCAL Y REMOTO
4		POTENCIOMETRO	AIW0	POTENCIOMETRO PARA FIJAR SET POINT LOCAL
5		SELECTOR_LOCAL	I1.0	
6		KD_HMI	VD50	KD DESDE HMI
7		KI_HMI	VD132	KI DESDE HMI
8		KP_HMI	VD124	KP DESDE HMI
9		SET_POINT_HMI	MW30	SET POINT DESDE HMI
10		SELECTOR_REMOTO	I0.7	
11		STOP_HMI	M0.1	PARO DESDE HMI
12		STAR_HMI	M0.0	MARCHA DESDE HMI
13		SISTEMA_FUNCIONANDO	Q0.4	
14		ACTIVA_VDF	Q0.0	BIT QUE ACTIVA EL VARIADOR
15		STOP_LOCAL	I0.6	
16		STAR_LOCAL	I0.5	
17		SENSOR_ESCALADO	VD110	SENSOR ESCALADO
18		TX_NIVEL	AIW4	TRANSMISOR DE NIVEL
19				
20				
21				

Figura 5.1. Variables utilizadas para la implementación de los controladores.

COMENTARIOS DEL PROGRAMA

Network 1 Título de segmento

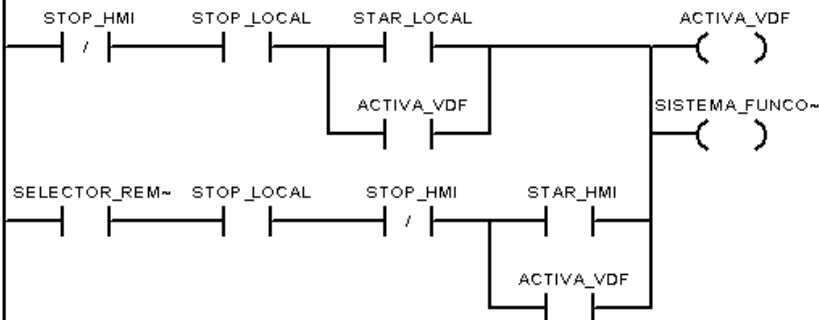
ESCALAMIENTO DEL SENSOR



Símbolo	Dirección	Comentario
SENSOR_ESCALADO	VD110	SENSOR ESCALADO
TX_NIVEL	AMV4	TRANSMISOR DE NIVEL

Network 2

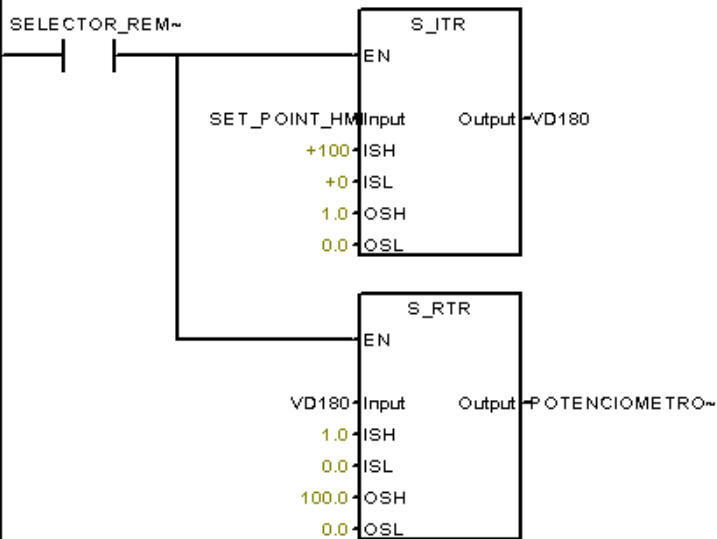
MARCHA Y PARO DE FORMA LOCAL Y REMOTA



Símbolo	Dirección	Comentario
ACTIVA_VDF	Q0.0	BIT QUE ACTIVA EL VARIADOR
SELECTOR_REMOTO	I0.7	
SISTEMA_FUNCIONANDO	Q0.4	
STAR_HMI	M0.0	MARCHA DESDE HMI
STAR_LOCAL	I0.5	
STOP_HMI	M0.1	PARO DESDE HMI
STOP_LOCAL	I0.6	

Network 3

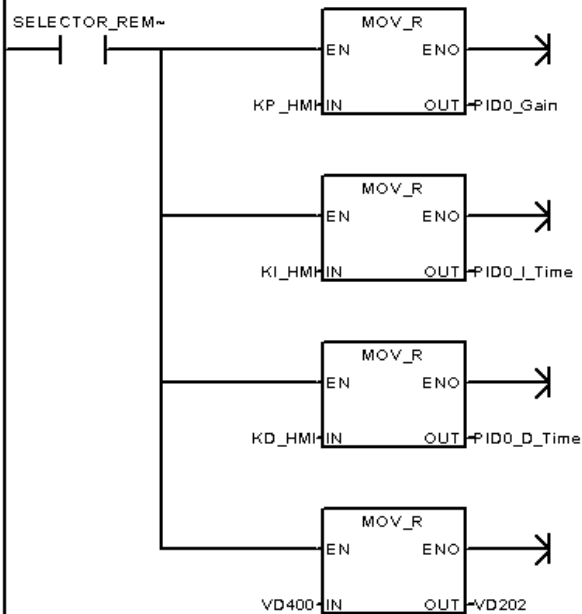
CAMBIO DEL SET POINT DESDE EL HMI



Símbolo	Dirección	Comentario
POTENCIOMETRO_ESC ALADO	VD190	MEMORIA SET_POINT LOCAL Y REMOTO
SELECTOR_REMOTO	I0.7	
SET_POINT_HMI	MW30	SET POINT DESDE HMI

Network 4

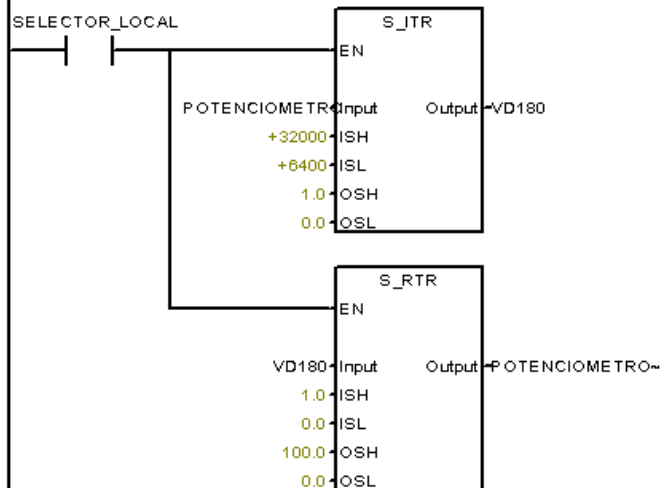
CAMBIO DE CONSTANTES DE SINTONIA DESDE HMI



Símbolo	Dirección	Comentario
KD_HMI	VD50	KD DESDE HMI
KI_HMI	VD132	KI DESDE HMI
KP_HMI	VD124	KP DESDE HMI
PID0_D_Time	VD138	Tiempo de acción derivativa
PID0_Gain	VD126	Gainancia del lazo
PID0_Time	VD134	Tiempo de acción integral
SELECTOR_REMOTO	I0.7	

Network 5

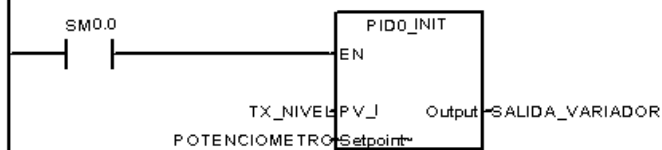
ESCALAMIENTO DE LA SE;AL SET_POINT DE FORMA LOCAL



Símbolo	Dirección	Comentario
POTENCIOMETRO	AMV0	POTENCIOMETRO PARA FIJAR SET POINT LOCAL
POTENCIOMETRO_ESC	VD190	MEMORIA SET_POINT LOCAL Y REMOTO
ALADO		
SELECTOR_LOCAL	I1.0	

Network 6

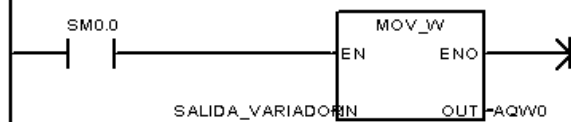
BLOQUE PID



Símbolo	Dirección	Comentario
POTENCIOMETRO_ESC	VD190	MEMORIA SET_POINT LOCAL Y REMOTO
ALADO		
SALIDA_VARIADOR	MMV20	MEMORIA PARA TRASFERENCIA DE PIDN HACIA EL VARIADOR
TX_NIVEL	AMV4	TRANSMISOR DE NIVEL

Network 7

TRANSFERENCIA AL VARIADOR



Símbolo	Dirección	Comentario
SALIDA_VARIADOR	MMV20	MEMORIA PARA TRASFERENCIA DE PIDN HACIA EL VARIADOR

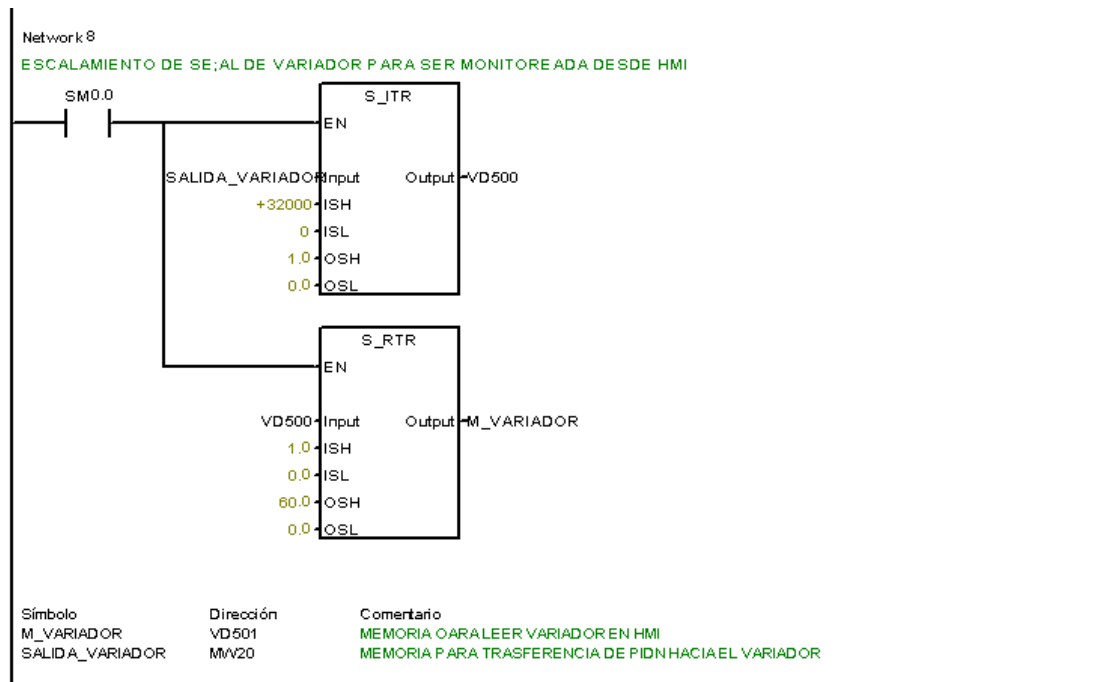


Figura 5.2. Programación final en Ladder.

5.2. Creación de las variables en el OPC.

Para realizar el control y monitoreo de forma remota de nuestra planta de nivel, se declara las variables en el servidor OPC KEPServer.

En la **figura 5.3** se muestra los tags que fueron configurados para leer y escribir en el PLC, memorias tipo booleanas, Word, float, byte.

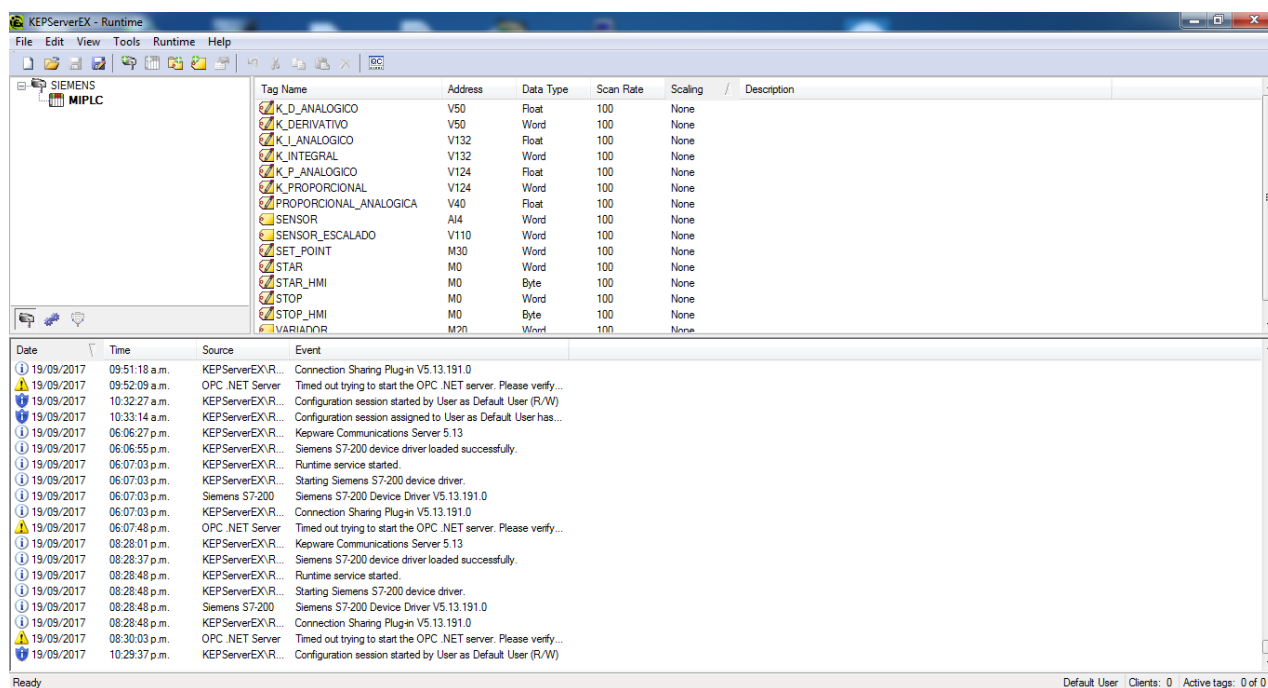


Figura 5.3. Vista de los tags configurados en el OPC.

5.3 Programación en LABVIEW de los controladores.

Para verificar el correcto funcionamiento del SCADA con software libre en primer lugar se desarrolló una aplicación en LabView para monitorear y controlar el PID de la planta de nivel.

La respuesta de la planta al SCADA con Labview debe ser similar al SCADA implementado en el software libre, ya que estas están enlazadas con las mismas direcciones del PLC S7-200 que controla la planta de nivel.

La aplicación consiste en dar orden de marcha y paro desde Labview y luego fijar el SET_POINT, ingresar nuestras constantes de sintonía obtenidas con Matlab de acuerdo al tipo de control a implementar ya sea P, PI, PID.

En el histograma se puede visualizar la gráfica Nivel Vs Tiempo y SET_POINT vs tiempo.



Figura 5.4. Panel frontal de programación en labView del SCADA para la planta de nivel.

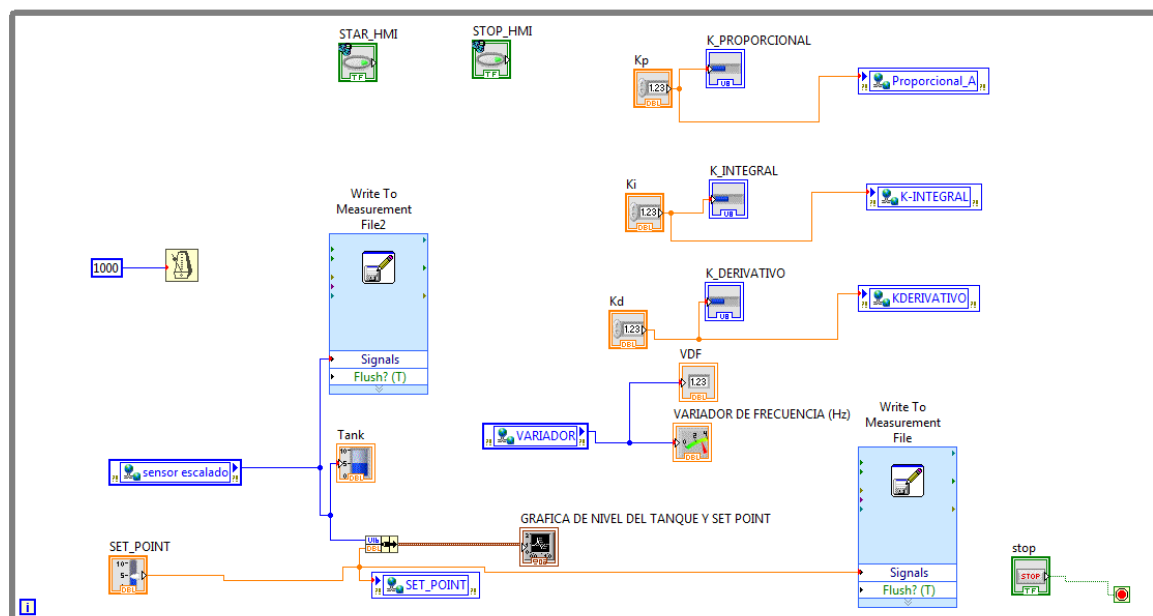


Figura 5.5. Diagrama de bloques del SCADA en LabView y enlazadas con sus respectivas direcciones en el PLC S7-200.

5.4. Creación de las variables en el SCADA libre.

Una vez creadas las variables en el servidor OPC y verificar la conectividad OPC-PLC, se procede a enlazar las variables del OPC con el SCADA.

- a) Creamos la variable para visualizar el estado del sensor de nivel, click derecho en la pantalla de diseño, opción nuevo-analog elements, le asignamos un nombre y lo enlazamos con la variable S-ESCALADO declarada en el servidor OPC, de esta manera ya podemos monitorear el estado del sensor de nivel.

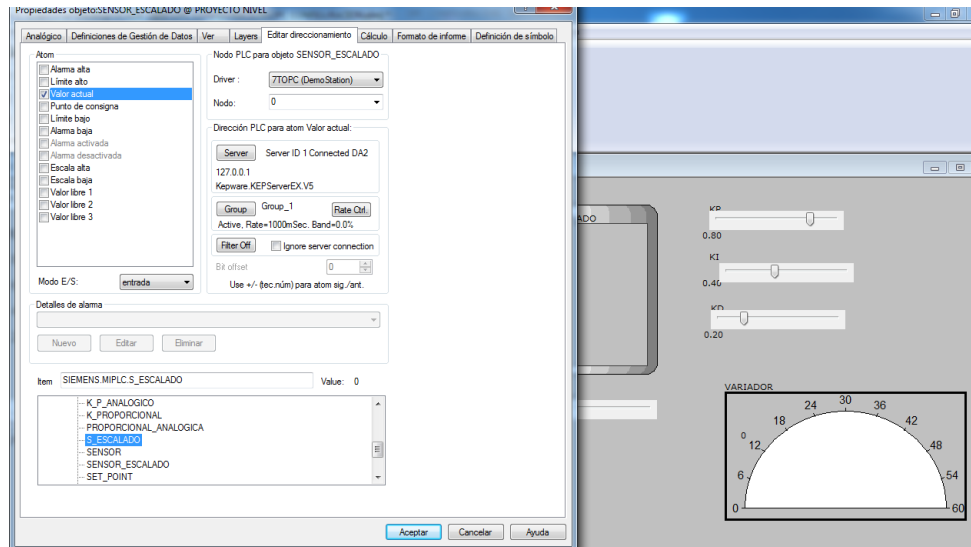


Figura 5.6. Creación de variables en el SCADA.

- b) Para monitorear el variador de frecuencia repetimos los pasos de la parte "a", pero esta vez enlazamos con la variable OPC VARIADOR.

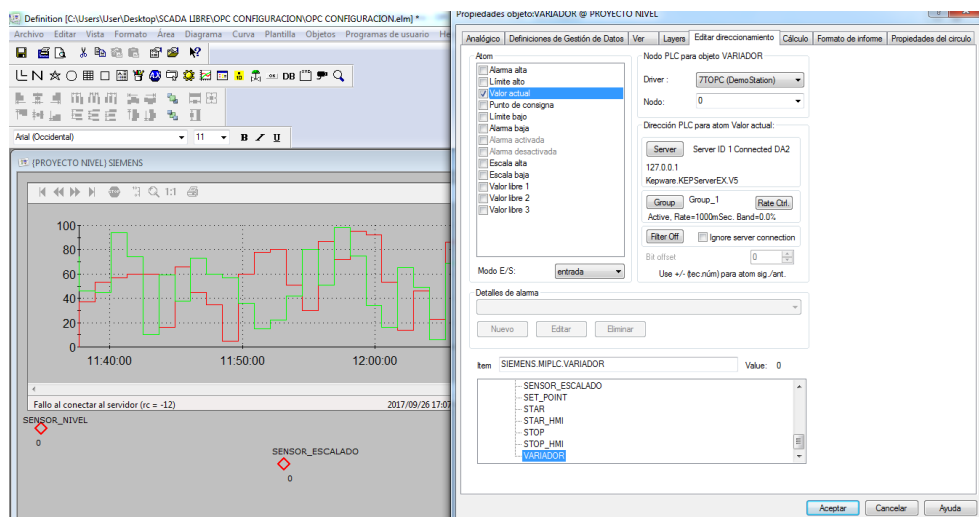


Figura 5.7. Enlazar variables con OPC.

- c) Para agregar los controles SET_POINT, Kp, Kd, Ki, se da click derecho en la pantalla de diseño y seguimos los pasos de la **figura 5.8**, luego los conectamos con sus respectivos TAGS OPC.

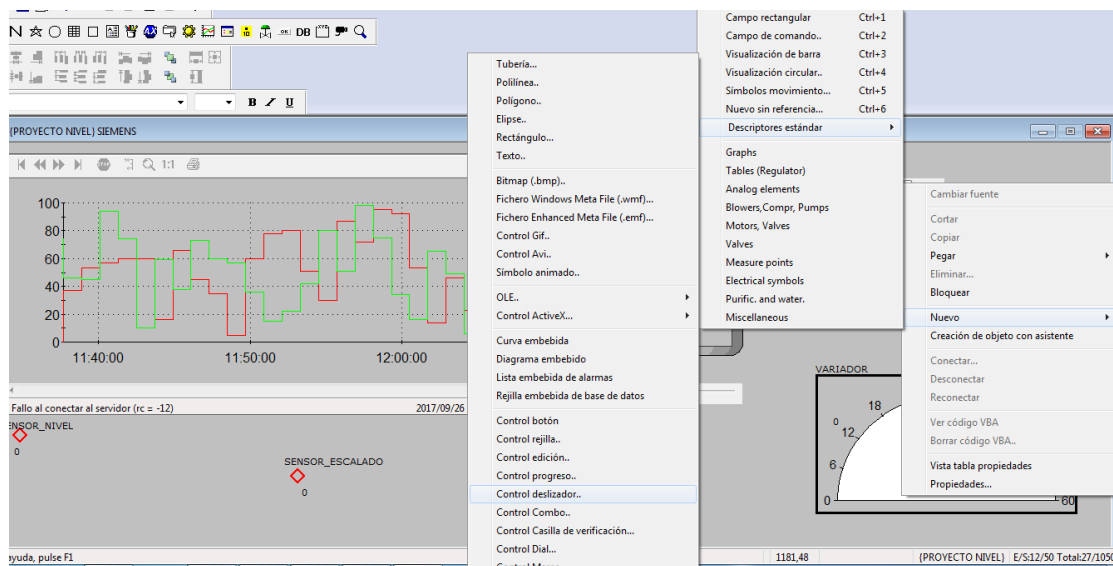


Figura 5.8. Creación de Set point y contantes de sintonía.

- d) Para visualizar una animación de la frecuencia del variador se conectó a un visualizador circular como se muestra en la figura 5.9.

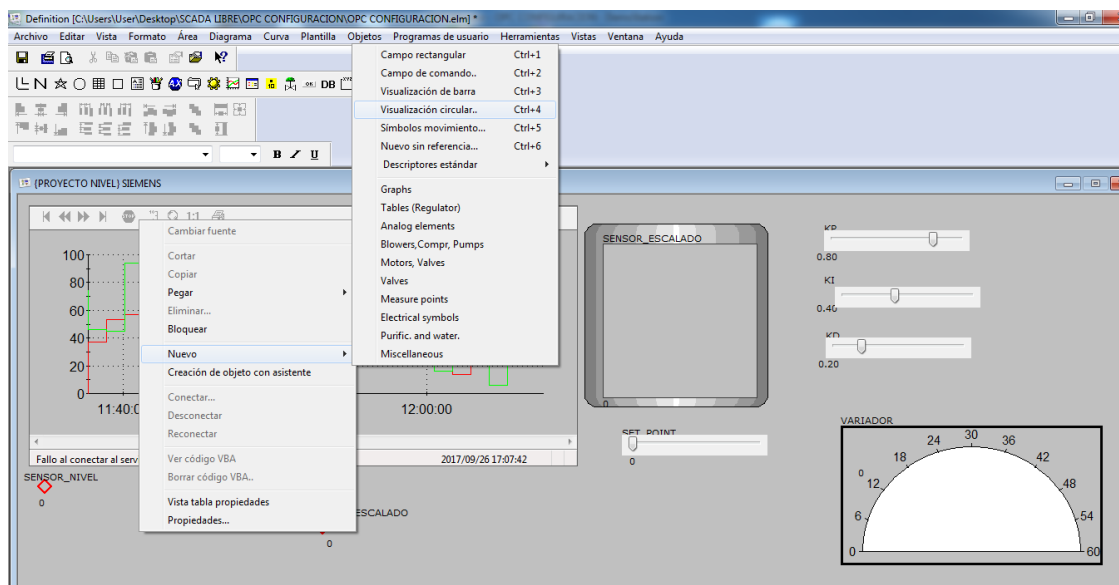


Figura 5.9. Configuración del visualizador circular y conectado al tag VARIADOR.

- e) Para agregar la función grafica de históricos se hace click en el icono “curva enbebida” luego le asignamos las variables que deseamos graficar en este caso SET POINT y sensor de nivel.

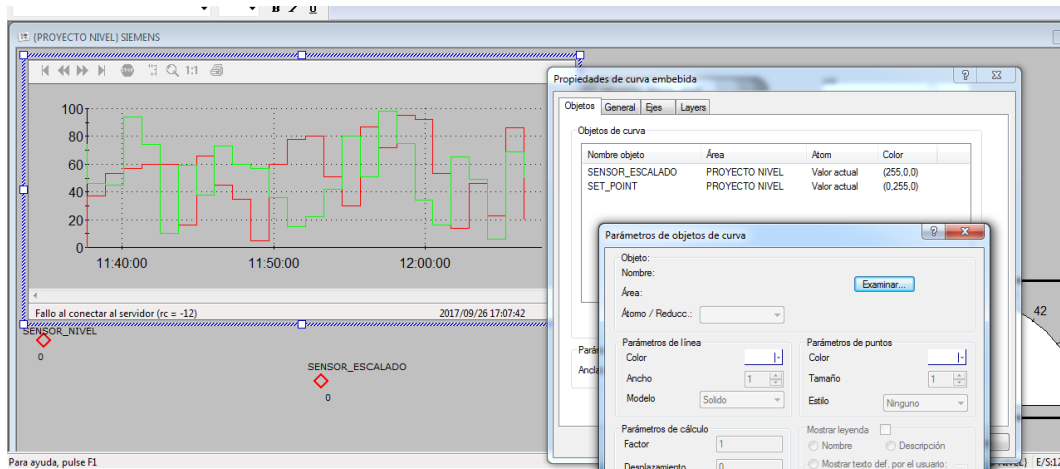


Figura 5.10. Configuración de la curva embebida para visualizar el comportamiento de la señal nivel.

- f) Para poner en marcha y paro la plata de nivel se utilizó “Botón de Windows”, los cuales conectamos con los TAGS de marcha y paro del OPC.

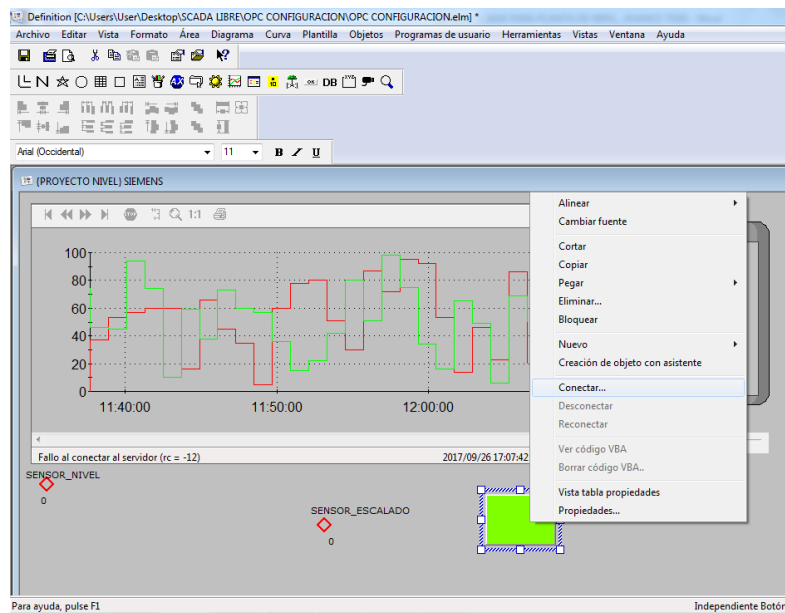


Figura 5.11. Agregar botones para marcha y paro.

- g) En la figura 5.12 se muestra el programa final desarrollado en IGSS SCADA para monitorear y realizar los tres tipos de control (P, PI, PID) de la planta de nivel del laboratorio de ingeniería Electrónica-UNPRG.

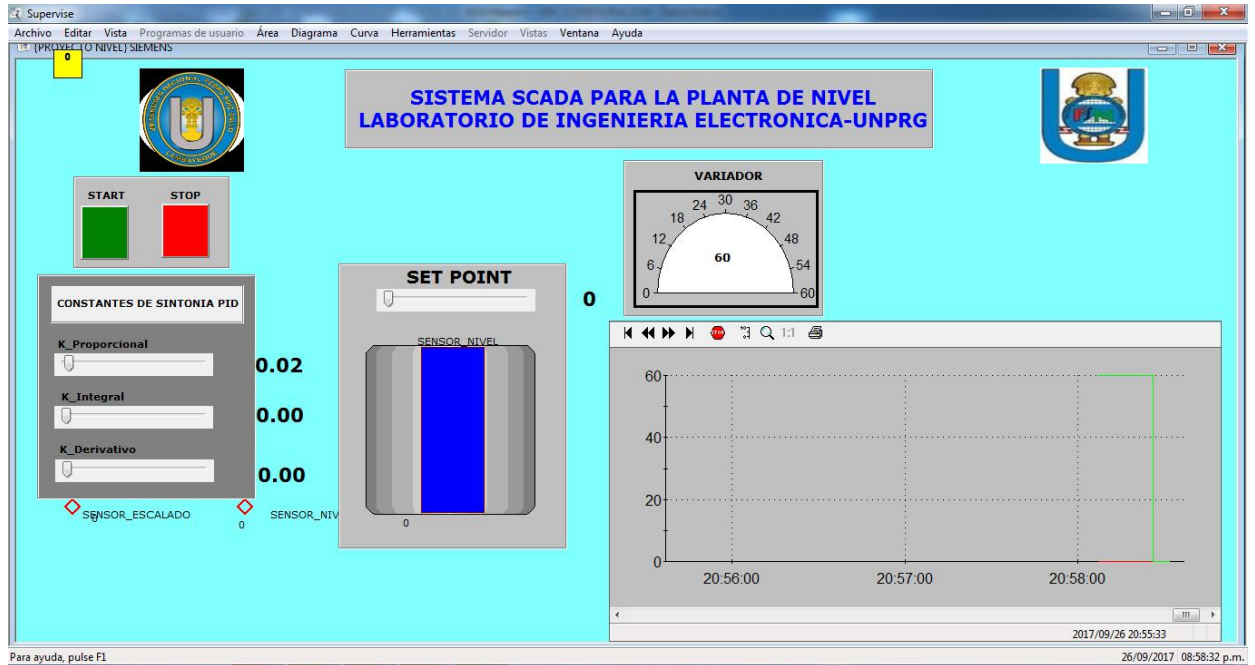


Figura 5.12. Imagen animación final SCADA libre desarrollado en IGSS.

CAPÍTULO 6

RESULTADOS EXPERIMENTALES

Se verificó el funcionamiento de los controladores Proporcional (P), Proporcional Integral (PI), Proporcional Integral Derivativo (PID), implementados en la planta de nivel del laboratorio de ingeniería electrónica-UNPRG.

6.1. Respuesta al Control Proporcional (P)

Para obtener un control netamente proporcional las constantes $K_i=0$, y $K_d=0$.

De la simulación del **capítulo 3** utilizando MATLAB se obtuvo para un control proporcional $K_p= 0.6$.

Se realizó pruebas con valores cercanos a 0.6.

En la **Figura 6.1** se puede observar un control proporcional con $K_p=0.2$, el set point es 20, la señal logra estabilizarse y el error de estado estacionario se aproxima a cero.

Esto se comprueba observando que la variable SENSOR_ESCALADO Y SET POINT tienen el mismo valor.

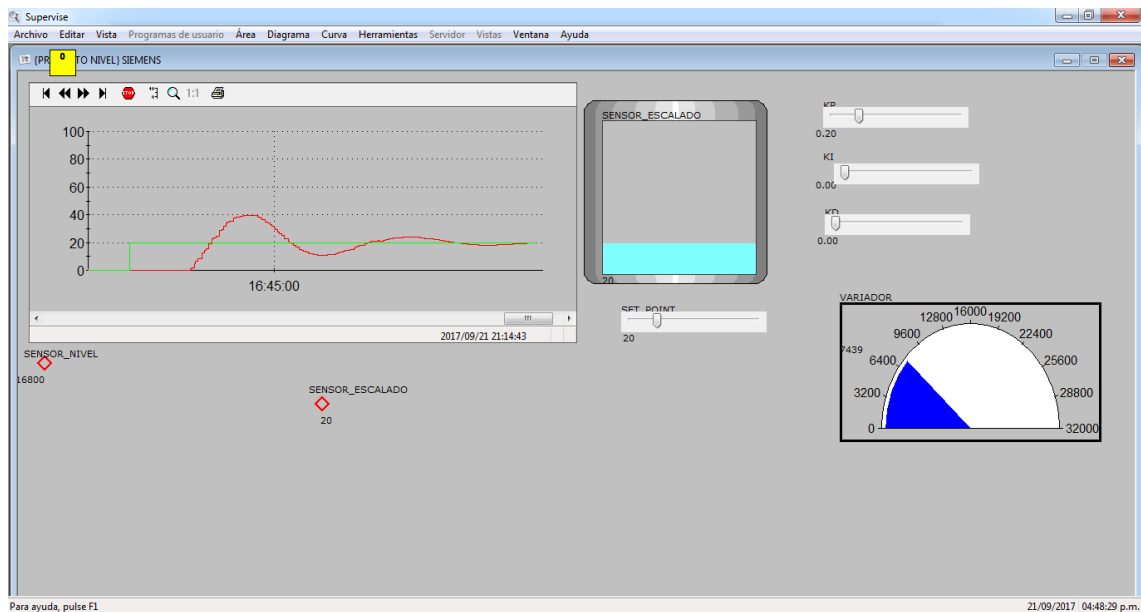


Figura 6.1. Control Proporcional con $K_p=0.2$.

Para $K_p=0.6$ la respuesta de la planta se observa en la Figura 6.2, responde de buena forma tal como se obtuvo en la simulación del capítulo 3.

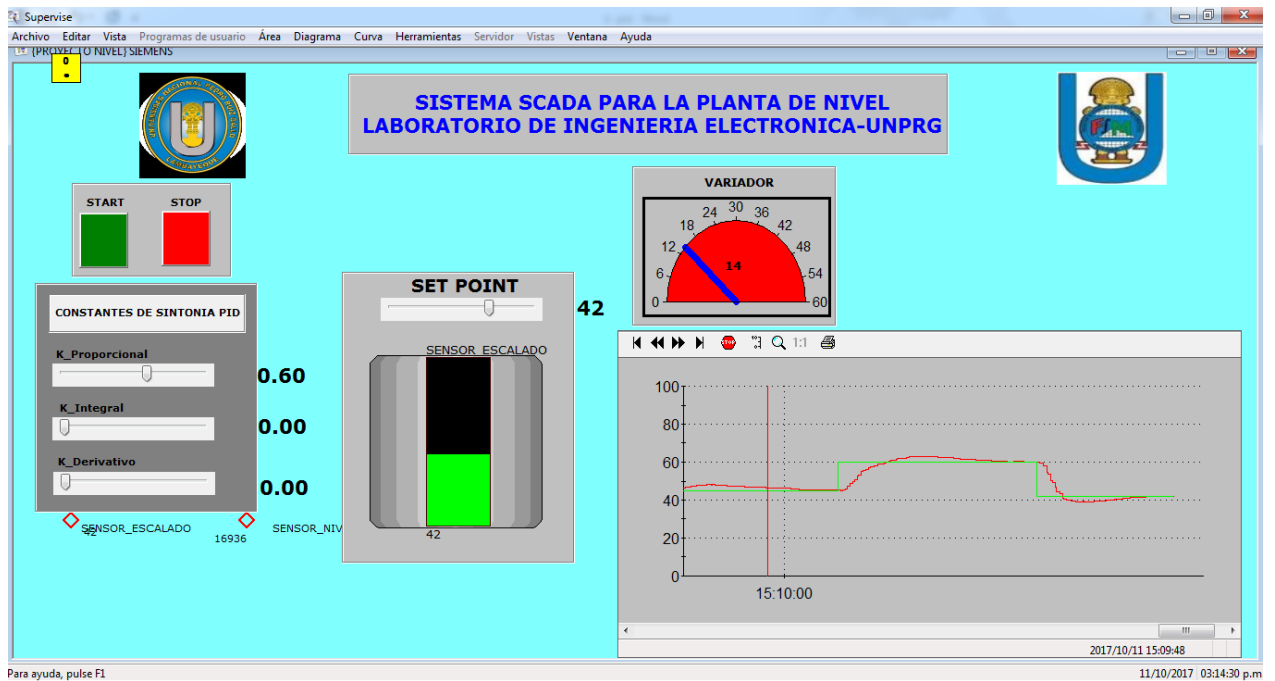


Figura 6.2. Respuesta de la planta de nivel a un control Proporcional $K_p=0.6$.

6.2. Respuesta al Control Proporcional Integral (PI)

Para que nuestro controlador sea de tipo PI, $K_d=0$ como lo podemos observar en la **figura 6.3**.

Donde:

$K_p=0.6$

$K_i=0.1$

$K_d=0$.

Estos datos se obtuvieron al modelar la planta de nivel en el Capítulo 3.

Al manipular la variable SET_POINT el sistema logra llegar al mismo valor de la señal de referencia (SET_POINT=47), progresivamente se fue aumentando el Set Point y la planta responde de buena manera al control PI.

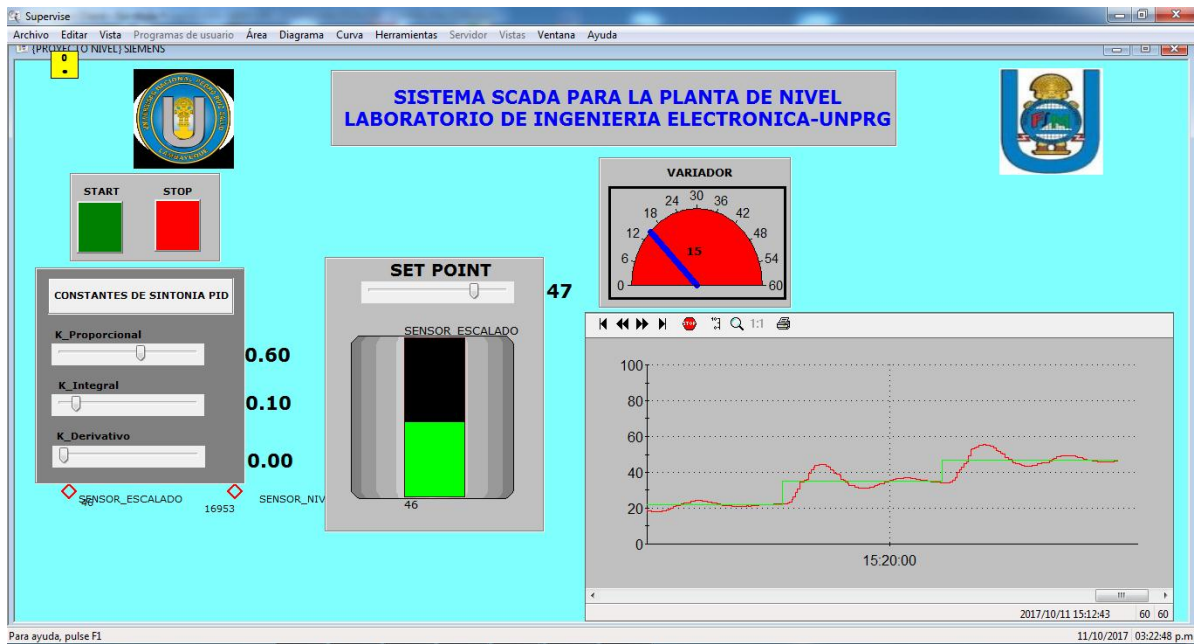


Figura 6.3. Control PI con $K_p=0.6$ y $K_i=0.1$.

Para una segunda prueba se utilizó:

$K_p=0.6$

$K_i=0.2$

$K_d=0$

En la figura 6.4 se observa la respuesta al controlador PI. Con error en estado estacionario cero (set point tiene el mismo valor que la variable de proceso).

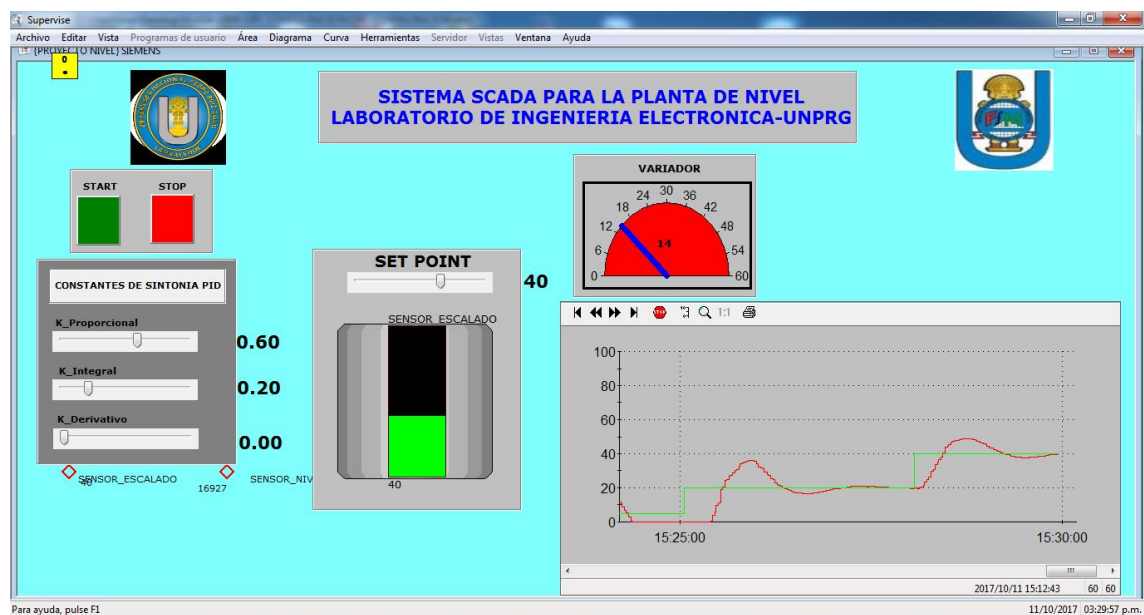


Figura 6.4. Control PI con $K_p=0.6$ y $K_i=0.2$.

6.3. Respuesta al Control Proporcional Integral Derivativo (PID).

Con las constantes de sintonía obtenidas con Matlab, en el capítulo 3 se nota que la planta no responde al control PID como se observa en la **figura 6.5** se manipula el SET_POINT pero la planta no responde, no logra controlarse.

Por lo tanto este tipo de controlador no se puede aplicar al control de nivel.

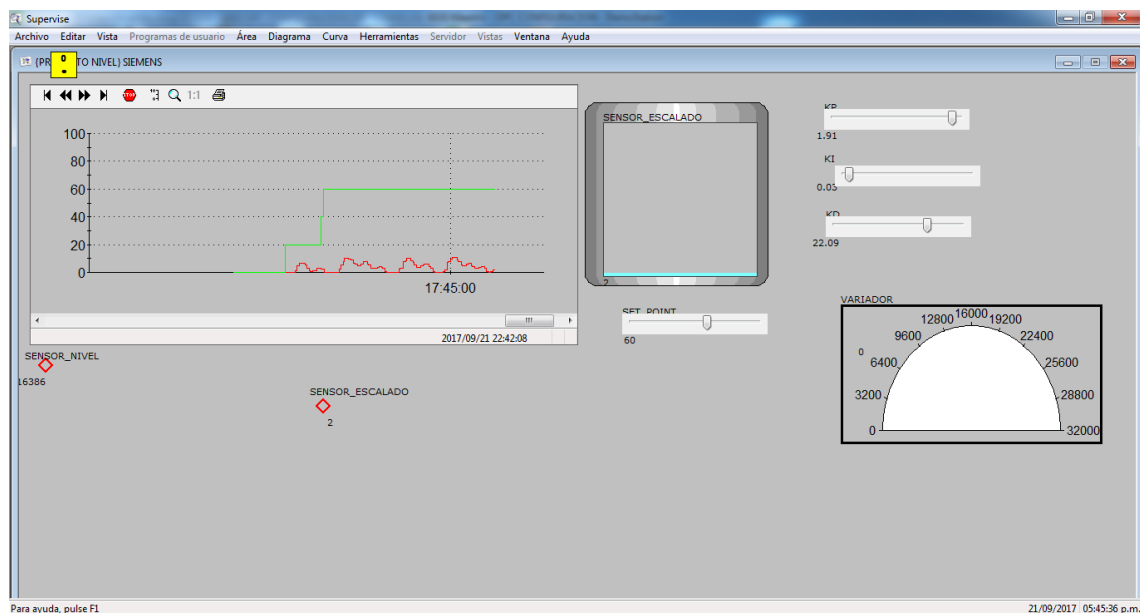


Figura 6.5. Respuesta de la planta al control PID

CONCLUSIONES

- Se logró realizar nuestro objetivo principal que fue diseñar un sistema SCADA con software libre, el cual se desarrolló con IGSS SCADA, para llegar a esto se probó con varios software libres.
- Se realizaron pruebas con LabView para verificar la conectividad de la planta de nivel con múltiples sistemas SCADA.
- Se logró realizar los objetivos secundarios propuestos, el diseño del control PROPORCIONAL funciona con $K_p=0.5$, cada vez que se aumenta K_p el sistema alcanza más rápido el set point pero tiene sobre picos más altos.
- En la simulación del control PROPORCIONAL se pudo notar que la respuesta a este no al cansaba la amplitud de la señal escalón, pero cuando se implementó en la planta real no se tuvo problemas.
- El control PI se realizó con $K_p=0.6$, $K_i=0.1$, el cual cumple los requisitos de diseño y tiene error de estado estacionario cero.
- El problema que tuvimos al realizar el control PID, es que la planta no respondía a nuestros valores obtenidos mediante la simulación de la función de transferencia por lo que llegamos a la conclusión que el control PID no es aplicable al control de nivel o podemos decir que la constante derivativa tiende a cero.
- Es suficiente con el control Proporcional y Proporcional Integral, con estos se puede realizar un control óptimo de la planta de nivel del laboratorio de Ingeniería Electrónica.

RECOMENDACIONES:

- Actualmente el sensor de presión para medir el nivel de agua se encuentra en la parte superior y central del tanque, debido a esto la medición está más propensa a errores debido al movimiento y el vórtice que se forma cuando cae el agua al depósito, por tal motivo se recomienda la reubicación del sensor a una de las partes laterales del tanque.
- Se debe utilizar un módulo de comunicación Ethernet para facilitar la comunicación del PLC S7-200 con otros dispositivos de red.
- Se recomienda el mantenimiento y uso de la válvula proporcional para ser utilizada como variable manipulada y no forzar mucho al Variador de frecuencia.
- SCADA IGSS cuenta con un servidor OPC incorporado, para desarrollar aplicaciones se recomienda utilizar el mismo OPC de SCADA IGSS.
- Para trabajar de manera adecuada con el software de programación y no tener problemas al momento de hacer la comunicación, se recomienda instalar en sistema operativo Windows XP.

BIBLIOGRAFÍA

1. Kuo, Benjamin C. Sistemas de Control Automatico. Departament of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, 2012.
2. Ogata, Katsuhiko. Ingeniería de Control Moderno. University of Minesota, 2010.
3. Vega Zuloeta, Javier, Piscoya Siaden, Alonso, Diseño e implementación de un sistema SCADA con PLC MODICON M340 para el control PID de la planta de presión de la escuela profesional de ingeniería electrónica de la universidad nacional Pedro Ruiz Gallo, utilizando software libre. 2015
4. Peña Devia, Fabián Eugenio, Poveda Ramírez Oscar, Diseño e implementación de una interface de control de Caudal e interfaz gráfica de usuario en planta didáctica del laboratorio de mecánica de la facultad tecnológica. 2015
5. <https://www.certs.es/blog/sistemas-control-software-libre>.
6. SIMATIC, Manual del sistema de automatización s7-200.
7. Rafael Villela, Aurelio Beltrán, Control de temperatura con PLC S7-200. 2007
8. <http://isa.umh.es/asignaturas/asc/practicas/ControlPIDAutomatas.PDF>.
9. <http://igss.schneider-electric.com/products/igss/download/free-scada.aspx>.
10. <http://igss.schneider-electric.com/products/igss/support/help-and-manuals-v11.aspx>.

ANEXO A

Sistemas de control de software libre

Publicado el 21/12/2015, por INCIBE



Las aplicaciones que conforman un sistema SCADA o DCS suelen venir integradas dentro de un mismo paquete de software, que incluye aplicaciones para crear el HMI, el servidor SCADA, histórico de datos, etc. Estas aplicaciones se instalan de forma individual y por separado dependiendo de las necesidades del sistema y los condicionantes que impone el desarrollador.

Históricamente, estos paquetes software han sido proporcionados por un único desarrollador en cada instalación, que ha trabajado de forma exclusiva para entornos industriales, creando en muchas ocasiones soluciones a medida para entornos concretos. Entre estos grandes fabricantes de paquetes software están Wonderware, Telvent, Honeywell, Siemens, etc.

La tendencia hacia la implantación del software libre en los sistemas corporativos y en los hogares, sobre todo por el ahorro de costes y el soporte ofrecido por los desarrolladores de las comunidades que suelen ser más activos a la hora de participar en el desarrollo de un proyecto. Esta tendencia también se ha trasladado hacia los sistemas de control, aunque con cierto retraso.

La característica principal del código libre es que no se patenta y se basa en licencias de tipo GPL, AGPL, Copyleft, etc., lo que otorga permisos para ejecutar, copiar, distribuir, estudiar, modificar el software e incluso redistribuirlo una vez modificado. No se debe confundir el software libre con el software gratuito, aunque muchas veces ambos términos confluyen. El beneficio de los desarrolladores de software libre proviene de donaciones y de los servicios asociados que proporcionan, como instalación, personalización o el soporte técnico.



-Iniciativa Open Source-

La iniciativa de software libre tiene como objetivo fomentar el desarrollo y colaboración entre desarrolladores dentro de comunidades. Todos ellos, podrán desarrollar libremente y serán poseedores de un acceso a todos los avances realizados, pudiendo modificar los mismos para mejorar los proyectos.

Sistemas operativos

Los sistemas operativos necesarios para la instalación de las aplicaciones que conforman los sistemas SCADA o DCS eran siempre propietarios, utilizando en muchos casos sistemas Windows, pero también otros como AIX, HP-UX, etc. La razón principal para la utilización de estos sistemas operativos era el respaldo por parte de una gran empresa (Microsoft, IBM, HP, etc.) y la posibilidad de disponer de soporte en cualquier momento en caso de incidente.

Estos sistemas operativos se siguen utilizando hoy en día, sobre todo por los principales desarrolladores de software de control, pero sistemas operativos gratuitos como Linux (en varias de sus distribuciones) o Solaris también se están comenzado a tener en cuenta por algunos desarrolladores.

Hay tres razones principales para apoyar los sistemas SCADA y DCS sobre sistemas operativos libres (principalmente Linux):

- Seguridad: Windows, debido a su amplia distribución es un objetivo principal de ataques. Aunque en menor medida, los sistemas Linux también reciben ataques, por lo que es importante no olvidar llevar a cabo una correcta configuración de los mismos.

- Estabilidad y confiabilidad: Aunque en este sentido Windows ha mejorado sensiblemente en sus últimas versiones, es de sobra conocida la estabilidad de los sistemas Linux, su mejor gestión de actualizaciones generalmente sin necesidad de reinicio contribuye a lograr una mayor disponibilidad. Esta razón unida a una mayor robustez y rendimiento de los sistemas Linux fortalece la idea de su utilización como sistema operativo para mejorar los tiempos de producción, reduciendo las caídas o pérdidas de integridad del sistema y, por tanto, los problemas de las industrias
- Menor coste: Además de que Linux es un sistema operativo gratuito, los costes de mantenimiento son inferiores incluso contratando un soporte.

Aplicaciones

Las aplicaciones necesarias para el control de un sistema SCADA o un DCS eran, en general, propietarias y requerían de una licencia para poder funcionar. En la mayoría de los casos, el software que controlaba una instalación no podía ser elegido por el cliente, sino que el propio fabricante de los dispositivos decidía cual utilizar. Esta imposición también fijaba el sistema operativo sobre el que tenían que funcionar, ya que los desarrolladores trabajan, de forma general, con un único sistema operativo para sus aplicaciones.

Hoy en día esta práctica sigue vigente en muchas instalaciones, pero en aquellas donde el entorno de dispositivos es heterogéneo y procede de distintos proveedores, el cliente dispone de mayor granularidad y control sobre los aplicativos que decide utilizar, lo que les permite también eliminar la dependencia a un sistema operativo. Las condiciones y restricciones que las aplicaciones propietarias imponían en los sistemas de control limitaban en gran medida la incorporación de nuevo software a las plantas. Pero cuando la dependencia con un único fabricante se fue relajando surgieron nuevas opciones, con nuevos desarrolladores dedicados exclusivamente al software y que han ido introduciendo paulatinamente diversos componentes libres en los sistemas de control.

Por otra parte, las necesidades globales de información, la aparición de nuevos dispositivos como teléfonos inteligentes o tabletas dentro de la industria, también han ayudado a la revolución del software industrial. Las nuevas tecnologías han propiciado que empresas de software comercial y grupos de usuarios se interesen también por los sistemas industriales.

Algunos ejemplos de software libre en la Industria

Los nuevos desarrollos de código abierto han surgido en muchas ocasiones de proyectos de investigación, aunque muchas empresas privadas también han desarrollado productos libres, habitualmente versiones limitadas de sus opciones comerciales.

A continuación se muestra un ejemplo de cada una de las diferentes opciones de software libre que se pueden encontrar.

ScadaBR

ScadaBR es un sistema de control completo que funciona en sistemas operativos Windows (Windows XP y Windows 7), disponible en la licencia de código abierto (software libre) desarrollado a partir de un proyecto de investigación. El software ha sido desarrollado en el marco de la CERTI.

El software se compone de los siguientes componentes:

- ScadaBR Core: Núcleo central del sistema SCADA
- Módulo ScadaBR DA Server: Mantiene las etiquetas de las variables y permite leer y escribir sus valores.
- Módulo ScadaBR A&E: Encargado de la emisión de alarmas y la gestión de eventos.
- Módulo ScadaBR Historian: Base de datos de histórico de datos.
- Módulo ScadaBR Batch & Scripting: Encargado del control de las secuencias de comandos para realizar operaciones básicas.

ScadaBR es un software totalmente funcional y sin versiones limitadas.

IGSS FREE50

IGSS FREE50 es la versión libre del software SCADA IGSS V11 creado por Schneider Electric. Esta versión libre dispone de todas las funcionalidades de la versión comercial pero está limitado al uso máximo de 50 objetos.



-Logotipo del SCADA IGSS-

Con IGSS FREE50 se pueden crear proyectos con total funcionalidad. Algunas características que presenta son:

- Definir proyectos con un máximo de 50 objetos
- Recoger información de PLC sin límites temporales
- Permite usar más de 70 drivers para PLC
- Es posible actualizarlo a la versión comercial en cualquier momento.

Esta versión libre del SCADA IGSS también incluye proyectos preconfigurados que pueden ser utilizados en el modo Demo. Esta opción permite mostrar las capacidades del software sobre diversos entornos simulados.

IGSS FREE50 funciona bajo entornos Windows.

IndigoSCADA

IndigoSCADA es un software SCADA-DCS para las plataformas Windows y Linux. Está desarrollado principalmente en lenguaje C/C++. Sus características principales son:

- Uso de datos para generar informes de gestión diarios / semanales / mensuales.
- Presentación gráfica de los datos de tiempo real e histórico.
- Notificación de alarma y eventos en tiempo real
- Bases de datos en tiempo real e histórica integrada
- Editor de SQL que permite el mantenimiento en línea de las bases de datos de tiempo real e histórico.
- Múltiples usuarios con diferentes permisos de acceso.
- Funcionamiento con IEC 60870-5-101.
- OPC DA 2.05, A & E 1.1, HDA 1.20, DNP 3.0, RFC 1006, Modbus.
- Copias de seguridad fáciles.
- Soporta múltiples ventanas del HMI.

Como ejemplo especial, a continuación se recoge un sistema SCADA diseñado en exclusiva para funcionar sobre dispositivos móviles.

TeslaMultiSCADA

TeslaMultiSCADA es un SCADA para el acceso a los datos de producción y de proceso desarrollado para dispositivos Android, utilizando protocolos industriales como Modbus TCP (UDP), Siemens ISO / TCP y Ethernet / IP para la comunicación.

La empresa proporciona una versión de evaluación que tiene las siguientes restricciones:

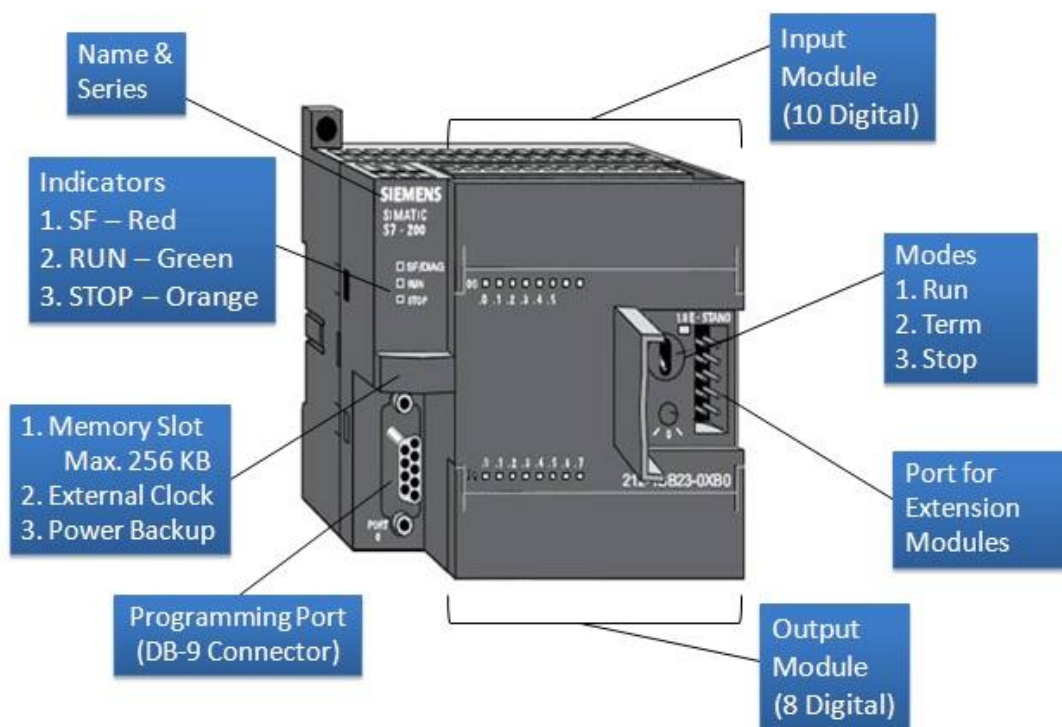
- No se puede utilizar más de 16 etiquetas en el proyecto.
- No se puede importar proyectos.
- El período de evaluación es de 60 días.

Software libre y sistemas de control: ¿un matrimonio con futuro?

Aunque la expansión del software libre para los sistemas de control industrial está aumentando su oferta de manera notable en los últimos años, parece complicado que una de estas soluciones llegue a consolidarse como solución definitiva para controlar procesos críticos dentro de la industria. Seguramente han de pasar varios años para que, por ejemplo pueda verse software de código libre controlando una central nuclear u otro entorno industrial de alta criticidad.

ANEXO B

PLC S7-200



ANEXO C

CONOCIENDO EL VARIADOR ATV31

- ❖ Conecte los cables de motor. La conexión realizada internamente es en triángulo ya que el motor es un 230/380 V y alimentaremos al variador 230 V monofásico. Conecte maleta y arme disyuntor.
- ❖ En el display debe aparecer la palabra "rDY". Apriete el ENT
- ❖ Mediante las flechas "▼" "▲" muévase por diferentes menús: **SET** (parámetros de ajustes), **DRC** (parámetros de placa motor y función "autoajuste"), **I-O** (asignación de E/S), **CtL** (parámetros de canales de control), **Fun** (configuración de funciones), **FLt** (menú de defectos), **CON** (parámetros de comunicación) y **SUP** (parámetros de supervisión).

Versión ATV...A

- 
- Led rojo "bus CC en tensión"
 - 4 dígitos de "7 segmentos"
 - Para pasar al menú o al parámetro previo o para aumentar el valor mostrado.
 - Para pasar al menú o al parámetro siguiente o para disminuir el valor mostrado
 - 2 indicadores de estado CANopen.
 - Para salir de un menú o de un parámetro o para desechar el valor mostrado y volver al valor anterior grabado en la memoria.
 - Para entrar en un menú o en un parámetro o para registrar el parámetro o el valor mostrado.