



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

Curso Académico:

Resumen

Lo que se persigue en este trabajo es conseguir una regulación automática de la calefacción de una granja mediante el control de temperatura del agua del circuito interno de una caldera de baja presión.

Para ello, primero se especificarán las funciones que debe cubrir el control, exigidas por usuario final en el pliego de condiciones y la forma con la que se tendrá que poder supervisar la instalación.

Lo siguiente será definir el equipo que constituirá el control. En este caso se ha exigido que se realice con un autómata TM241, una pantalla táctil HMIS5T y un variador de frecuencia V1000 como elementos principales. También se requerirá, para el correcto funcionamiento, de sondas de temperatura y (en caso de no ser suficientes las entradas y salidas ya existentes en el autómata) de tarjetas de entrada/salida digitales y/o analógicas.

Una vez conocido el qué y con qué se va a realizar el control, se mostrará cómo se ha realizado la programación del autómata y la pantalla táctil para conseguir que cumplan con lo exigido.

Como validación del trabajo, se incluirán algunas comprobaciones realizadas modificando las temperaturas manualmente mediante la pantalla para comprobar que el programa del autómata es correcto y que las salidas a los dispositivos que irán subordinados al autómata se activan correctamente, además de lo demandado en el pliego de condiciones.

Palabras Clave: autómata, panel HMI, variador, So-Machine, Vijeo-Designer.

Resum

El que es persegueix en aquest treball és aconseguir una regulació automàtica de la calefacció de una granja mitjançant el control de temperatura de l'agua del circuit intern d'una caldera de baixa pressió.

Per això, primer s'especificaran les funcions que ha de cobrir el control, exigides per l'usuari final en el plec de condicions i la forma amb què s'haurà de poder supervisar la instal·lació.

El següent serà definir l'equip que constituirà el control. En aquest cas s'ha exigint que es faci amb un autòmat TM241, una pantalla tàctil HMIS5T y un variador de freqüència V1000 com a elements principals. També es requerirà, per al correcte funcionament, de sondes de temperatura, de relés i (en cas de no ser suficients les entrades y eixides ja existents en l'autòmat) de targetes de entrada/eixida digitals i/o analògiques.

Una volta conegut el què i amb què es va a realitzar el control, es mostrarà com s'ha realitzat la programació de l'autòmat y la pantalla tàctil per aconseguir que compleixin amb l'exigint.

Com a validació del treball, s'inclouran algunes comprovacions realitzades modificant les temperatures manualment mitjançant la pantalla per a comprovar que el programa de l'autòmat es correcte y que les eixides als dispositius que aniran subordinats a l'autòmat s'activen correctament, a més del demandat en el plec de condicions.

Paraules Clau: autòmat, panel HMI, variador, So-Machine, Vijeo-Designer.

Abstract

What is sought in this work is to achieve automatic control of the heating of a farm by controlling the water temperature of the internal circuit of a low pressure boiler.

To do this, functions to be covered by the control required by the end-user and the way in which he will have to be able to monitor the installation shall be specified in the specifications first.

The following will be to define the hardware that will constitute the control. In this case it has been required to be executed with a TM241 PLC, a touch screen HMIS5T and a V1000 inverter as main elements. Will also be required for the proper operation of temperature sensors, relays and (if the PLC does not have enough inputs and outputs) modules of digital/analog inputs and/or outputs will be required.

Once is known what and with what the control will be performed, how to do the programming of the PLC and the touch screen will be explained.

As a validation of the work, some checks are included made by modifying the temperature manually using the screen to verify that the PLC program is correct and the outputs to the devices that will subordinate to the controller operate properly, in addition what has been required in the specification will be included.

Key Words: PLC, HMI panel, inverter, So-Machine, Vijeo-Designer.

ÍNDICE

MEMORIA	1
1. OBJETIVOS	3
2. PRESENTACIÓN DEL PROBLEMA	4
3. PLIEGO DE CONDICIONES	6
3.1. Especificaciones del Control.....	6
3.2. Especificaciones del Equipo.....	8
3.3. Especificaciones de Resultados	8
4. CARACTERÍSTICAS FUNCIONALES	9
4.1. Autómata.....	9
4.2. HMI	12
4.3. Variador de Frecuencia	13
5. CARACTERÍSTICAS ESPECÍFICAS DE LOS EQUIPOS	15
5.1. Autómata TM241CEC24T	15
5.2. Panel HMIS5T	16
5.3. Variador V1000.....	17
6. PROGRAMACIÓN	18
6.1. Programa del Autómata.	18
6.2. Programación de la Pantalla Táctil	33
6.3. Configuración del Variador.....	44
7. COMUNICACIÓN ENTRE LOS EQUIPOS	46
7.1. Configuración E/S	47
7.2. Transferencia del Programa.	50
7.3. Conexiones	52
8. COMPROBACIONES Y CORRECCIONES	57
9. CONCLUSIONES	61
10. BIBLIOGRAFÍA	62
PRESUPUESTO	63
ANEJOS	67
Listado de Instrucciones	69
Lista de Variables	72
Paneles para el Control y Monitorización	73



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

VIVES JOVANÍ, VICENTE

**PROYECTO DE CONTROL Y
MONITORIZACIÓN DEL SISTEMA DE CALEFACCIÓN
DE UNA GRANJA AVÍCOLA MEDIANTE CALDERA,
AUTÓMATA PROGRAMABLE Y
PANEL DE USUARIO**

MEMORIA

Curso Académico: 2014-15

1. OBJETIVOS

El objetivo principal de este trabajo es realizar una integración de diversas tecnologías, muy presentes en la industria actual, como son los autómatas programables, los paneles táctiles HMI y los variadores de frecuencia para que puedan llevar a cabo la maniobra de control automática de la calefacción de una granja avícola y dar solución así a un problema real. Además de crear esta maniobra de control también es un objetivo el probar, aunque no sea posible en el sistema real, su correcto funcionamiento con elementos disponibles en el laboratorio.

Otro objetivo importante es la creación de funciones propias, como lo ha sido en este trabajo la función del control de temperatura mediante PID, y no utilizar así funciones predefinidas, cerradas a los usuarios y que en caso de cambiar el programa de autómatas podrían causar problemas.

Con esto último se busca que este trabajo no solo sirva para resolver un problema específico sino también para otros posibles trabajos que requieran la implementación de estas mismas funciones o similares.

Como objetivo secundario, se intentará exponer de forma clara como realizar la programación y configuración de cada uno de estos equipos por separado y como realizar la posterior integración de estos equipos para que comunicándose entre ellos cumplan con el control exigido.

2. PRESENTACIÓN DEL PROBLEMA

La nave de 1375m² está situada en Chert, Castellón coordenadas 40°30'11.63" N 0°10'49.97" E.



Figura 1: Localización de la nave avícola

Constaba anteriormente de un sistema de calefacción con estufas que quemaban el combustible en el interior de la granja para mantener la temperatura. La regulación era del tipo todo o nada, es decir, cuando bajaba la temperatura se activaba la entrada de aire y cuando la temperatura era demasiado elevada se detenía la entrada de aire. Este sistema demostraba ser ineficiente porque el calor iba directamente al techo y a veces las estufas se apagaban por estar demasiado tiempo sin entrada de aire, además del espacio físico que ocupaban y el humo que aparecía a causa de la combustión en el interior de la nave.

Por estos motivos se ha decidido implementar una instalación de calefacción por suelo radiante. Esto es, tubos de polietileno por debajo del hormigón del suelo de la nave por los que circula agua a una determinada temperatura de forma que calienta el espacio de abajo hacia arriba.

El agua caliente es proporcionada por la caldera, y dos circuladoras mueven el agua hacia dentro de la nave. Las circuladoras funcionan de forma independiente una de la otra y cada una se encarga de una mitad de la nave también con sondas de temperatura independientes para cada mitad. De esta forma podría darse el caso que la temperatura de una mitad de la nave bajase por debajo de la temperatura consigna y esto obligará a que la circuladora correspondiente a esa mitad aporte agua caliente para volver a compensar la temperatura. Debe tenerse en cuenta que el agua que se manda desde la caldera viaja por tubos hasta llegar a la nave. Por eso en días con temperatura exterior baja habrá pérdidas de calor importantes.

La caldera dispone de un circuito interno con un volumen de agua de 2000 litros de donde las circuladoras suministrarán a la nave de agua caliente cuando sea necesario. El agua dentro de este circuito debe mantenerse a una temperatura establecida, de forma que siempre esté disponible para compensar pérdidas de temperatura en la nave lo más rápido posible.



Figura 2: Caldera y sistema alimentación

Motor que acciona el sinfín de alimentación y que debe controlar el variador V1000.



Figura 3: Circuladoras 1 y 2 de la Nave

El sistema de alimentación de la caldera consta de dos etapas. En la primera etapa el pellet, que se encuentra en un almacén, se transporta mediante un sinfín hasta una tolva. Desde ahí la caldera mediante otro sinfín se alimentará. El motor que actúa sobre este último sinfín es el que tiene que estar controlado por el variador de frecuencia, para aumentar caudal de alimentación o reducirlo en función de la demanda de calor.

3. PLIEGO DE CONDICIONES

En este documento se especifican las condiciones bajo las cuales ha de realizarse el trabajo. Cuando todo lo exigido aquí se haya realizado será posible dar por completada la maniobra exigida.

3.1. Especificaciones del Control

A continuación se enumeran todas las funciones que debe cumplir el control a diseñar y que se han exigido.

La caldera tiene que ejecutar cuatro funciones: Encendido, Control de la temperatura, Parada y Limpieza.

Para poder encenderse la caldera tiene que activar su entrada de aire, la entrada de combustible y unas resistencias que junto a un ventilador (al conjunto se le hace referencia como mecheros) encenderán el combustible. Se exige que la cantidad de combustible, para asegurar el encendido, ha de ser superior a 2Kg.

Una vez la caldera ya este encendida tendrá que pasar directamente a su función de control de temperatura. Para saber si está encendida se puede emplear la sonda de humos existente en la caldera. Esta función tiene que mantener la temperatura del circuito de la caldera a una temperatura determinada y regular el funcionamiento de las bombas que aportan agua caliente a la nave.

El mantenimiento de la temperatura del agua de la caldera se realizará modificando la cantidad de combustible que entra en la caldera a través del variador de frecuencia que modificará la frecuencia de giro del motor del sinfín de alimentación.

Las circuladoras tienen que funcionar de forma independiente en función de la zona de la nave que requiera compensar una bajada de temperatura. Para controlar la temperatura de la nave existen dos sondas de temperatura situadas en los extremos de la nave. Además, se exige que para no forzar el funcionamiento de la alimentación de la caldera, cuando lleven un tiempo paradas funcionen durante un cierto tiempo.

El motivo es el siguiente: cuando las circuladoras no mueven el agua del circuito, el agua que se encuentra dentro de la caldera se mantiene a temperatura suficiente para compensar una bajada de temperatura en la nave, mientras que el resto del agua está enfriándose. Si el volumen de agua del circuito de la caldera no es suficiente para compensar la bajada de temperatura, la caldera tendrá que calentar muy rápido el agua que ha estado enfriándose. Para evitar que el sistema tenga que trabajar cerca de su límite se ha exigido esta metodología de funcionamiento.

Dentro de esta función también ha de existir un sistema de compensación por pérdidas de calor del agua que se active cuando la temperatura exterior sea muy baja. Esto es necesario porque al tener que viajar el agua por tuberías desde la caldera hasta la granja se producen pérdidas de temperatura cuando la temperatura exterior es muy inferior a la temperatura del agua provocando un incremento

del tiempo que se tarda en compensar una bajada de temperatura. Este sistema de compensación ha de elevar la temperatura del agua de la caldera una cierta cantidad de grados cuando la diferencia entre la temperatura exterior y la del agua de la caldera supere un cierto valor. Si la diferencia supera por más del doble este valor, entonces el incremento será del doble de grados.

La función de apagado detendrá el sistema de alimentación y la entrada de aire provocando el apagado de la caldera.

La función de limpieza solo ha de activarse cuando la caldera está parada y activará un sinfín que incorpora la caldera para la extracción de la ceniza acumulada.

En el panel se exige que aparezcan los siguientes elementos para poder efectuar la monitorización y control de la maniobra.

Para la monitorización se pide que aparezcan pilotos y valores numéricos que indiquen:

-**Estado de la caldera:** Encendiendo, control de temperatura, limpieza o parada.

-**Potencia de trabajo:** Sabiendo que la caldera es de 600000Kcal/h se quiere visualizar el volumen de trabajo en el que se encuentra la caldera.

-**Consumo de pellet:** Este valor ha de mostrar el consumo en ese mismo instante en Kg/h.

-**Todas las temperaturas:** Temperaturas de la nave, temperatura exterior, temperatura del agua de la caldera y temperatura de humos.

Por otra parte se requieren de variables modificables por pantalla (del tipo botones o introducción numérica) para poder efectuar el control del sistema.

-Variables tipo botón que modifiquen la función de la caldera:

- **Encendido:** empieza la secuencia de arranque de la caldera
- **Limpieza:** activa el sistema de extracción de ceniza.
- **Parada:** activará la secuencia de paro.

-Temperatura consigna del agua a la que se debe mantener el agua dentro del circuito de la caldera.

-Temperatura consigna de la nave. Este parámetro es el que obligará a las circuladoras a bombear agua caliente a la nave en caso que una de las sondas marque una temperatura por debajo de esta.

-Diferencia entre la temperatura exterior y la temperatura del agua a partir de la cual se activará la compensación y con qué incremento de temperatura.

-Temperatura del agua de la caldera por encima de la cual las circuladoras podrán bombear agua caliente a la nave (evitar un funcionamiento innecesario).

-Los tiempos de funcionamiento y de espera de las circuladoras cuando llevan un tiempo sin activarse por una bajada de temperatura.

3.2. Especificaciones del Equipo

Se exige que el control se realice con un autómatas TM241CEC24T, un panel HMI5ST y un variador V1000 de Omron. El autómatas y el panel se programarán en los entornos So-Machine y Vijeo-Designer respectivamente.

El autómatas tiene que controlar el encendido y apagado del variador de frecuencia, circuladoras 1 y 2, sinfín ceniza, entrada de aire y mecheros de la caldera. Cada uno de estos ocupará una de las diez salidas de las que dispone el autómatas con el siguiente orden:

Q0: variador de frecuencia.

Q1: circuladora 1.

Q2: circuladora 2.

Q3: ventilador entrada aire caldera.

Q4: mecheros encendido caldera.

Q5: sinfín ceniza

Además para el control de la frecuencia del variador se requerirá de una salida analógica aportada por una tarjeta mixta de E/S analógicas TM3AM6/G. Este módulo posee dos salidas analógicas y cuatro entradas analógicas, para el control del variador se utilizará la salida QW0 del módulo. Aunque solo para la comprobación, también se utilizará una entrada analógica 0-10V de este mismo módulo para la simulación con un potenciómetro de la temperatura del agua de la caldera.

Para este control no se requerirá de ninguna entrada digital, pero sí de entradas analógicas de temperatura. Al no disponer el autómatas de entradas analógicas se han facilitado dos módulos de expansión analógicos TM3TI4/G con cuatro entradas analógicas cada uno. Las temperaturas conectadas al primer módulo serán la temperatura exterior, la temperatura del agua de la caldera y la temperatura de humos cada una de estas conectadas en la posición IW0, IW1 e IW2 respectivamente. En el segundo módulo se conectarán las dos sondas de temperatura que indican la temperatura a la que se encuentran las dos mitades de la nave, una sonda en IW0 y la otra en IW2. De esta forma si se quisiera poner otra sonda para controlar mejor la temperatura en una mitad estarían separadas IW0 e IW1 para una mitad y las otras dos, IW2 e IW3 para la otra mitad.

3.3. Especificaciones de Resultados

Como resultados finales a entregar se generará un documento con el listado de instrucciones del programa del So-Machine, así como una lista de todas las variables que en el programa se utilizan. También ha de aparecer la forma final que tendrán los paneles con los que se controlará el proceso de calefacción de la nave. Estos dos programas (el del So-Machine y el del Vijeo-Designer) se entregarán en un pendrive al usuario final por si hubiera que realizar modificaciones al programa en un futuro.

4. CARACTERÍSTICAS FUNCIONALES

En este apartado se va a describir que es un autómata, una pantalla HMI y un variador de frecuencia. La mayor parte de esta información es de tipo descriptiva y no tiene más trascendencia que aportar cierta información para facilitar el entendimiento del trabajo por alguien que no esté familiarizado con los equipos utilizados.

4.1. Autómata

Definición

Un autómata programable es un aparato electrónico que ejecuta sus funciones siempre siguiendo la misma secuencia. Es programable porque dispone de una memoria en la que se pueden introducir instrucciones para que este las ejecute dentro de su bucle de ejecución y conseguir así un funcionamiento automatizado.

Estructura del bucle

Las funciones que se ejecutan con cada bucle son las siguientes:

-Lectura de entradas del autómata: Al principio de cada bucle el autómata comprueba el estado de las entradas que están conectadas al autómata. Estas entradas pueden estar activas o desactivas, en cualquier caso el autómata guarda este valor en su memoria. Si las entradas son analógicas, lo que se guarda en memoria es un número que representa el valor de la magnitud en función de un voltaje o intensidad producido por esta magnitud. La secuencia en bucle implica que un cambio en las entradas una vez se ha pasado a la siguiente función no será detectado hasta la siguiente iteración.

-Ejecución del programa: una vez actualizadas las entradas, el programa ejecutará por orden todas las instrucciones que se hayan introducido en su memoria. De forma similar a lo que ocurría con la modificación de las entradas, cuando el autómata cambia a la siguiente instrucción ya no importan las instrucciones anteriores. Por ejemplo, si primero se activa una variable pero más adelante se cambia su valor la variable solo conservará el último valor. La ejecución del programa modificara valores en la memoria de variables del autómata y por tanto en sus salidas, las cuales, realizarán las acciones del proceso.

-Escritura de salidas: cuando el autómata acaba la ejecución del programa, revisa todas las variables de salidas y actualiza los nuevos valores. Si al final del programa una variable de salida del tipo booleana esta activa el autómata activa esa salida independientemente del valor anterior. Por lo que a las analógicas respecta, el autómata modificara su valor de tensión o intensidad de forma proporcional al valor que aparezca en su memoria de salida.

-Tareas internas del autómata: Antes de empezar un nuevo bucle, el autómata comprueba si se han producido errores, almacena la duración del ciclo completo, actualiza valores internos de sus tablas de datos, etc.

Perro Guardián

Es importante tener en cuenta que cada una de estas fases del ciclo hacen incrementar el tiempo de duración de este (principalmente la ejecución del programa). En el caso de que este tiempo no fuese despreciable frente al tiempo de ejecución del proceso, se podrían producir desfases entre el sistema de control y el proceso. Por ejemplo si una entrada se activa, pero durante un corto periodo de tiempo, es posible que si el tiempo de ciclo del autómata no es despreciable respecto a este periodo de activación, el autómata no se dé cuenta de que se ha producido este impulso.

Este es el motivo por el que los autómatas incorporan una función denominada "Perro guardián" (Wach dog). Esta función compara el tiempo que tarda en realizarse cada ciclo con un tiempo especificado, y en caso de que el primero sea superior al segundo mandara una señal de error.

Para el caso de este proyecto esto no requiere especial atención ya que al ser procesos donde se controla temperaturas y no señales que aparezcan durante un reducido periodo de tiempo no importa que el autómata tarde unas décimas de segundo más en reconocer un incremento de temperatura de una décima de grado.

Hardware

Un autómata está constituido por una fuente de alimentación, CPU, bastidor, E/S digitales, E/S analógicas y otros módulos.

La **fuente de alimentación** aporta energía eléctrica al autómata. El autómata funciona internamente a 5V DC pero normalmente se conectan a 220V AC o 24V DC.

La **CPU** del autómata, al igual que en un ordenador, actúa como cerebro del sistema. Consta de microprocesador, unidad de memoria y una unidad aritmética.

El **bastidor** es un soporte sobre el que se montan los distintos módulos que se pueden incorporar al autómata y sirve como sistema de comunicación entre los módulos y el autómata.

Las **E/S digitales y analógicas** pueden aparecer en el autómata porque el propio autómata ya las incorpora o se pueden incluir o aumentar su número conectando módulos de E/S. Las entradas/salidas digitales solo permiten conocer un estado (activado/desactivado) de un sensor. Mientras que las E/S analógicas permiten conocer el valor (dentro de un rango determinado) que puede tomar una magnitud física. Las entradas analógicas son convertidores Analógicos/Digitales, de esta forma convierten un voltaje de entrada en un valor numérico, comprendido dentro de un intervalo, de forma que para el menor voltaje tomara el valor más bajo de ese intervalo. Las salidas analógicas funcionan de igual forma pero transformando de Digital a Analógico, el autómata modifica el voltaje de salida en función del valor numérico que toma esa salida.

Algunas operaciones especiales requieren de módulos especiales para que el autómata pueda realizarlas. Como ejemplo de estas operaciones podemos mencionar el posicionamiento de ejes, arrancadores de motores, interconexión con otros autómatas, etc.

Software

La programación de la mayoría de los autómatas modernos puede realizarse en cinco lenguajes distintos.

Lenguaje Ladder: cuando aparecieron los primeros autómatas, se estableció un lenguaje de programación lo más parecido a como se realizaban los automatismos en ese tiempo, de esta forma el personal de la empresa sería preparado más rápida y fácilmente. De ahí apareció el lenguaje ladder o diagrama de contactos, similar a diagramas eléctricos para representar esquemas de lógica.

Lista de Instrucciones: es el lenguaje más literal y por tanto el más potente, parecido al lenguaje ensamblador que utiliza en el ordenador donde cada línea es una instrucción. Cualquier programa escrito en cualquier otro lenguaje puede traducirse a este lenguaje de programación.

Diagrama de Bloques Funcionales: surge de la forma de representar los circuitos lógicos en electrónica. Cada símbolo representa una operación, una puerta lógica, etc. Un conjunto de estos símbolos puede encapsularse dentro de otro símbolo que los englobe y utilizarse en otros sitios del programa. Se trata por tanto de un lenguaje muy gráfico.

Texto Estructurado: hoy en día este lenguaje está ganando mucha fuerza gracias en parte a la enorme expansión que han experimentado las aplicaciones programadas en java ya que es un lenguaje similar. Otra razón es por ser un lenguaje muy potente con el que es relativamente fácil programar algoritmos complicados en los que se requieren bucles, condicionales, etc.

Lenguaje Diagramas Funcionales Secuenciales (SFC): está considerado como una evolución del lenguaje GRAFCET, aunque este último no es un lenguaje de programación el SFC sí lo es. Es un lenguaje potente para la representación de automatismos secuenciales.

4.2. HMI

HMI significa Interfaz Humano Maquina, a través de estos dispositivos recibimos información del proceso y se transfieren órdenes al sistema. Hay diversos tipos de interfaces, algunas de las cuales son:

Interfaces gráficas: mediante el uso de periféricos del ordenador se introducen entradas de variables y a través de la pantalla del ordenador se realiza la motorización de variables, salidas, etc.

Interfaces de usuario web: podríamos situarlas dentro del grupo de interfaces gráficas, pero estas reciben entradas y envían salidas a través de Internet. La visualización se realiza mediante navegador web.

Pantallas táctiles: reciben entradas por contacto de los dedos con la pantalla y la información de variables y salidas aparece también en la pantalla. Este sistema es ampliamente utilizado en dispositivos móviles, procesos industriales, maquinas, etc.

Interfaces de líneas de comando: la transmisión de órdenes y respuestas se realiza mediante escritura de comandos por ordenador y por impresión de texto en la pantalla respectivamente.

Interfaces de voz: aunque también pueden introducirse entradas con botones, es por la capacidad de recibir información mediante mensajes de voz por lo que destacan. De igual forma se obtienen las respuestas del sistema.

Panel HMI

En este trabajo se utilizará una interfaz por pantalla táctil. Estas pantallas pueden tener distintos tamaños, existen (por establecer un orden de magnitud) desde 3.5" hasta 22", y distinta resolución en función del tamaño. Suelen necesitar de una fuente de alimentación externa ya que muchas funcionan a 24V DC y no a 220V AC.

Su principal función es la de actuar como medio a través del cual se pueda establecer una comunicación entre la persona y el autómatas que controla el proceso.

Esta comunicación puede realizarse en los dos sentidos:

El autómatas recibe información del sistema que está controlando a través de sus entradas analógicas y digitales, para posteriormente al procesamiento del programa sobrescribir valores en sus propias salidas. Toda esta información sobre los valores que toman las entradas y salidas es transferida a la pantalla táctil, juntamente con los valores que puedan tomar las distintas variables internas que el programa del autómatas utilice. Si se ha programado el panel para que aparezca toda esta información, seremos capaces de visualizarla en él.

El otro sentido que puede tomar la comunicación es desde la persona hasta llegar al autómata. En el panel táctil se pueden activar botones (representando a variables del autómata) o escribir valores numéricos de forma que cuando el autómata reciba esta información cambie sus variables internas o salidas.

Por lo general hay que tener en cuenta que existen muchas marcas tanto para paneles táctiles como para autómatas (Siemens, Schneider, Omron, Rockwell, etc...). Esto implica que hay que asegurarse de que será posible la comunicación entre panel y autómata en caso de que no sean de la misma marca, aunque ya hoy en día, una pantalla por ejemplo de Pro-face pueda comunicarse con más de seis marcas de autómatas distintas. En este trabajo se utilizan ambos elementos de la marca Schneider por lo que no habrá problemas de este tipo.

Software

La forma de programar estos dispositivos es a través de Ethernet (los que dispongan de tal conexión) y mediante un cable USB desde el ordenador al panel. Existen varios software para la programación de estos paneles, podría decirse que uno por cada marca de paneles que existe (Vijeo Designer para pantallas de Schneider, GP-Pro-Ex para la marca Pro-Face, etc.)

4.3. Variador de Frecuencia

Un variador de frecuencia es un dispositivo electrónico que permite el control de la velocidad de un motor de corriente alterna por medio de la variación de la frecuencia de alimentación o bien controlar la velocidad de un motor de continua por regulación del voltaje.

Algunos constan de entradas y salidas de control tanto digitales (contactores, pulsadores, contactos de relé...) como analógicas mediante variación de voltajes (por ejemplo 0 a 10V) o intensidades (4 a 20mA o parecidos).

También poseen sistemas de comunicación del tipo RS-232, RS-485 o conexiones del tipo RJ-45 o USB para comunicaciones con ordenadores o autómatas. Cada fabricante de variadores dispone de un software para poder programarlo y monitorizarlo mediante dispositivos de control.

Para explicar de forma simple el sistema que emplea el variador de frecuencia para controlar la velocidad del eje de un motor asíncrono se debe tener en cuenta la ecuación:

$$n = 60 \frac{f}{2p}$$

Ecuación 1: Velocidad de giro de un motor

Donde n es la velocidad del eje del motor en rpm, f la frecuencia a la que se alimenta el motor y $2p$ es el número de pares de polos del motor.

De esta forma si por ejemplo tenemos un motor de cuatro polos (2 pares de polos) alimentado a una frecuencia de 50Hz tendrá una velocidad nominal de:

$$n = 60 \frac{50}{2} = 1500 \text{ rpm}$$

Ecuación 2: Ejemplo sin variador

Si mediante el uso de un variador se modifica la frecuencia de alimentación entre un rango de 5Hz y 100Hz entonces se obtiene:

$$n = 60 \frac{5}{2} = 150 \text{ rpm}$$

Y

$$n = 60 \frac{100}{2} = 3000 \text{ rpm}$$

Ecuación 3: Velocidad baja frecuencia

Ecuación 4: Velocidad alta frecuencia

Hay que tener en cuenta que incrementar la velocidad por encima de la nominal implica al mismo tiempo reducir el par que el motor es capaz de proporcionar. Por lo tanto un incremento del doble de velocidad en un motor que proporciona alimentación a un sistema (como el de este trabajo) no implica que la cantidad de combustible proporcionada incremente en el mismo porcentaje.

Otras dos funciones importantes que debe realizar el variador, además del control de velocidad del motor, son las de aportar un par de arranque suficiente y proteger el motor de sobre intensidades. En sistemas como sinfines para que el motor arranque debe vencerse un par resistente provocado por rozamientos y por la carga ya presente en el sistema. Sin embargo el variador debe controlar que este par de arranque no provoque intensidades superiores a las que el motor puede soportar sin quemarse. De forma similar a lo anterior, cuando el motor esté funcionando y el sistema se quede bloqueado, por ejemplo una piedra atascada en un sinfín bloqueando su movimiento, el motor tratará de aportar un mayor par debido al incremento del par resistente, en consecuencia aumentará la intensidad que pasa a través del motor. Es trabajo del variador detener el motor antes de que esté acabe siendo inservible.

Aunque los variadores pueden controlar otros muchos aspectos del funcionamiento de los motores como podrían ser el frenado, la aceleración, la parada, la inversión de giro... son funciones que no presentan una necesidad de control obligatoria en este trabajo debido a las características del sistema.

5. CARACTERÍSTICAS ESPECÍFICAS DE LOS EQUIPOS

Continuando con el apartado anterior pero centrando la información en los equipos con los que se trabajará de ahora en adelante y en definitiva con los que se ha realizado el trabajo.

5.1. Autómata TM241CEC24T

El primer componente a presentar y el que constituirá la base del proyecto es el controlador lógico Modicon M241 de Schneider Electric.



Figura 4: Autómata TM241

El autómata, alimentado a 24V DC o 100-240V AC, posee una RAM de 64Mbytes 8 de los cuales se destinan principalmente en ejecutar el código de aplicación, mientras que el resto se encarga de las tareas de comunicaciones. La memoria Flash de 128Mbytes se emplea para realizar copias de seguridad en caso de corte de luz.

El modelo CEC24T consta por si solo de 24 E/S digitales (14 entradas de 24V DC y 10 salidas 0,5A), 2 puertos serie y puerto Ethernet. El autómata permite conectar módulos analógicos de E/S o de temperatura, en este caso se utilizará una tarjeta mixta (TM3AM6) de 4 entradas (0-10V DC) y 2 salidas (0-20mA, 4-20mA) analógicas con resolución de 12bits y ya en la instalación real se añadirán dos módulos de temperatura TM3TI4 con cuatro entradas analógicas cada uno.

Este autómata posee conexión Ethernet, dos líneas serie y conexión USB-mini-B. La conexión con el ordenador para programar el autómata será mediante USB (ordenador)-mini USB (autómata), mientras que la comunicación entre el panel y el autómata se realizará por línea serie. También sería posible la comunicación del autómata por Ethernet tanto con el ordenador como con el panel.

Además de las comunicaciones anteriores, este autómata también tiene un puerto CANopen que permite la comunicación con variadores de velocidad. Lo cual a pesar de resultar muy interesante, no se utilizará en este trabajo ya que la marca del variador a utilizar en este caso no es la de Schneider Electric.

La programación de este autómata se realiza mediante el software So-Machine y puede realizarse en seis lenguajes distintos:

- Lenguaje Lista de instrucciones (IL)
- Lenguaje de Contactos (LD)
- Diagramas de Funciones de Boque (FBD)
- Lenguaje Grafcet (SFC)
- Lenguaje Texto Estructurado (ST)
- Lenguaje CFC

El lenguaje empleado en este trabajo será el estructurado, ya que se trata de un lenguaje potente con el que resulta fácil realizar estructuras de bucle complejas y es relativamente fácil explicar el funcionamiento de un programa que utilice este lenguaje.

5.2. Panel HMIS5T

Lo que nos permitirá monitorizar y comunicarnos con el sistema es el panel que aparece en la imagen. Este panel tiene 2 tallas posibles de pantalla a utilizar (3,5" y de 5,7").



Figura 5: Panel HMIS5T y pantalla

Tal y como se observa en la base del panel, consta de cuatro conexiones, de izquierda a derecha son: conexión USB, Modbus, Ethernet y la alimentación. Existe también, aunque no sea visible en la imagen, una conexión USB-mini-B en el lateral izquierdo como la del autómata.

A parte de la alimentación (se realizará con la misma fuente de alimentación a 24V DC del autómata), la conexión USB, Modbus, Ethernet y mini-B se utilizan respectivamente para que la pantalla funcione con un programa en un dispositivo USB, para comunicar la pantalla con el autómata, para comunicar el panel con el autómata o con un ordenador y para conectar el panel y un ordenador.

La programación del panel se realizará mediante la conexión USB (ordenador)-mini B (panel) y con el software Vijeo-Designer.

5.3. Variador V1000

Existen distintos modelos en función de la potencia (hasta 15KW), del voltaje de entrada (200V o 400V) y del tipo de entrada (trifásica-monofásica).

Este variador puede configurarse mediante los botones que pueden verse en la parte frontal del mismo o bien con el software CX-Drive.

En este trabajo, al disponer el variador también de entradas analógicas y digitales, no se configurará y la frecuencia de giro a la que tiene que hacer girar el motor se la indicará en todo momento el autómata a través de una salida analógica.



Figura 6: Variador V1000

6. PROGRAMACIÓN

A partir de este punto se explicará paso a paso como realizar la programación del autómata y el panel HMI, al mismo tiempo que se constituye el programa que el autómata ejecutará para que se realice el control deseado.

Primeramente se explicará la programación del autómata y luego la del panel, ya que al utilizar dos software distintos de programación resultaría confuso explicar ambos a la vez.

6.1. Programa del Autómata.

La programación del autómata consiste en escribir un conjunto de instrucciones agrupadas en funciones dependiendo del propósito y luego introducir estas funciones dentro de la memoria del autómata para que las ejecute en cada ciclo. En este caso se crearán seis funciones y se introducirán en la sección MAST para activar su ejecución.

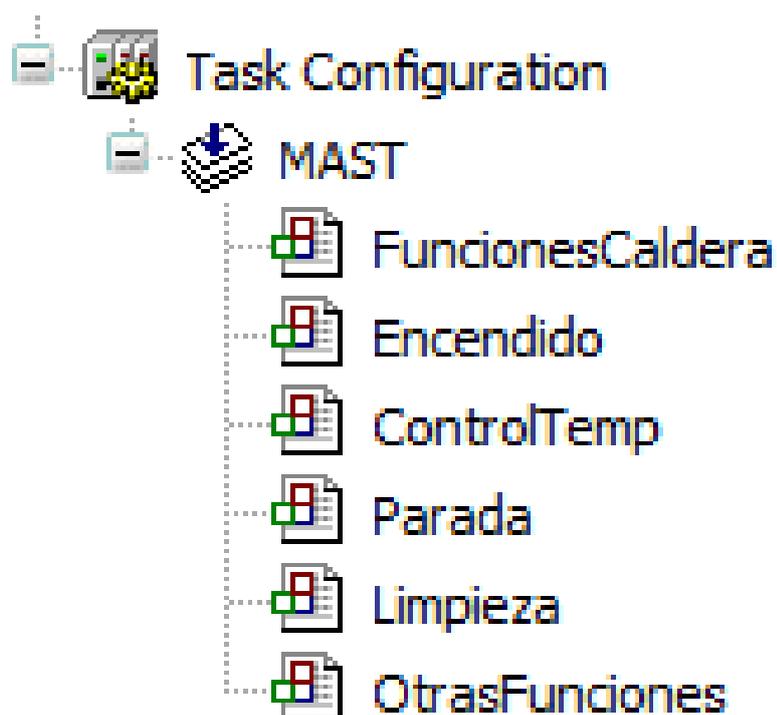


Figura 7: Lista funciones del programa

El autómata ejecuta las líneas del programa en orden empezando con la primera función que aparece en MAST, FuncionesCaldera y acabando con OtrasFunciones.

Una vez presentado el resultado final de la programación del autómeta se va a explicar la interfaz del programa So-Machine y también como se han constituido paso a paso los programas de las funciones que controlarán el proceso tal y como se ha especificado.

6.1.1. Creación de un nuevo proyecto.

En primer lugar hay que abrir el software de programación de los autómetas programables de *Telemecanique*, el So-Machine.

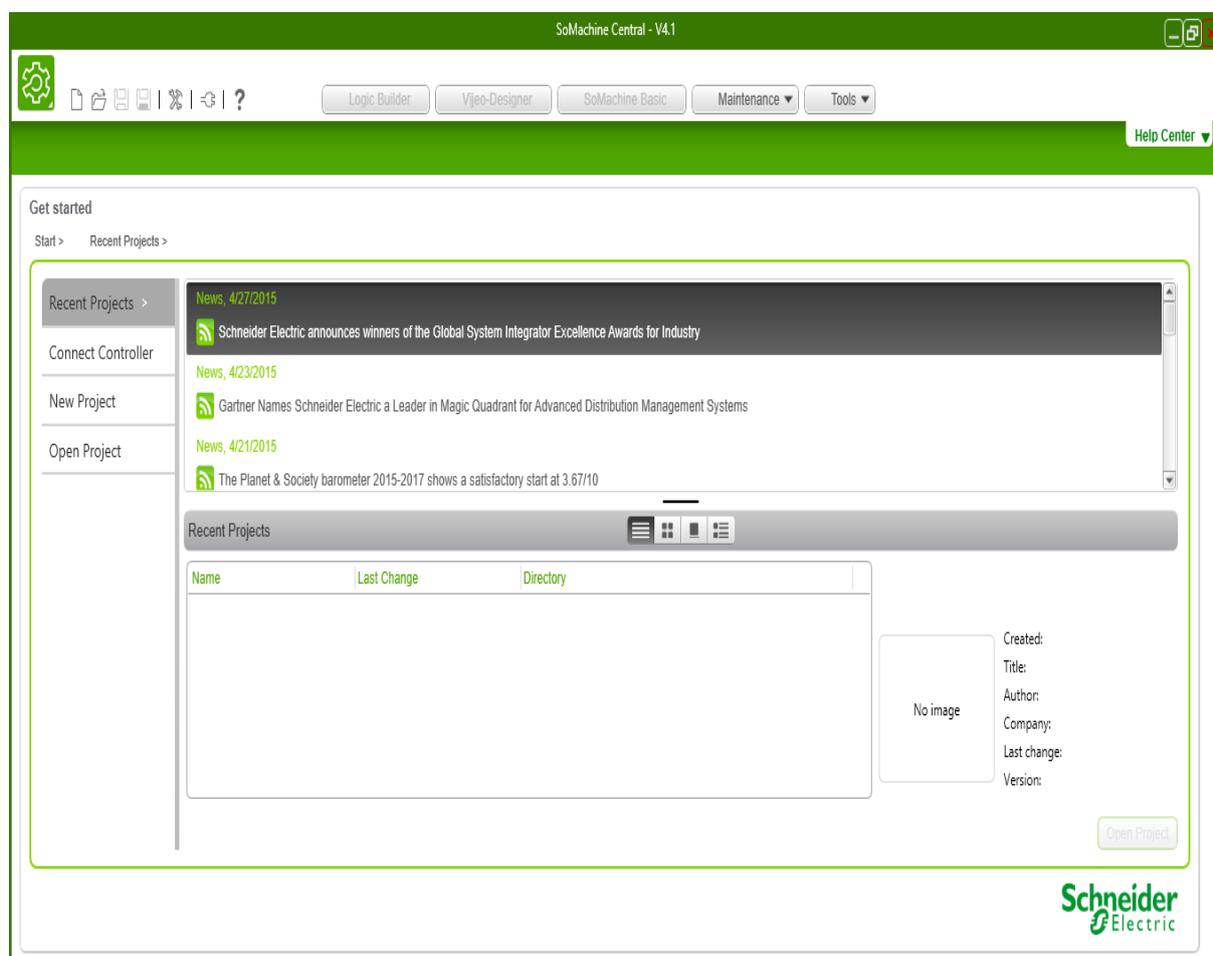


Figura 8: Ventana de inicio So-Machine

Una vez iniciado el programa hay que crear un proyecto nuevo, para ello pulsando en New Project y luego Empty Project hasta llegar a la ventana de la Figura 9.

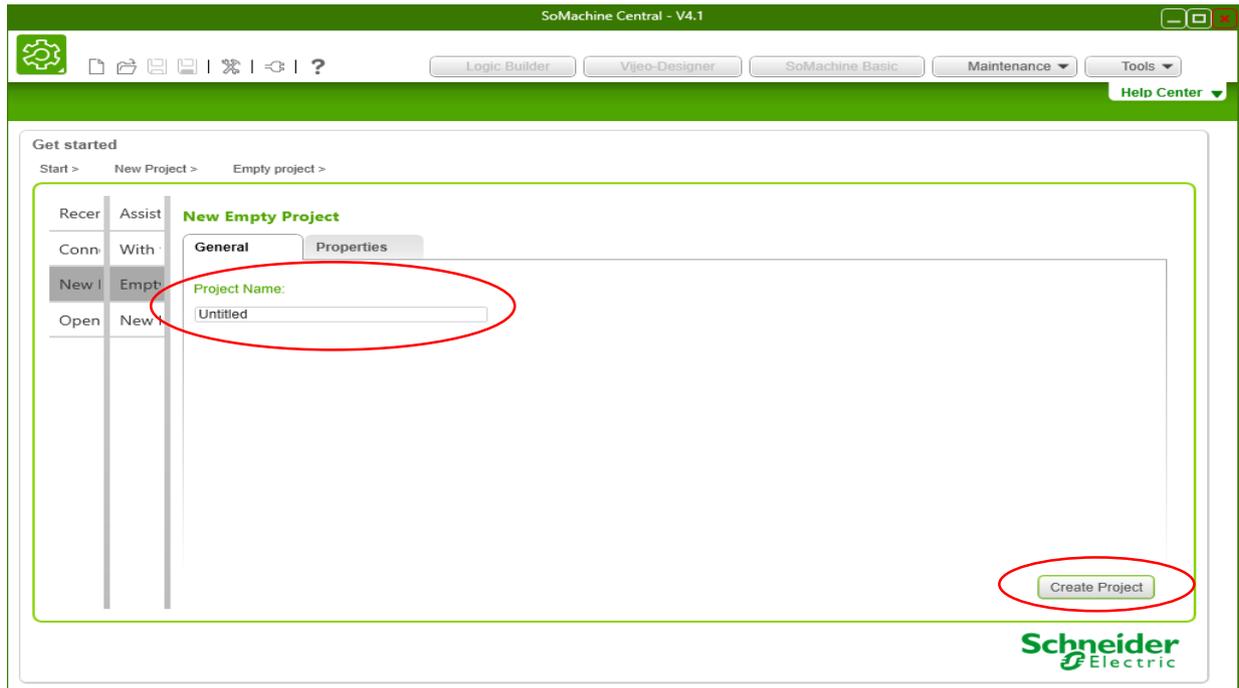


Figura 9: Crear proyecto

Ahora se dará nombre al proyecto en Project Name y se pulsará Create Project. En la siguiente ventana ya podemos guardar el proyecto (tercer icono en la parte superior izquierda) y simplemente clicando en Logic Builder ya es posible empezar a introducir los equipos que se utilizarán.

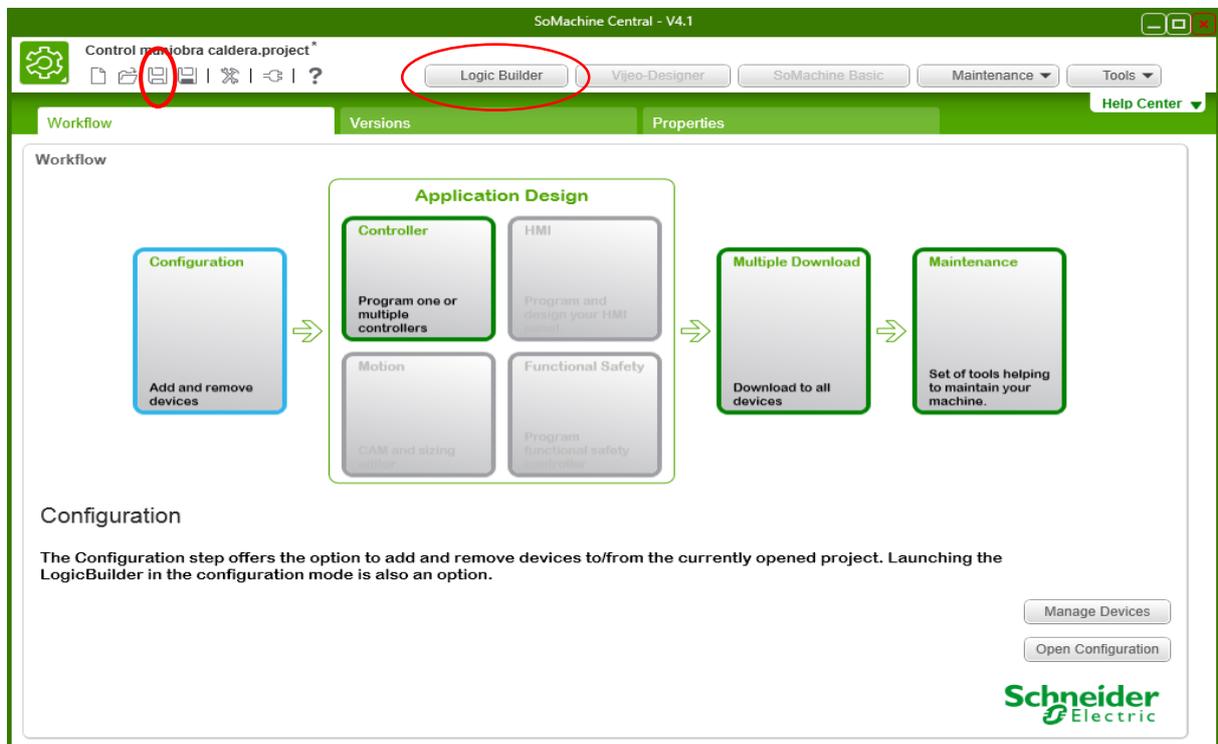


Figura 10: Guardar proyecto

La Figura 11 corresponde a la parte derecha de la nueva ventana que aparece una vez pulsado Logic Builder. Tal y como puede observarse aparece una lista de autómatas en la pestaña Logic Controller. Para introducir el autómata con el que se quiera trabajar hay que pulsar con el botón izquierdo y arrastrarlo hasta la parte izquierda de la ventana (Figura 12).

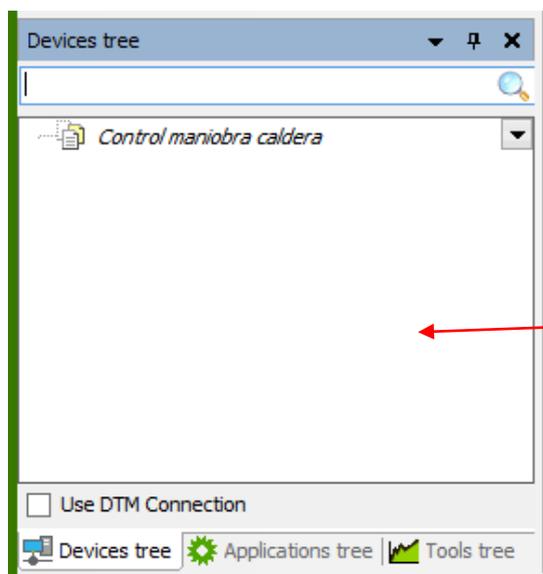


Figura 12: Ventana de elementos

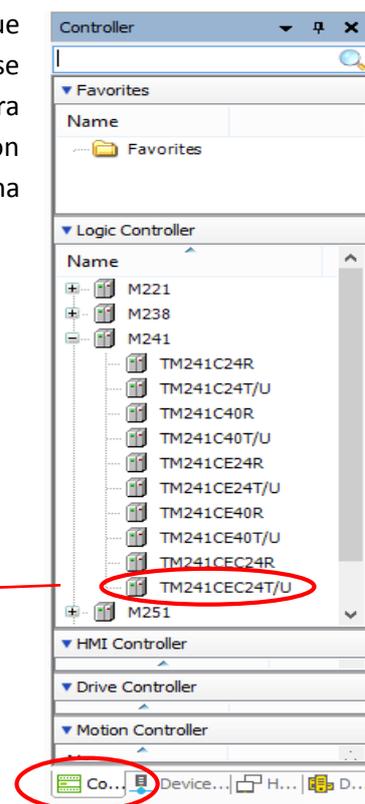


Figura 11: Selección controlador

A partir de este punto ya se podría empezar a escribir el programa, pero en este caso para el control del variador se requiere de una salida analógica que el controlador TM241CEC24T/U no incorpora. Por este motivo hay que repetir el proceso anterior arrastrando esta vez hasta el icono IO_Bus la tarjeta de expansión E/S analógicas mixta TM3AM6/G.

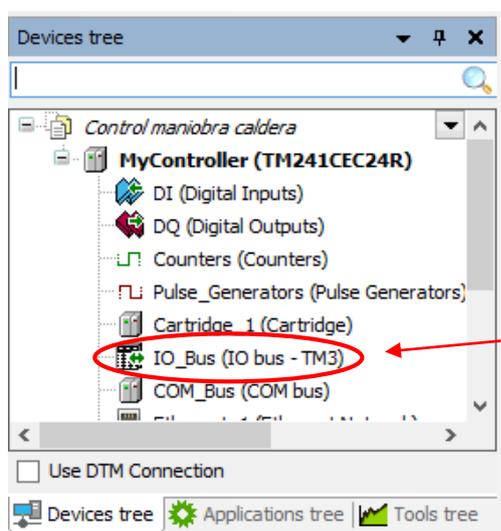


Figura 14: Introducción del módulo

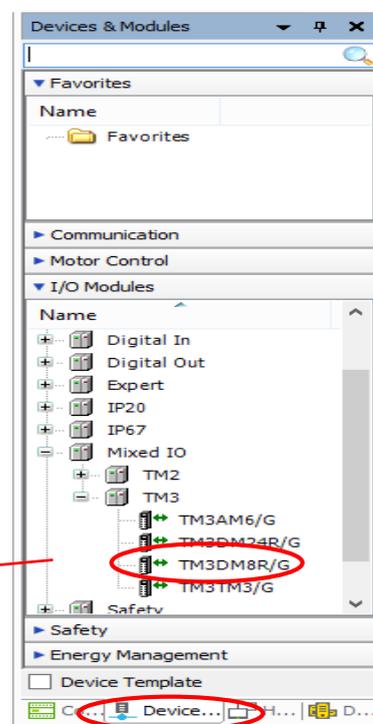


Figura 13: Selección módulo

Notar que los módulos de expansión se encuentran en la pestaña Devices & Modules y no en la pestaña Controllers donde estaban los autómatas. El módulo se arrastra hasta el Bus de entradas y salidas del autómata marcado en la Figura 14.

El programa estará escrito en POU (Unidades de Organización de Programa). Para crearlos simplemente con el botón derecho sobre Application (dentro del cuadro Devices), Add Object, POU.

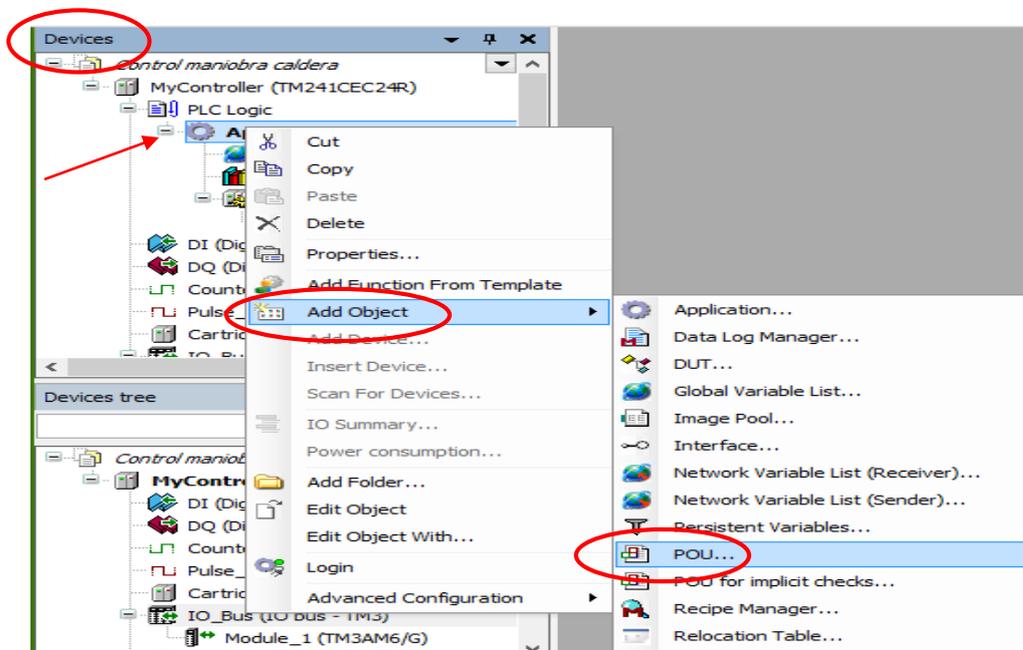


Figura 15: Añadir POU

Una vez seleccionado aparecerá una nueva ventana (Figura 16) donde se puede elegir el tipo de lenguaje a utilizar (estructurado en este caso) para programar el POU, el nombre del mismo (sin espacios) y tipo (siempre elegirán tipo Program en este trabajo). Por último se pulsa en Add.

De ahora en adelante se crearán POU cuando se requiera, ya que trabajando con diversos de ellos se consigue un programa con una estructura más clara que solo utilizando un solo POU. Los pasos a seguir serán los mismos que para crear este POU.

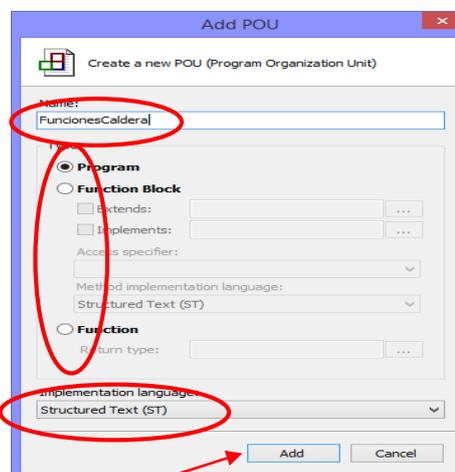


Figura 16: Crear POU

6.1.2. Funciones Caldera

En este primer POU creado se escribirá el sistema de selección entre las distintas funciones que se especifica en las demandas que tenga la caldera: encendido, calefacción, apagado y limpieza. Lo que conseguirá en este POU es que cuando la función encendido se active ninguna otra función esté activa, pero en el momento en que se active cualquier otra (como por ejemplo apagado), entonces se desactiva la función encendido y pasara a estar activa la nueva función.

Cuando se empieza a escribir el programa (en la parte central de la ventana principal de So-Machine) y se utiliza una variable que aún no ha sido creada (en este caso Encender), aparece una ventana que permite crear la variable en ese momento, tal y como se observa en la Figura 17.

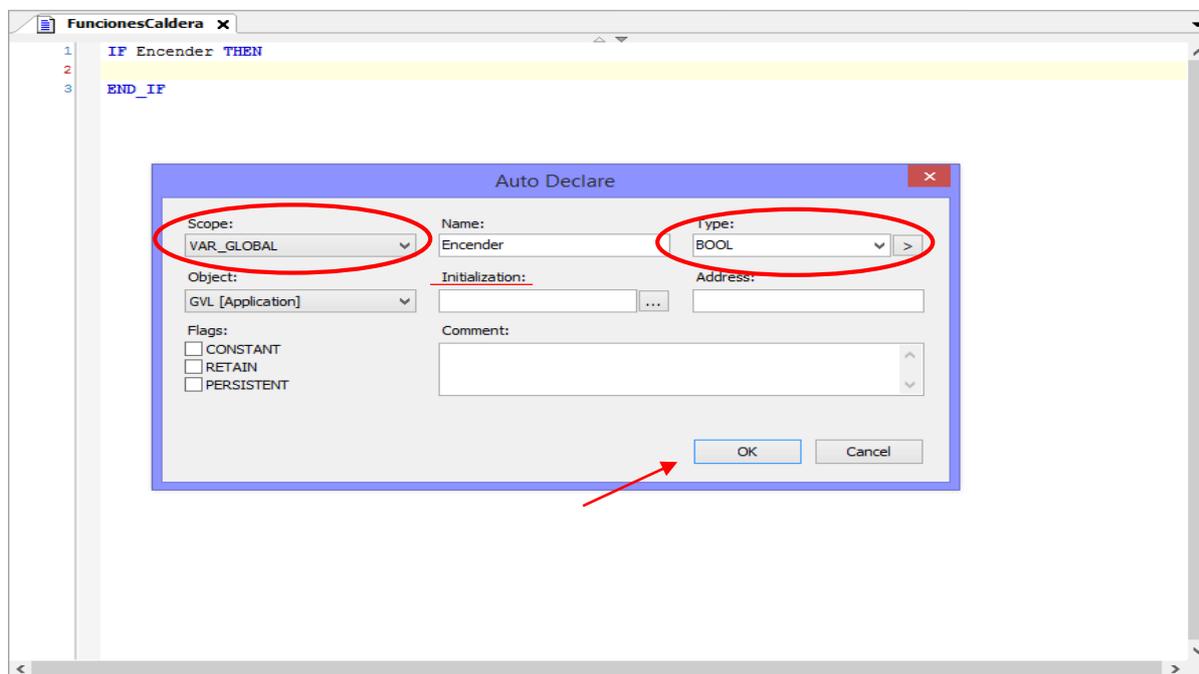


Figura 17: Ventana creación de variables

En esta ventana hay que elegir el tipo y el lugar al que pertenecerá la variable. En este caso se trata de una variable booleana (BOOL) y eligiendo en el apartado Scope la opción VAR_GLOBAL se coloca dentro de GLOBAL. También podría elegirse el valor que ha de tener inicialmente, por defecto todas las variables booleanas están inicializadas a FALSE y las de tipo INT y similares a 0. Todas las nuevas variables, exceptuando el apartado tipo, se declararán siguiendo los mismos pasos.

Se debe elegir el tipo de variable en función de aquello que se quiere representar mediante la variable. Las temperaturas y números para realizar operaciones serán tipo INT o REAL, los temporizadores pueden ser variables tipo TON, TOFF o TP en función del tipo de temporizador, las variables que se utilicen para modificar tiempos en los temporizadores tipo TIME, etc. Hay que prestar atención al tipo de variable cuando se declara ya que de no elegir correctamente el programa dará errores de compilación o no funcionará correctamente.

En la Figura 18 también se ha señalado una pestaña al lado de la función que se está programando. En esta pestaña GVL aparece el listado de todas las variables creadas y que se ha seleccionado que pertenezcan a GLOBAL. En caso de necesitar crear una variable que no aparece escrita en ningún POU (por ejemplo una entrada digital que solo quiere verse por pantalla), y por tanto no se tendrá opción a crearla mediante Auto Declare, en la pestaña GLV se pueden crear siguiendo la sintaxis: *Nombre de la variable: Tipo de variable* (BOOL, INT, WORD, TIME....). Para que aparezca esta pestaña hay que buscarla en el cuadro Devices y dentro de Application hacer doble clic sobre el icono GVL.

El programa escrito en el primer POU es el que aparece en la Figura 18.

Lo que se pretende conseguir con este programa es que si se activa la variable Encender (será un botón en la pantalla), entonces solo se ejecute el programa que realiza la función de encendido (Fencender), mientras que todas las otras funciones estén sin ejecutarse.

La estructura IF Encender THEN se puede interpretar como: SI (está activa la variable) Encender ENTONCES se ejecutan las siguientes líneas de programa hasta llegar al END_IF. Si por el contrario se requiriera que algo ocurriera cuando una variable no se encuentra activa, la estructura a utilizar sería: IF NOT Encender THEN (SI NO esta activa la variable Encender ENTONCES).

El valor que toma una variable Booleana se puede modificar utilizando VARIABLE:= TRUE (activada)/FALSE (desactivada).

```

1  IF Encender THEN
2      Fencender:=TRUE;
3      Fcalefaccion:=FALSE;
4      Fapagar:=FALSE;
5      Flimpiar:=FALSE;
6  END_IF
7
8  IF Calefaccion THEN
9      Fencender:=FALSE;
10     Fcalefaccion:=TRUE;
11     Fapagar:=FALSE;
12     Flimpiar:=FALSE;
13 END_IF
14
15 IF Apagar THEN
16     Fencender:=FALSE;
17     Fcalefaccion:=FALSE;
18     Fapagar:=TRUE;
19     Flimpiar:=FALSE;
20 END_IF
21
22 IF Limpiar THEN
23     Fencender:=FALSE;
24     Fcalefaccion:=TRUE;
25     Fapagar:=FALSE;
26     Flimpiar:=FALSE;
27 END_IF
28

```

Figura 18: Funciones caldera

Lo siguiente será crear cuatro nuevos POU's donde se escribirán los programas que se tendrán que ejecutar cuando se activen las distintas funciones: Fencender, Fcalefaccion, Fapagar y Flimpiar. Los POU's se crearan siguiendo los pasos ya mencionados anteriormente. La explicación del código de cada uno de ellos se hará por separado, empezando primero por el POU que contiene la función de arranque (Figura 19).

6.1.3. Función Encendido

Esta función tiene que provocar el encendido de la caldera tal y como se especifica en el pliego.

Para que se pueda realizar el encendido de la caldera se deben cumplir dos condiciones: que este activa la variable Fencender y que la caldera no estuviera ya encendida. Para comprobar esta última condición hay que utilizar el valor de la temperatura de humos de la caldera que nos aporta una sonda. Por debajo de 60°C consideramos que está apagada y por encima de 60°C encendida. La variable que representará la temperatura de humos será TempHumos y por tanto se declarará tipo INT. En caso de cumplirse estas dos condiciones lo primero es activar un temporizador, mediante la variable Entrada, que permitirá la entrada de combustible en la caldera durante dos minutos. Cuando el temporizador llegue a este tiempo activa la variable TONAlim.Q y cambia a cero el valor de la salida analógica que controla al variador y por tanto no seguirá entrando combustible en la caldera.

```

1  IF Fencender THEN
2
3      IF TempHumos < 60 THEN
4
5          Entrada:=TRUE;
6          TONAlim (IN:= Entrada, PT:= T#120S);
7          Aire:=TRUE;
8          Mecheros:=TRUE;
9          SalAnalog:=10000;
10
11         IF TONAlim.Q THEN
12             SalAnalog:=0;
13         END_IF
14     END_IF
15
16     IF TempHumos >= 60 THEN
17         Entrada:=FALSE;
18         Mecheros:=FALSE;
19         Encender:=FALSE;
20         Calefaccion:=TRUE;
21     END_IF
22 END_IF

```

Figura 19: Función encendido

Al mismo tiempo que se activa el temporizador, habrá dos salidas digitales que encenderán la entrada de aire forzado en la caldera (Aire) y unas resistencias que junto a un ventilador (Mecheros) encenderán el combustible en el interior de la caldera.

El tipo de temporizador es TON: empieza la cuenta cuando se activa su variable de entrada, cuando llega al tiempo indicado activa su variable de salida .Q. Si se desactiva la variable de entrada se reinicia la cuenta y la variable de salida se desactiva también (en caso de haberse llegado a activar). En la Figura 20 se representa su funcionamiento gráficamente donde IN es la entrada que activa el temporizador ET es el avance del tiempo desde cero hasta PT y Q es la variable salida que se activa cuando ET alcanza PT.

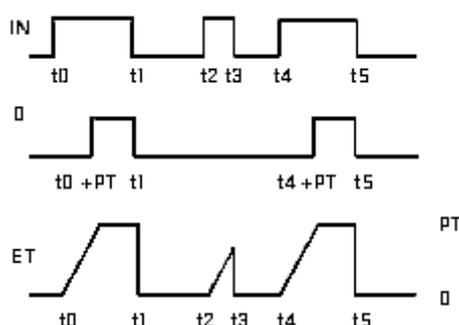


Figura 20: Funcionamiento TON

La sintaxis de la función TON es primero el nombre del temporizador (TONalim declarado como tipo de variable TON) luego indicar la variable que actúa como IN (activa el temporizador) y por último indicar el tiempo que se va a temporizar PT. En la Figura 19 puede verse claramente. Se han elegido dos minutos porque se sabe que a 50Hz el sinfín proporciona 760gr/min de combustible y por tanto girando a 100Hz (más adelante se configura el variador para que a la máxima salida la frecuencia sea de 100Hz) durante 2 minutos se cumple con la aportación mínima necesaria para que se produzca el encendido.

Por último para encender la entrada de aire y los mecheros, se emplearan dos salidas digitales del autómeta (una enlazada con la variable Aire y la otra con Mecheros) que conectadas a relés permitirán la alimentación a 230V de la entrada de aire forzado y las resistencias y ventilador.

En posteriores apartados de este trabajo donde ya se mostrará cómo se conectan los equipos para que puedan comunicarse entre ellos también se explicará cómo configurar una variable para que cuando esté activa también active una salida y como se puede configurar una variable para que tome el valor de una sonda de temperatura.

En el programa anterior puede verse que cuando la caldera ya está encendida (TempHumos > 60°C), se desactiva la variable Mecheros y se activa Calefaccion manteniendo activada la variable Aire. Haciendo esto se consigue que la caldera ya no esté en la función Fencender y pase directamente a la función de control de la temperatura de la nave Fcalefaccion.

6.1.4. Función de calefacción

Una parte del programa de esta función de calefacción, tiene que controlar el arranque y paro de las bombas de agua que impulsaran el agua caliente hasta la nave y la otra parte controlará la frecuencia de giro del motor de alimentación para que se mantenga estable la temperatura del agua de dentro de la caldera. Estas partes se explicaran por separado debido a su extensión.

Circuladoras:

En la Figura 21 aparece el programa que servirá para controlar la circuladora1. El programa de la circuladora2 es idéntico pero cambiando las variables que acaban en "1" por un "2".

```

1  IF Fcalefaccion THEN
2      IF TempACaldera > TempSuf THEN
3          IF Temp1 < TempConsig THEN
4              Banderal:=TRUE;
5          END_IF
6          IF Temp1-2 > TempConsig THEN
7              Banderal:=FALSE;
8          END_IF
9
10         IF Banderal THEN
11             Circl:=TRUE;
12         ELSE
13             Circl:=FALSE;
14         END_IF
15
16         Blink1(enable:= NOT Banderal, timelow:=Tespera, timehigh:=Tfuncion, out=>Circl);
17
18     END_IF
19 END_IF
20 Tespera:=DINT_TO_TIME(iTespera*1000);
21 Tfuncion:=DINT_TO_TIME(iTfuncion*1000);
22
23
24

```

Figura 21: Función circuladora 1

Las circuladoras podrán arrancar cuando se cumplen dos condiciones: está activa la función Fcalefaccion y si la temperatura del agua de la caldera (TempACaldera) está a temperatura suficiente (TempSuf).

Las circuladoras pueden arrancar por dos motivos diferentes:

-En el primer caso las circuladoras tienen que arrancar cuando su correspondiente sonda marque una temperatura (Temp1 o Temp2) inferior a la temperatura de consigna de la nave (TempConsig). Pero esto podría causar que las circuladoras estuvieran arrancando y parando la mayor parte del tiempo (arrancarían cuando la temperatura es 0.1°C inferior a la consigna y pararían cuando fuera 0.1°C superior). Para que esto no ocurra, ya que este funcionamiento tan intermitente podría acabar por estropearlas, lo que se utiliza es una variable auxiliar llamada Bandera. Esta variable se activa cuando la temperatura es inferior a temperatura consigna y se desactiva cuando la temperatura de la sonda -2°C es superior a la temperatura consigna (la temperatura que marca la sonda esta dos grados por encima de la consigna).

Una vez escrito lo anterior en el programa se enlazará esta variable Bandera a la variable que actuará sobre la salida digital que provocará el arranque de las circuladoras (Circ1, Circ2) mediante la estructura IF... ELSE. Esta estructura tal y como se ha mencionado se interpreta de la siguiente manera: SI (IF) Bandera1 activa ENTONCES (THEN) Circ1 se activa (:=TRUE), SINO (ELSE) Circ1 desactivada (:=FALSE).

-El arranque de las circuladoras también ha de producirse cuando estas llevan mucho tiempo paradas, el motivo está explicado en el apartado Especificaciones del Control. Para hacer esto se ha utilizado la función Blink.

A esta función hay que introducirle una variable para que se active (en este caso la función Blink1 o Blink2 se activarán cuando NO estén activas las variables Bandera1 o Bandera2 respectivamente), un tiempo donde la variable de salida estará desactivada, otro tiempo que indicará el tiempo que la variable de salida estará activa y la propia variable de salida (Circ1 o Circ2).

Cuando aparece la ventana de auto declaración de la variable Blink1 se elegirá tipo de variable Blink.

Por ejemplo, su funcionamiento para la circuladora1 será el siguiente: cuando no este activa la variable Bandera1 la variable Circ1 estará desactivada durante un tiempo Tespera y cuando haya transcurrido este tiempo, se activara Circ1 durante un tiempo Tfuncion. Si en algún momento se activa Bandera1 porque ha bajado la temperatura en la nave, entonces el ENABLE del Blink no se cumplirá y se omitirá la función.

Podría pensarse que al estar la variable Circ1 enlazada a la variable Bandera1 (cuando Bandera1 está desactivada/activa la variable Circ1 también está desactivada/activa) la función Blink no funcionará. Lo cierto es que la función no funcionaría si se escribiera el programa de la forma siguiente:

```

ControlTemp x GVL
1  IF Fcalefaccion THEN
2      IF TempACaldera > TempSuf THEN
3          IF Temp1 < TempConsig THEN
4              Bandera1:=TRUE;
5          END_IF
6          IF Temp1-2 > TempConsig THEN
7              Bandera1:=FALSE;
8          END_IF
9
10         Blink1(enable:= NOT Bandera1, timelow:=Tespera, timehigh:=Tfuncion, out=>Circ1);
11
12         IF Bandera1 THEN
13             Circ1:=TRUE;
14         ELSE
15             Circ1:=FALSE;
16         END_IF
17     END_IF
18 END_IF
19 Tespera:=DINT_TO_TIME(iTespera*1000);
20 Tfuncion:=DINT_TO_TIME(iTfuncion*1000);
21
22
    
```

Figura 22: Ejemplo de posible error

El programa escrito de esta forma sobrescribiría la variable Circ1 en función del valor que tenga Bandera1. De esta forma si Bandera1 no está activada, aunque la función Blink1 active la variable Circ1, como en la línea siguiente se sobrescribe el valor de Circ1 en función del valor de Bandera1 el programa la variable Circ1 tomara el valor FALSE. Esto no puede ocurrir si el programa está escrito tal y como aparece en la Figura 21, ya que a la inversa la función Blink1 no se activa si la variable Bandera1 está activa.

Se requería que los tiempos de estado de parada y de funcionamiento de la función Blink fueran modificables por pantalla. Por este motivo se necesita de las líneas del programa:

```
Tespera:= DINT_TO_TIME (iTespera*1000)
Tfuncion:=DINT_TO_TIME (iTfuncion*1000)
```

La variable iTespera es una variable tipo DINT que será posible modificar por pantalla, pero la función Blink requiere variables de tipo TIME. Por este motivo se utilizan estas líneas para que las variables tipo TIME (Tespera y Tfuncion) tomen el valor de las correspondientes variables enteras que introduciremos por pantalla (iTespera y iTfuncion). Las variables introducidas por pantalla han de multiplicarse por mil para que se cumpla que si la variable iTespera=3600 el valor de Tespera sea de 3600s. Este es el motivo por el que se requieren de variables DINT y no simplemente INT. Las variables INT solamente pueden tomar valores inferiores a 32.767, valor que se supera si se requiere de un tiempo superior a 33s el cual es muy inferior a los tiempos que se pretenden dar.

Una vez establecido el sistema de arranque y paro de las circuladoras, se explicará la siguiente parte de la función Fcalefaccion que mantendrá a una temperatura que se indique por pantalla el agua en el interior de la caldera.

Control PID:

Lo que se utilizará para conseguir esto es un control por PID. En este caso a pesar de que se pueden utilizar funciones en las que solamente hace falta indicar entradas y salidas a un bloque de funciones para implementarlo, el PID se realizará “manualmente”. Para ello primero se normalizan todas las variables involucradas en el PID como un “%” de un valor de referencia. En este caso lo que hay que normalizar son las temperaturas, por ello se ha elegido una temperatura de referencia de 100°C.

La variable ValTempMax es la temperatura de referencia. Consig es el % de la temperatura de referencia al que se quiere que llegue la variable ValActual. Esta última variable representa el % de la temperatura de referencia al que se encuentra en un determinado momento la temperatura del agua de la caldera. Por ejemplo, si arranca la caldera y a TempCCaldera se le da un valor de 40°C y en ese momento TempACaldera tiene un valor de 20°C, la consigna valdrá un 40% de la referencia y el valor actual un 20%.

```
47 ValTempMax:=100;
48 Consig:=TempCCaldera*100/ValTempMax;
49 ValActual:=TempACaldera*100/ValTempMax;
50
```

Figura 23: Normalización variables

Una vez normalizadas las variables hay que implementar la estructura del PID.

Es innecesario que se ejecuten todas las operaciones del PID en cada ciclo ya que prácticamente no se modifican las variables. Por este motivo se utiliza un temporizador (TONPID) que ejecutará las operaciones una vez transcurra el tiempo que se especifique por pantalla TPID.

Un PID consta de tres errores, el proporcional (Err), el integral (ErrI) y el derivativo (ErrD).

```

51  TONPID (IN:= NOT TONPID.Q, PT:= TPID);
52  IF TONPID.Q THEN
53
54  Err:= Consig-ValActual;
55  ErrI := ErrI+Err;
56  ErrD := Err-ErrAnt;
57  ErrAnt:= Err;
58  ErrT:=(Err*(Kp/100)+ErrI*(Ki/100)+ErrD*(Kd/100));
59  END_IF
60
61  SalAnalog:=REAL_TO_INT((10000/100)*ErrT);

```

Figura 24: Implementación PID

El error proporcional es la diferencia entre el valor al que se quiere que llegue una variable y el valor que actualmente tiene y por tanto este error es el que hay que intentar que se mantenga a 0.

Si solo se utilizara el error proporcional, cuando se alcanzara la temperatura deseada y hubiera error=0 se detendría el sistema de alimentación. Pero como el sistema está continuamente cediendo calor tendría que volver a arrancar hasta que se detuviera otra vez, estaría arrancando y parando el sistema de alimentación todo el tiempo. Por este motivo es necesario el error integral. Este error toma como valor su propio valor (en el ciclo anterior) más el nuevo error que exista entre la consigna y el valor actual. En caso de que el valor actual supere a la consigna, la variable Err tomará valores negativos y reducirá el valor del error integral. De esta forma el error integral acabará estabilizándose manteniendo constante la entrada de alimentación y a cero el error entre la consigna y el valor actual.

En este caso, la parte derivativa también es importante. Este sistema es de respuesta lenta, es decir, el tiempo que se tarda desde que se modifica una salida hasta que se recibe una respuesta del sistema no es despreciable. Por este motivo aunque el error proporcional se mantenga constante, el error integral aumentará mucho antes de que el sistema llegue a responder. Para evitar una respuesta demasiado brusca, una vez el sistema empiece a responder, el error derivativo tomará un valor contrario al valor del error integral y proporcional.

Un ejemplo de funcionamiento podría ser el siguiente (tomando un tiempo de 1s en el temporizador): en el primer ciclo si la diferencia entre la consigna y el valor actual es de 20°C el error proporcional tomara un valor de 20% de la temperatura de referencia (al estar normalizadas las variables Consig y ValActual). El error integral también tomará un valor del 20% mientras que el derivativo seguirá con el valor cero. A cada segundo que pase el error proporcional y derivativo se mantienen constantes, pero el integral sigue sumando a su valor anterior 20%. Cuando el sistema responda, aumentando por ejemplo en 0.5°C la temperatura del valor actual respecto al anterior, el error proporcional valdrá 19.5%, el integral seguirá aumentando pero más lentamente (ahora suma cada segundo 19.5% y no 20%) y el derivativo al ser el nuevo error (Err) menor que el anterior (ErrAnt) tomará el valor de -0.5%. Es evidente que si por ejemplo han transcurrido 30 segundos desde que se ha activado el funcionamiento del PID hasta que el sistema empieza a responder, el valor del error derivativo será despreciable frente al valor del error integral. Es por eso que se utilizan los coeficientes Kp, Ki y Kd.

Si se multiplica por una Ki de valor 0.1 un error integral que llegue a valer 600% mientras que el error derivativo con un valor -0.5% se multiplica por una Kd con valor 20 se consigue que el error derivativo tenga un mayor peso en el control del sistema al mismo tiempo que se reduce la importancia del error integral.

La variable ErrT es el resultado de la suma de todos los errores multiplicados por sus correspondientes constantes. En función del valor de esta variable, la salida analógica que acelera el sistema de alimentación tendrá un valor u otro: si el error total equivale a un 60% de la temperatura de referencia, entonces la salida tomará como valor un 60% del total (valor máximo de la salida 10000).

Las variables Kp, Ki y Kd se han declarado como enteras y es posible modificar su valor por pantalla. Al estar declaradas como enteras, aunque por pantalla se introduzca Kd=2.50 (si se eligen dos decimales) el valor que tomará en el programa será de 250. Por este motivo estas variables están divididas entre cien. Los errores pueden tomar valores no enteros y por este motivo se declaran como variables reales (REAL). Sin embargo, la variable que esta enlazada con la salida analógica tiene que ser una variable entera (esto se explicara en el apartado de conexiones) y por este motivo se necesita de la función REAL_TO_INT que transforma la operación que contiene la variable ErrT (REAL) en un resultado entero.

Las dos últimas funciones son más simples. La función de limpieza solo puede ejecutarse si la caldera esta parada y simplemente activa un sinfín que sirve para extraer la ceniza del interior de la caldera. La de parada desactiva las variables de entrada de aire, alimentación, circuladoras y mecheros.

6.1.5. Funciones limpieza y apagado

Tal y como puede observarse en el POU, el sinfín (variable SinfinCeniza) solo se activará si esta activa la función Flimpiar (podrá activarse desde la pantalla con la variable Limpiar) y si la temperatura de humos está por debajo de 50°C (caldera apagada). El sinfín se parará cuando se cambie de función, ya sea a la de Fapagar o Fencender. Cuando se active esta función se pondrá a cero el valor de la alimentación de la caldera.

El sistema para hacer funcionar el sinfín será el mismo que para la entrada de aire y los mecheros. Con un relé que permita el paso de corriente cuando se active la variable SinfinCeniza.

```

1  IF Flimpiar THEN
2      SalAnalog:=0;
3      IF TempHumos < 50 THEN
4          SinfinCeniza:=TRUE;
5      ELSE
6          SinfinCeniza:=FALSE;
7      END_IF
8  END_IF
9
10 IF NOT Flimpiar THEN
11     SinfinCeniza:=FALSE;
12 END_IF
    
```

Figura 25: Función limpieza

Las variables Aire y la señal analógica del variador han de desactivarse para que se produzca el apagado de la llama en la caldera. Las variables de las circuladoras y mecheros, han de sobrescribirse a falso porque podrían quedarse activas si se cambia de la función en la que ellas estén activas a la función de parada.

```

1  IF Fapagar THEN
2      SalAnalog:=0;
3      Aire:=FALSE;
4      Mecheros:=FALSE;
5      Circ1:=FALSE;
6      Circ2:=FALSE;
7  END_IF
    
```

Figura 26: Función apagado

6.1.6. Funciones restantes del control

Para cumplir con todas las demandas falta que el programa haga las siguientes cosas:

- Variable que indique el consumo por unidad de tiempo de combustible.
- Mostrar el valor de una variable que represente el nivel de trabajo al que está sometida la caldera en función de la alimentación.
- Establecer rangos de diferencia de temperaturas entre la exterior y la del agua de la caldera para modificar la temperatura del agua y compensar pérdidas de calor.

Todo ello se incluirá en un único POU y se realizará una pequeña modificación en la función de control de la temperatura del agua de la caldera para incluir la compensación de temperatura.

Sabiendo que el sinfín mueve a 50Hz 760gr/min, que la frecuencia de trabajo del variador esta entre 0 y 100Hz y que el pellet tiene una energía de 4.88KWh/Kg es fácil realizar el programa conseguir los dos primeros puntos.

No se conoce la frecuencia del variador, pero sí se sabe que es directamente proporcional a la salida analógica del autómatas que toma valores entre 0 y 10000 y que por tanto 50Hz equivalen a un valor de la variable analógica de 5000. Como se quiere que el consumo tenga unidades de Kg/h hay que pasar los gramos a kilogramos y los minutos a horas, quedando la ecuación:

$$\text{Consumo} = \frac{\text{SalAnalog}}{5000} \times 760 \frac{60}{1000}$$

Ecuación 5: Cálculo del consumo instantáneo

Para calcular la potencia a la que está trabajando la caldera hay que multiplicar el consumo por la energía por kilogramo que proporciona el pellet. Esta potencia tendrá unidades en KW, para poder comparar esta potencia con la potencia de la caldera (600000Kcal/h) hay que multiplicar por 3600 para pasar de Kj/s a Kj/h y dividir por 4.18 para tener Kcal y no Kj.

$$\text{Potencia} = \text{Consumo} \times \frac{3600}{4.18}$$

Ecuación 6: Cálculo potencia instantánea

El programa quedaría de la siguiente forma:

```

17 Consumo:=(SalAnalog*760*60/1000)/5000;
18
19
20 PotCald:=Consumo*EngPellet*3600/4.18;
21
22 NivelTrabajo:= PotCald/600000*100;

```

Figura 27: Valores numéricos a monitorizar

Siendo todas las variables tipo reales y que la variable que representa el poder calorífico del pellet sea posible modificarla por si se cambia el material de alimentación en el sistema.

Para la compensación de la temperatura se utilizaran tres nuevas variables dos de ellas modificables por pantalla. La variable DifComp representa el valor de la diferencia de temperaturas entre la temperatura de consigna de la caldera y la temperatura exterior a partir de la cual habrá que empezar a compensar. Y GradosComp es el valor que se quiera incrementar la temperatura del agua.

```

3  IF TempCCaldera-TempExt > 2*DifComp THEN
4      IncGrados:=2*GradosComp;
5  END_IF
6
7  IF TempCCaldera-TempExt < 1.5*DifComp THEN
8      IncGrados:=GradosComp;
9  END_IF
10
11 IF TempCCaldera-TempExt < DifComp THEN
12     IncGrados:=0;
13 END_IF

```

Figura 28: Función compensación

Para que la compensación funcione, al valor que toma la variable TempAcaldera (proporcionado por la sonda) se le restara el valor de la variable IncGrados. De esta forma si por ejemplo las variables toman los valores DifComp=10 y GradosComp=5, cuando la diferencia entre la temperatura de consigna de la caldera y la temperatura exterior sea menor de 10°C el valor de la temperatura del agua no sufrirá ninguna modificación. Pero si esta diferencia se incrementara en más de 10°C (pero menos de 20°C) entonces IncGrados tomaría el valor de 5, provocando que a pesar de que la temperatura real del agua es de 40°C, al restarle este valor el programa entendería que el agua se encuentra a 35°C y no a 40°C.

6.2. Programación de la Pantalla Táctil

La monitorización y control del sistema se realizará con los siguientes paneles:

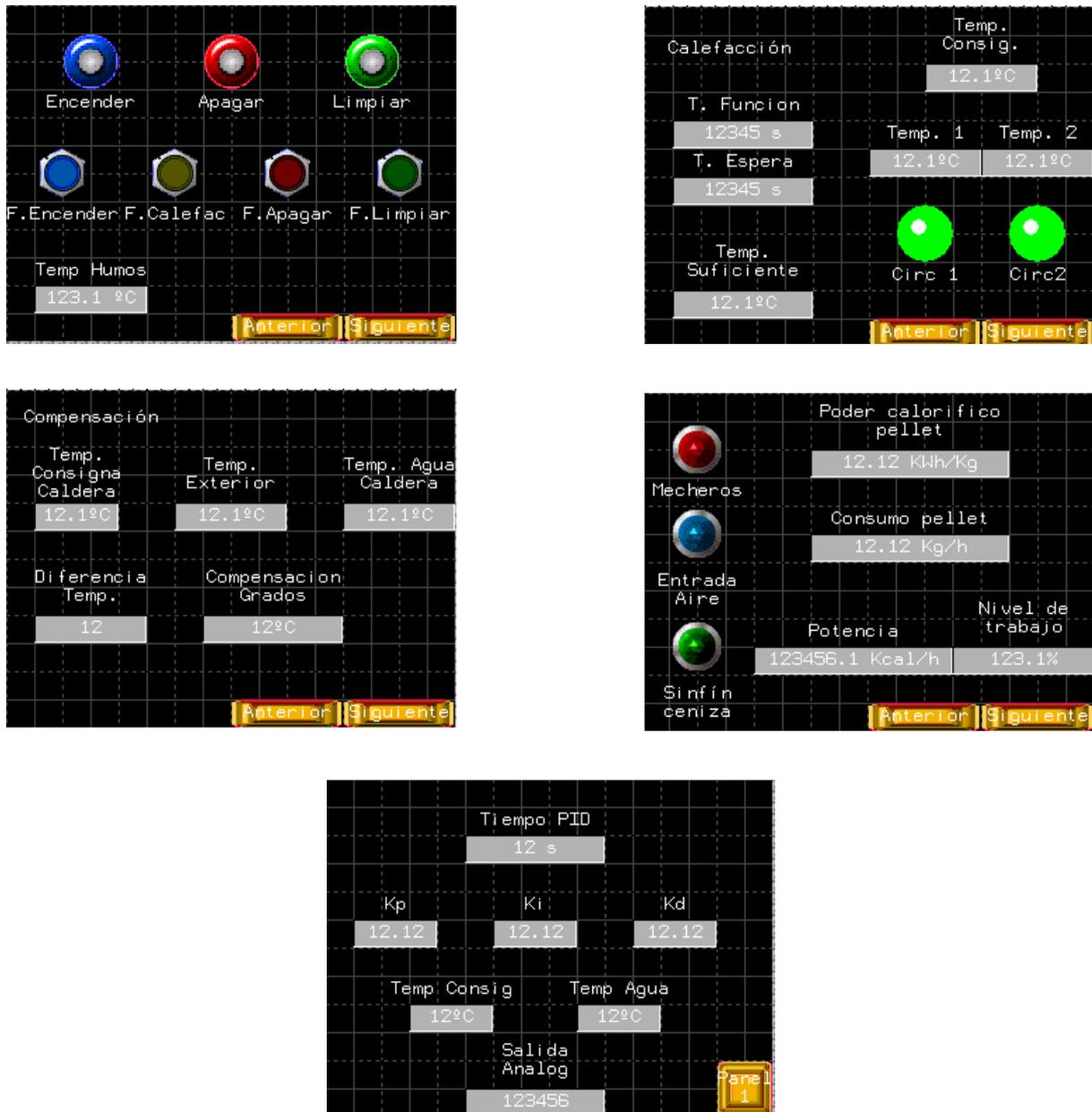


Figura 29: Paneles del control

En las siguientes páginas se mostrará cómo se ha procedido para crear estos paneles y la explicación de cada uno de los elementos que los componen.

6.2.1. Nuevo proyecto

Para empezar a programar el panel primero hay que incorporarlo al proyecto tal y como se incorporó el autómata.

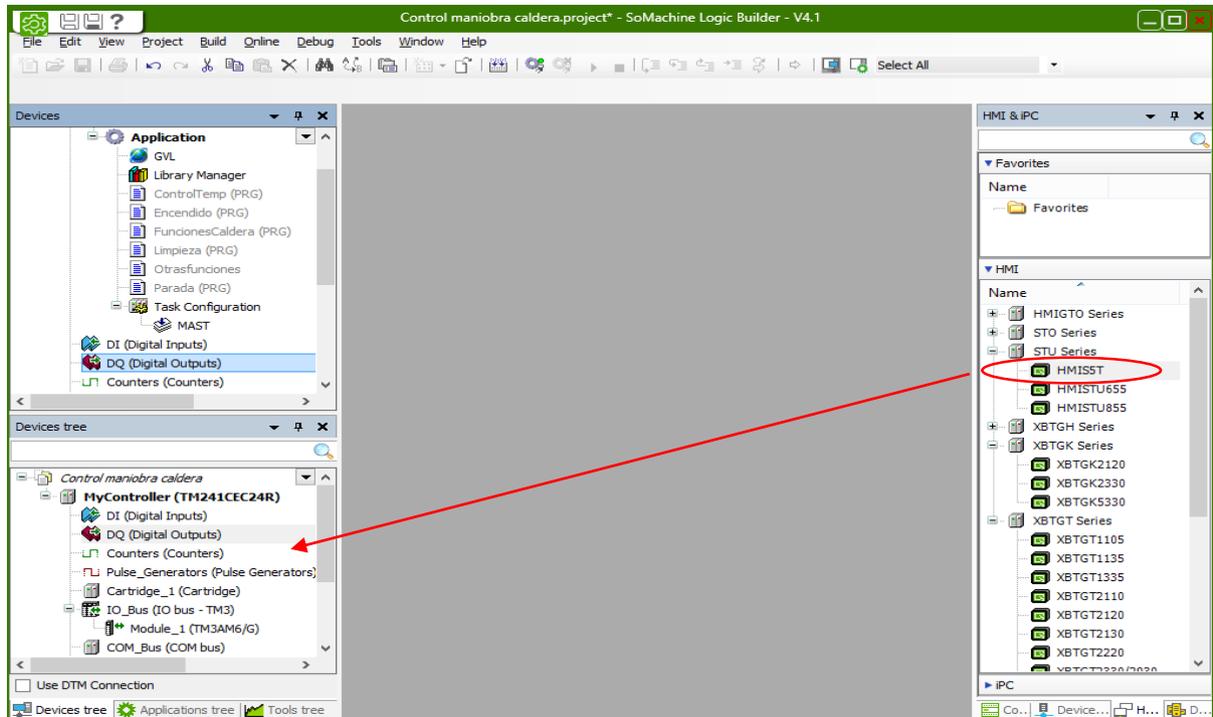


Figura 30: Introducir panel HMIS5T

En la parte derecha de la ventana principal, dentro de la pestaña HMI y se arrastra el panel que corresponda (HMIS5T en este caso) hasta el recuadro de la esquina inferior izquierda.

Una vez hecho esto es necesario hacer un paso intermedio antes de pasar al programa utilizado para programar el panel (Vijeo-Designer). Este paso consiste en introducir una Configuración de Símbolo (Symbol Configuration) dentro de Application. Haciendo esto se podrán elegir variables que forman parte del programa del autómata y utilizarlas en la programación del panel.

Una vez se ha pulsado en la configuración de símbolo (Figura 31) aparecerá una nueva ventana donde simplemente ha de pulsarse Add.

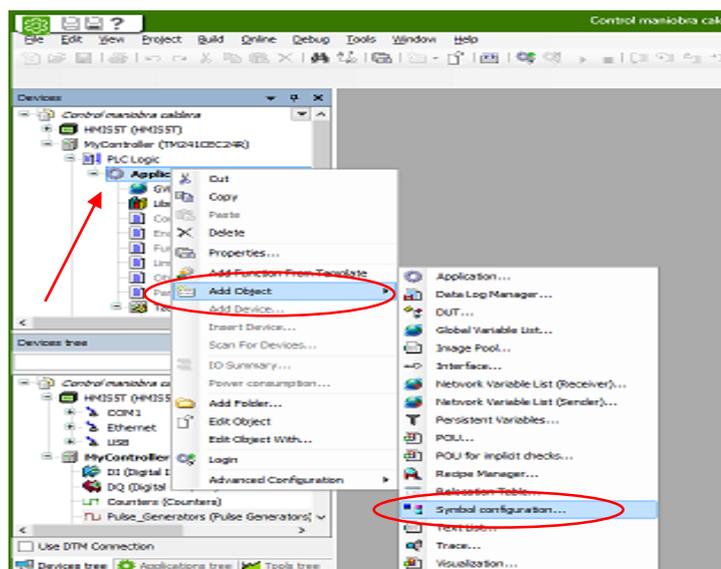


Figura 31: Configuración de símbolos

Ahora para que dentro de configuración de símbolo aparezcan todas las variables que se han ido introduciendo en Global hay que introducir todos los POU's dentro de MAST. Esto, además de cumplir con el propósito anterior también es necesario para introducir las funciones dentro del bucle de ejecución del autómatas. Es decir, si no se hace esto aunque se descargue el programa al autómatas, este no realizará ninguna acción.

Para introducir los POU's en MAST simplemente se seleccionan con el botón izquierdo y se arrastran hasta MAST. Es importante tener en cuenta el orden con el que se introducen. Tal y como se ha mencionado en la definición del autómatas, este ejecuta de forma secuencial sus instrucciones y por tanto también las funciones al ser conjuntos de instrucciones. Si dentro de una función hay una instrucción que se quiere evitar que sea sobrescrita es recomendable ponerla en último lugar.

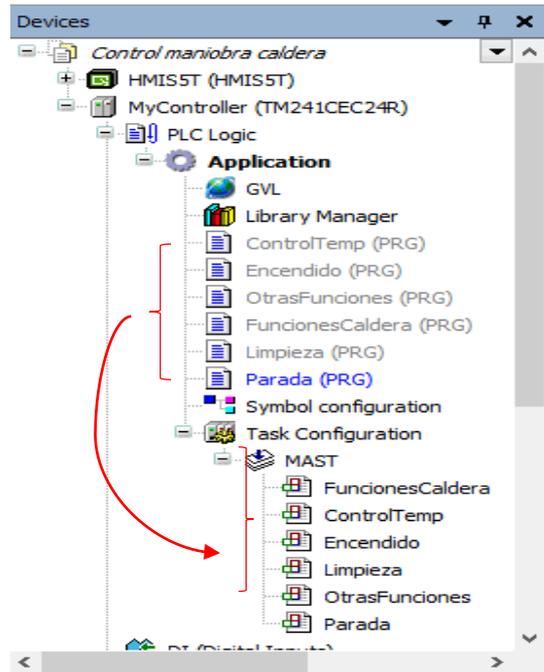


Figura 32: Funciones a tarea MAST

Una vez hecho esto pulsando en configuración de símbolo y en el cuadro central de la ventana principal se puede ver un desplegable con nombre GVL. Si se extiende, aparecen todas las variables del programa que haya en MAST. Si se crea una nueva variable habría que refrescar, pulsando build, para que apareciera en la lista.

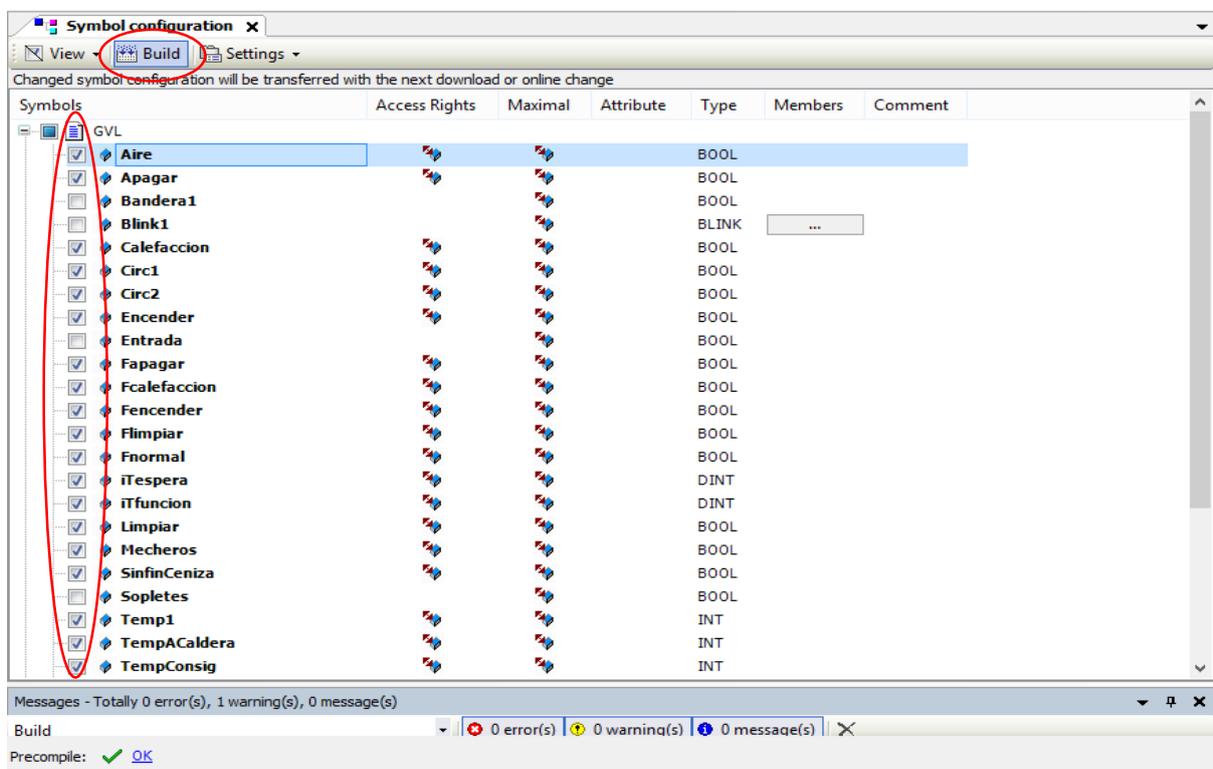


Figura 33: Traspaso de variables

Una vez extendido GVL se puede observar un cuadro al lado izquierdo de cada variable utilizado para seleccionar que variables se quieren transferir al programa de la pantalla. En este caso se seleccionaran todas aquellas que se quiera mostrar y/o modificar por pantalla.

Ahora ya se puede empezar a trabajar con el Vijeo-Designer. Para pasar del So-Machine al Vijeo, simplemente en la parte izquierda de la barra superior del So-Machine aparece al pasar el puntero por encima la opción de Quick Toolswitch donde es posible seleccionar el Vijeo-Designer.

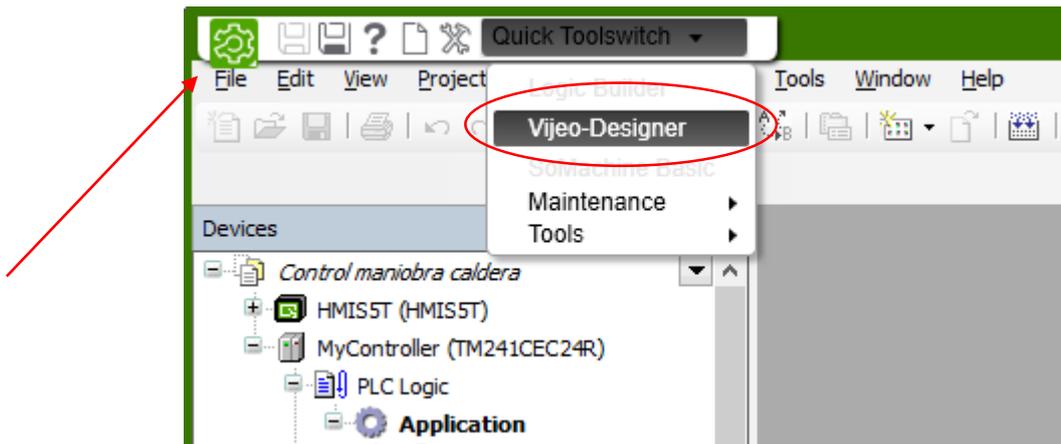


Figura 34: Cambio a Vijeo-Designer

La ventana principal del Vijeo-Designer es la que aparece en la Figura 35.

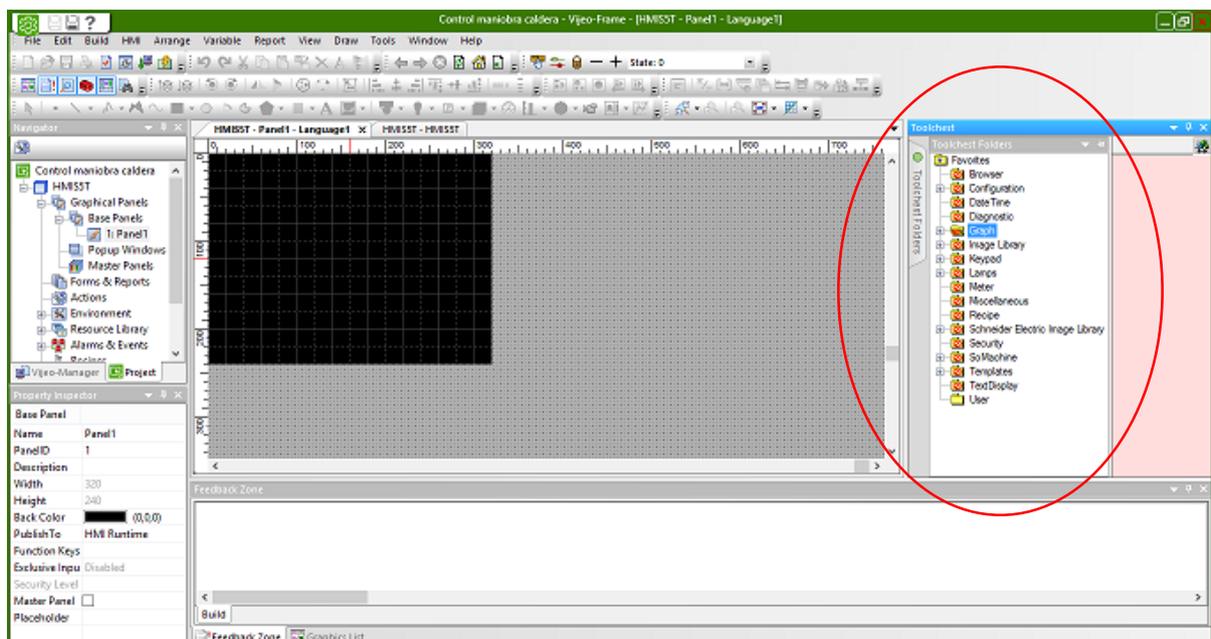


Figura 35: Ventana principal Vijeo-Designer

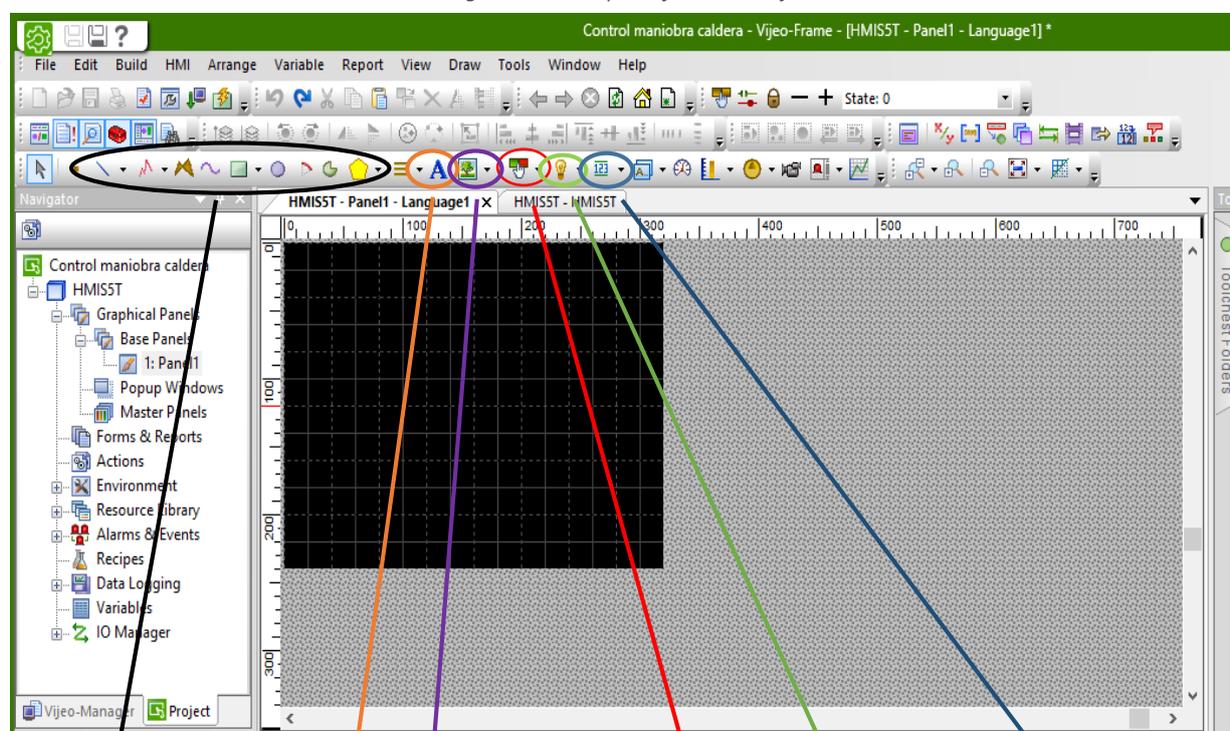
A la derecha de esta ventana aparece una pestaña llamada Toolchest Folders. Dentro hay un conjunto de carpetas que contienen diversas imágenes prediseñadas que podrían utilizarse para que aparezcan en la pantalla, por ejemplo un depósito, una bomba de agua, etc...

Pero en este caso tal y como puede apreciarse, la pantalla del panel HMISST es pequeña y por tanto solamente se introducirá lo estrictamente necesario para que se pueda realizar el control requerido.

En la figura de esta página se señalan algunas de las funciones más simples que pueden introducirse en la programación de los paneles. Las más utilizadas en este trabajo son la introducción de botones, pilotos, valores numéricos y texto.

Para introducir los botones hay que seleccionar el icono, marcado en rojo, en la Figura 36 y se elige al ponerlo en la pantalla el tamaño que tendrá.

Figura 36: Principales funciones Vijeo



Introducir Formas Geométricas

Introducir Cuadros de Texto

Introducir Capturas de Pantalla

Introducir Botones

Introducir Valores Numéricos

Introducir Pilotos

6.2.2. Programación del primer panel

Este panel permitirá el cambio entre funciones y la monitorización del estado de la caldera así como la temperatura indicada por la sonda de humos.

Una vez puesto aparece una ventana donde se tendrá que configurar el botón (Figura 37).

Lo primero es elegir el tipo de botón en el apartado Operation. En este caso se requiere de un botón del tipo bit y temporalmente activado, ya que la variable (por ejemplo Encender) es booleana y al activarse por un momento el programa escrito en FuncionesCaldera activará la variable Fencender. Para hacer esto se despliegan las opciones en Operation y se elige la opción de bit. La parte central de la ventana cambiará y aparecerán las siguientes opciones: set, reset, toggle, momentary ON y momentary OFF. Hay que marcar la opción Momentary ON.

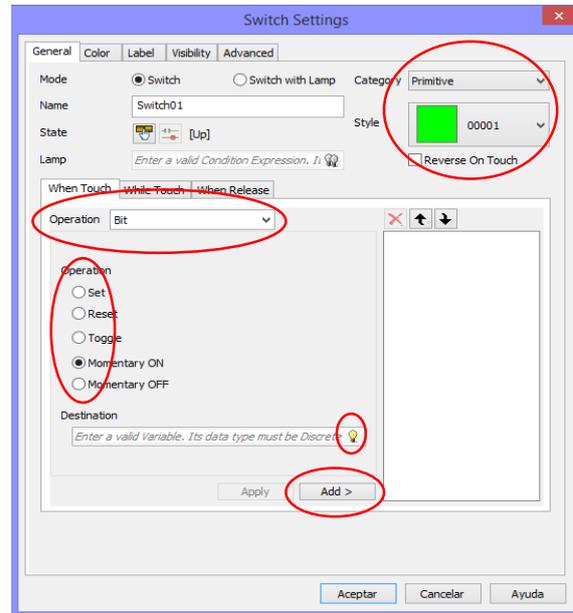


Figura 37: Ventana configuración botón

Ahora hay que enlazar la variable con el botón. Para ello hay que pulsar en la bombilla que aparece debajo de Destination marcada en la Figura 37. En la nueva ventana hay que buscar la variable dentro de la pestaña SoMachine, donde se encuentran todas las variables que se han transferido en la configuración de símbolos. Una vez seleccionada hay que pulsar en OK y en la ventana de la Figura 37 hay que pulsar Add> para que la variable se active con el botón.

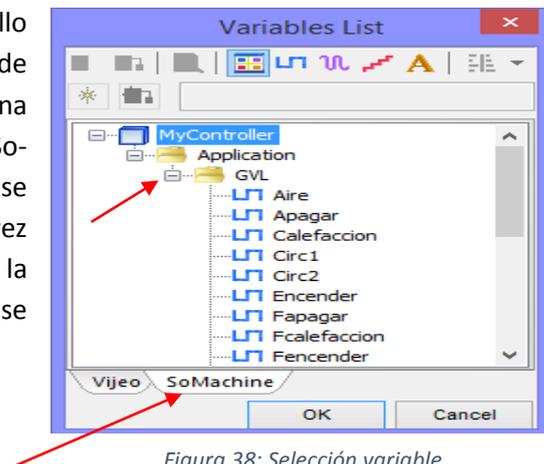


Figura 38: Selección variable

Por último y solamente con carácter estético, con las opciones de category y style, que aparecen en la esquina superior derecha en la Figura 37, se puede elegir el dibujo o la forma que tendrá el botón. Dentro de la opción Bitmap pulsando en Browse, aparecen un gran número de opciones posibles.

Repitiendo los pasos anteriores se crean los botones para las variables restantes: Apagar y Limpiar.

Para configurar el botón que se introducirá para cambiar de panel ha de elegirse la opción Panel en Operation y no Bit.

Pero antes de poder crear este botón es necesario que en el proyecto haya más de un panel. Para introducir un nuevo panel hay que pulsar con botón derecho sobre Base Panels y elegir New Panel.

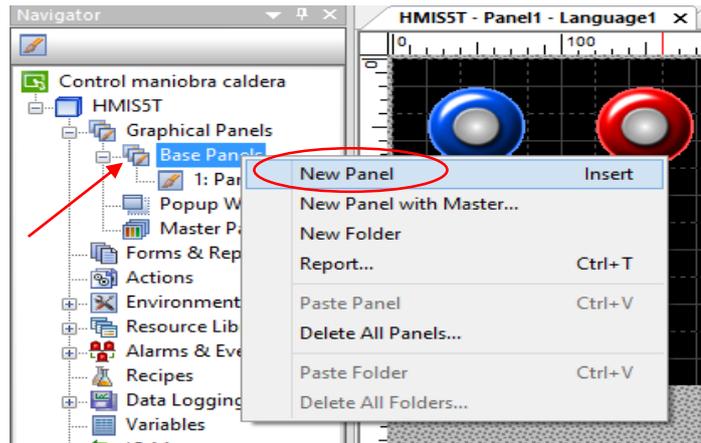


Figura 39: Insertar nuevo panel

Una vez introducido el botón y elegido el tipo de operación (panel) en Panel ID: se introduce el nombre del panel al que se quiere pasar cuando se pulse ese botón y seguidamente hay que pulsar en Add. En category se puede modificar su apariencia.

En todos los botones pueden aplicarse diversas configuraciones, por ejemplo que solo sean visibles cuando una variable se active, que les aparezca un letrero...

Todo esto puede configurarse mediante las pestañas en la parte superior de las ventanas para configurar los botones.

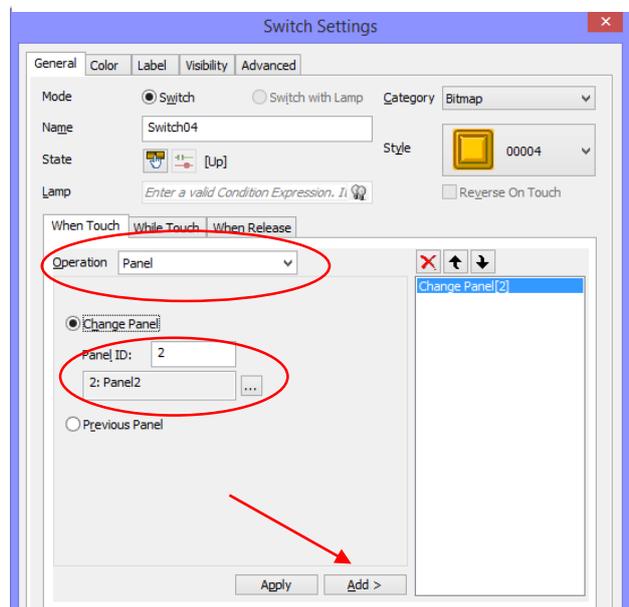


Figura 40: Botón cambio de panel

Para introducir pilotos, se utiliza el icono con forma de bombilla marcado en verde en la Figura 36 y al igual que para un botón se indica el tamaño clicando sobre el panel, extendiendo y volviendo a clicar sobre el panel. En la ventana que ahora aparece hay que seleccionar la variable que se quiera enlazar con el piloto (el piloto estará encendido cuando la variable se active) y la apariencia del mismo igual que en los botones. Pulsando en State puede probarse la forma con la que se ve el piloto cuando esta encendido y apagado.

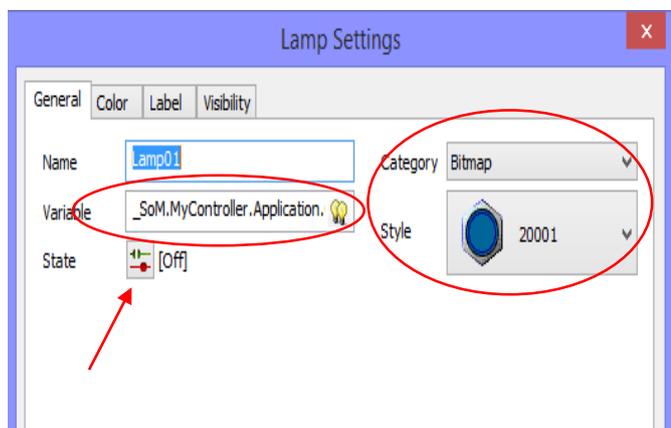


Figura 41: Configuración piloto

Este proceso de configuración es el mismo para todos los pilotos con los que se quiera visualizar una variable booleana.

Ahora se utilizará el icono señalado en azul en la Figura 36 para poder visualizar la variable TempHumos en el panel. La ventana que aparece se tiene que seleccionar la variable y el tipo (INT o FLOAT), el número de cifras que puede tener el número (en display digits), las unidades y la posibilidad de introducirse o modificarse por pantalla o no (enable input mode).

Para poder realizar las comprobaciones pertinentes será necesario poder modificar las temperaturas manualmente, ya que no se dispone de la instalación real, para ver cómo responde el programa del autómeta. Por este motivo, todas las variables del tipo temperatura será posible modificarlas por pantalla aunque en el sistema real simplemente tomarían el valor que marcara la sonda de temperatura.

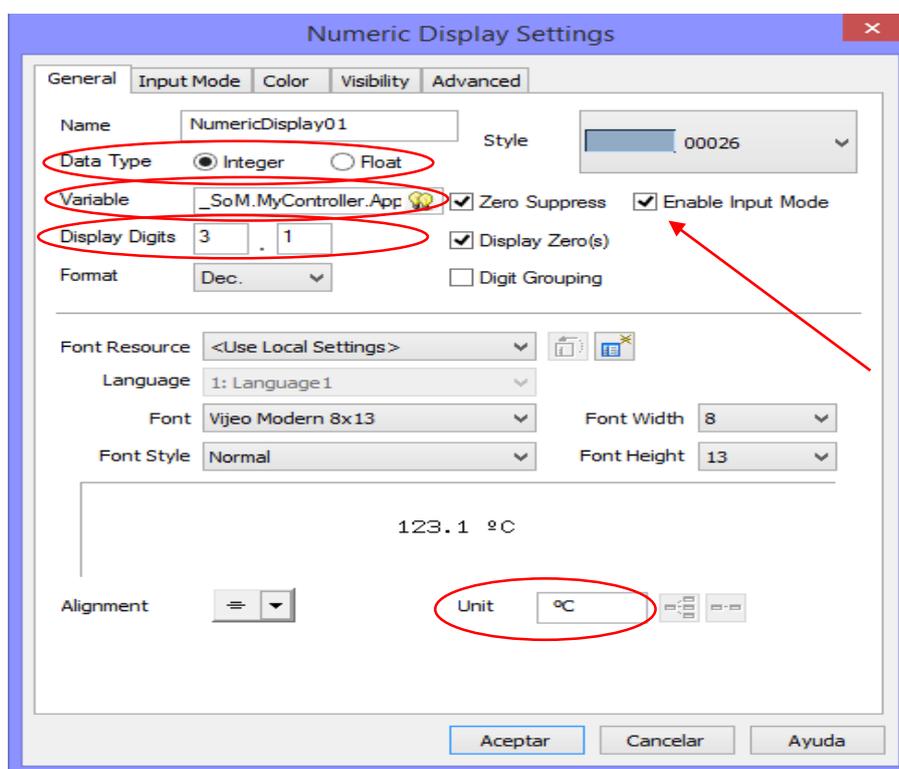


Figura 42: Configuración panel numérico

Por último y para que en el panel este todo más claro se introducirán cuadros de texto con el nombre de cada botón, piloto y número mediante el icono marcado en naranja en la Figura 36. El resultado es el que aparece en la Figura 43.



Figura 43: Primer Panel

Los siguientes paneles se van a realizar siguiendo pasos similares a los seguidos para crear el primer panel. El programa Vijeo-Designer, también permite crear dibujos utilizando formas geométricas, marcadas en negro en la Figura 36, o introducir imágenes mediante recortes de pantalla mediante el icono marcado en morado en la misma Figura. Esta última función permite introducir, por ejemplo, una foto de un equipo que se haya hecho con una cámara digital en la pantalla.

6.2.3. Programación segundo panel

En el panel 2 (Figura 44) pueden modificarse y visualizarse variables útiles para supervisar el funcionamiento de las circuladoras de la nave.

En el panel se puede establecer el tiempo de funcionamiento de las circuladoras cuando lleven un tiempo paradas (también modificable). Las dos temperaturas que se podrán modificar en este panel son la temperatura de consigna de la nave (cuando una temperatura, temp1 o temp2, este por debajo de Temp. Consig. la circuladora correspondiente tendrá que ponerse en marcha) y la temperatura a partir de la cual se estima que es suficiente para poder empezar a bombear agua hacia la nave. Adicionalmente, hay dos pilotos que permiten comprobar si las señales de activación de las circuladoras funcionan correctamente. Por último, dos botones para cambiar de panel.



Figura 44: Segundo Panel

6.2.4. Programación tercer panel

El panel 3 se ha utilizado exclusivamente para configurar la compensación en caso de que el exterior la temperatura sea muy baja en comparación con el agua de la caldera.

En esta pantalla se introducirá la temperatura a la que tiene que estar el agua de la caldera (Temp. Consigna Caldera). La variable diferencia de temperatura, tal y como puede verse en el POU donde se realiza la función de compensar la temperatura, sirve para establecer tres rangos de diferencia de temperaturas. Si la diferencia entre la temperatura de consigna del acumulador y la temperatura exterior supera en 2 veces el valor de la variable Diferencia Temp., entonces la temperatura del agua de la caldera se incrementará el doble del valor que haya en la variable Compensacion Grados. Si esta diferencia es menor que 2 veces la diferencia entre Temp. Consigna Caldera y Temp. Exterior la temperatura del agua de la caldera solo se incrementara el valor que tenga la variable Comp. Grados. Por último, si la diferencia de temperaturas es inferior al valor puesto en Diferencia Temp. no se incrementara en ningún valor la temperatura del agua de la caldera. Aunque se cumplan las dos condiciones en este último caso (si la diferencia de temperaturas es menor que la variable Diferencia Temp. también se cumplirá que es menor que $1.5 \cdot \text{Diferencia Temp.}$) la variable Comp. Temp. tomará el valor de la condición ejecutada en último lugar, en este caso cero.

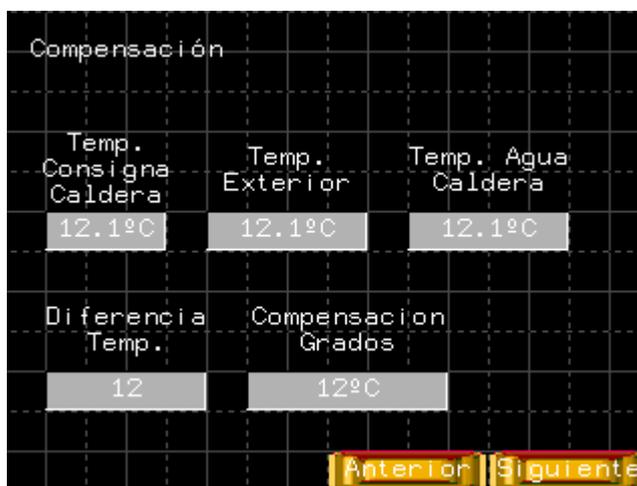


Figura 45: Tercer Panel

6.2.5. Programación cuarto panel

Este panel es necesario para que sea posible la monitorización de todas las variables restantes especificadas en el pliego

En el cuarto panel se podrá modificar el valor del poder calorífico del combustible en caso de cambiar de producto. En este panel también se podrá supervisar el valor que toman el resto de salidas que no aparecen en los otros paneles mediante pilotos (mecheros, entrada de aire y sinfín de la ceniza), el consumo de pellet, la potencia de trabajo de la caldera y el porcentaje que esta potencia representa de la máxima. Con el botón Siguiete de este panel se pasará al panel 5 mientras que con el botón del primer panel, Anterior, se pasara a este panel.



Figura 46: Cuarto Panel

El panel anterior es el último que será accesible, pero temporalmente se configurará el botón siguiente del panel 4 para pasar al panel 5. Al ser necesario configurar los parámetros del PID se realizará un panel que se utilizará mientras se configuren los parámetros del PID para que se puedan modificar los parámetros sin tener que cambiar el programa. Una vez configurados se dará un valor fijo a los parámetros en el programa del autómatas y el botón siguiente del panel 4 solo servirá para volver al panel 1 y así evitar una modificación de estas variables por alguien que no conozca lo que representan ya que podría causar que el sistema no consiguiera estabilizar la temperatura el agua.

6.2.6. Programación quinto panel

Para facilitar la configuración de los parámetros, además de permitir modificar las constantes del PID y el tiempo que tarda en actualizar valores, en el mismo panel también se podrá supervisar la temperatura del agua, el valor de la temperatura consiga y el de la salida analógica.



Figura 47: Panel Configuración PID

6.2.7. Ahorro energía del panel

Por último para acabar con la programación de la pantalla, se configurará para que cuando pase un minuto sin que sea pulsada se apague. Clicando dos sobre HMIS5T en la ventana de navigator aparece en la ventana inferior de Property Inspector la pestaña Backlight. En esta pestaña se puede elegir los minutos que han de pasar sin que sea pulsada antes de que se apague.

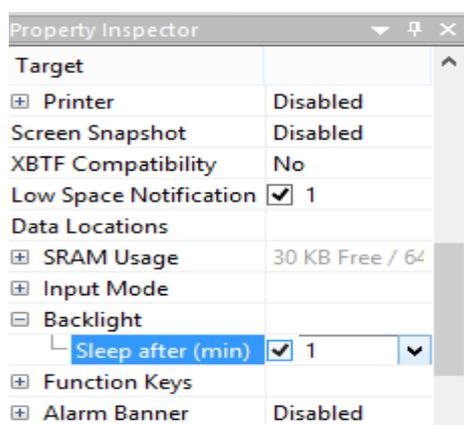


Figura 48: Configuración panel

6.3. Configuración del Variador

El variador ha de configurarse para que modifique su frecuencia de salida en función del valor de una entrada analógica 0-10V, esta entrada es la que se comunicara con la salida analógica del automático.

Para configurar el variador se utilizaran los botones de los que dispone para navegar por el menú de selección y modificar sus valores.

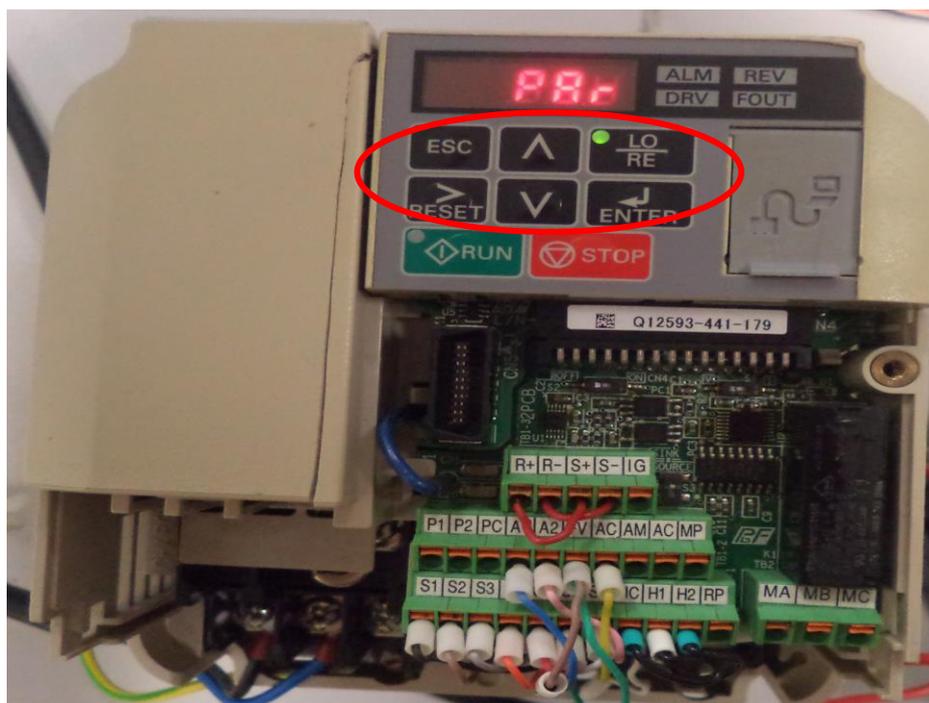


Figura 49: Variador V1000

Una vez encendido hay que buscar dentro del menú principal la opción Par utilizando las flechas de arriba y abajo. Una vez encontrado se pulsa ENTER para acceder dentro de la función y poder así modificar los parámetros necesarios para que el variador funcione como se desee.

Los parámetros a modificar así como el motivo por el cual se modifican son los siguientes:

B1-01: Se modificara su valor a 1 para que la frecuencia de salida sea modificable a través de su entrada 0-10V.

H3-03: Mantendrá su valor por defecto, 100%. Este parámetro permite establecer el porcentaje respecto a la frecuencia de referencia que tomará la frecuencia de salida cuando la entrada tenga valor 10V.

H3-04: Mantendrá su valor por defecto, 0%. Este parámetro modifica en un porcentaje respecto a la frecuencia de referencia la frecuencia de salida cuando la entrada es de 0V.

Los siguientes parámetros han de configurarse para crear la curva tensión-frecuencia que se espera obtener en la salida del variador.

E1-04: valor de la frecuencia máxima, 100Hz.

E1-05: tensión máxima, 220V.

E1-06: valor de la frecuencia base, 30Hz.

E1-07: valor de la frecuencia intermedia, 50Hz.

E1-08: voltaje para frecuencia intermedia, 100V.

E1-09: valor de la frecuencia mínima, 0Hz.

E1-10: voltaje para frecuencia mínima, 0V.

Además de lo anterior se modificara un último parámetro que permitirá que mediante una salida digital del autómatas se ejecute la función marcha/paro del variador. El parámetro a modificar es el H1-01 cambiando su valor a 40 (cuando se active la entrada digital el variador trabajara en sentido de giro directo). De esta forma se podrá controlar el encendido y el apagado del variador con una variable del programa del autómatas.

Otros parámetros como el C1-01 y el C1-02 podrían modificarse (introduciendo un número que representa tiempo en segundos) si se necesita que el tiempo de aceleración y deceleración sean distintos de cero (valor por defecto).

Si el variador ha sido configurado con anterioridad, es posible que no todos los valores de los parámetros coincidan con sus valores por defecto. Modificando el parámetro A1-03 al valor 2220 se inicializa el variador, restableciendo todos los parámetros por defecto.

7. COMUNICACIÓN ENTRE LOS EQUIPOS

En la siguiente figura se presenta de forma muy esquemática como se pretende establecer la conexión entre los distintos dispositivos necesarios para realizar las comprobaciones y pruebas de funcionamiento del programa.

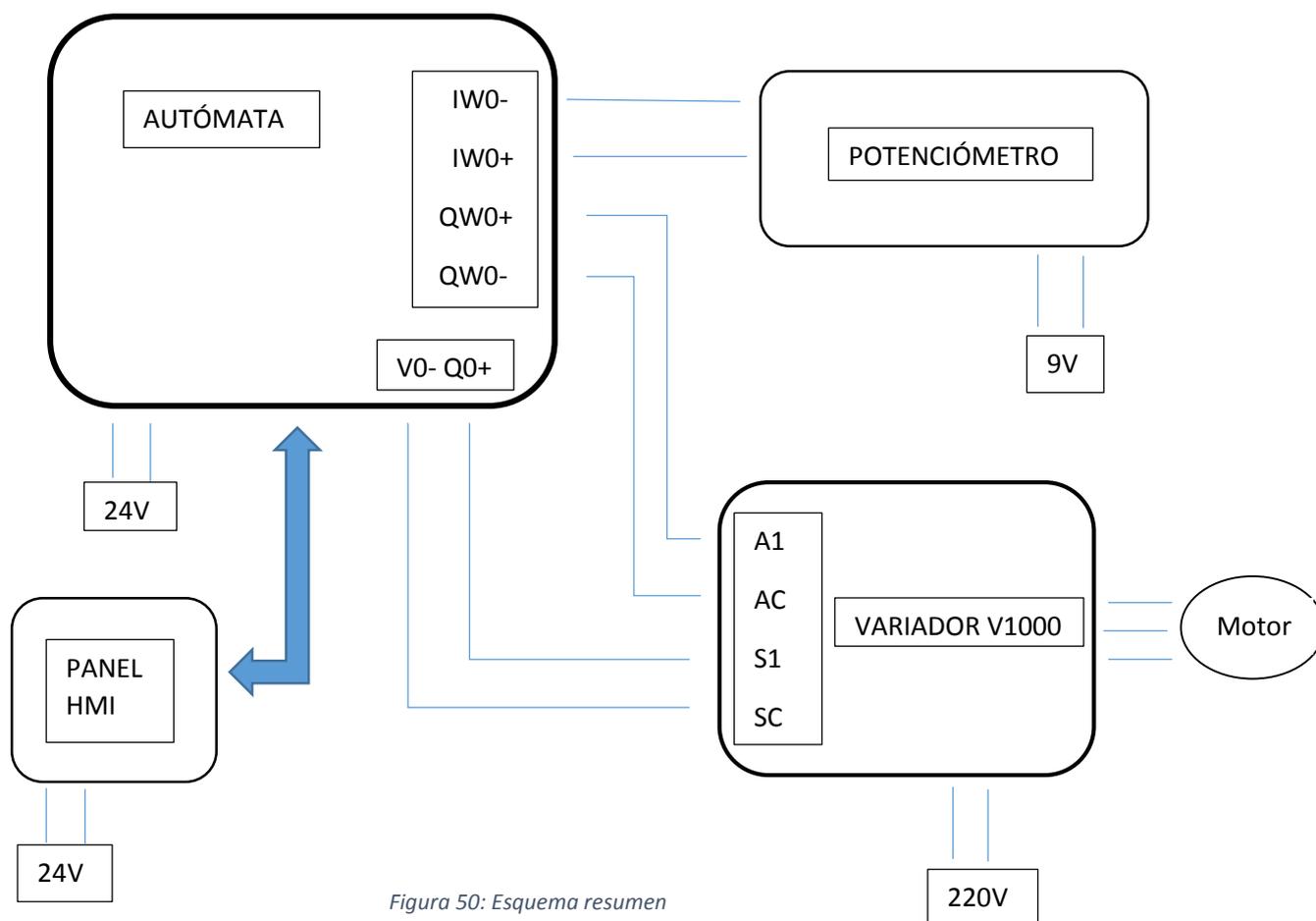


Figura 50: Esquema resumen

Todas estas conexiones se explicarán por separado más adelante dentro de este apartado, pero antes hay que realizar dos puntos muy importantes para que estas conexiones sean entendidas por los distintos dispositivos de forma correcta: configuración E/S y descarga de los programas. La importancia de este apartado radica en que ya no se realiza una programación de un equipo en concreto, sino que es a partir de ahora cuando se empiezan a integrar conjuntamente todas las tecnologías con las que hasta ahora se ha estado trabajado por separado.

7.1. Configuración E/S

A lo largo de la escritura del programa se han declarado muchas temperaturas como enteras a pesar de que toman valores con decimales. Esto es así porque el valor de estas temperaturas se obtendrá mediante una sonda a través de módulos de expansión diseñados para sondas de temperatura y estos módulos requieren de variables tipo INT (enteras) para que funcionen correctamente. En este caso como no se dispone de un sistema real, las temperaturas para comprobar cómo funciona el programa no se modificaran utilizando sondas. Pero una vez realizadas las comprobaciones será necesario saber configurar entradas de temperatura para que el control funcione en el sistema real.

Para configurar entradas de temperatura, primero se introducirá en el programa del So-Machine un módulo de expansión de entradas analógicas expresamente diseñado para sondas de temperaturas como el TM3TI4. La forma de introducirlo es la misma que con la tarjeta de entradas y salidas analógicas introducida al inicio del apartado de programación. Una vez introducida clicando dos veces sobre ella aparece en el centro la ventana de configuración de las entradas.

Variable	Mapping	Channel	Address	Type	Default Value	Unit	Description
Application.TempA...		IW0	%IW9	INT			
iiModule_2_IW1		IW1	%IW10	INT			
iiModule_2_IW2		IW2	%IW11	INT			
iiModule_2_IW3		IW3	%IW12	INT			

Figura 51: Entrada analógica

Clicando dos veces sobre el nombre de una entrada aparece una nueva ventana donde desplegando la pestaña GLV aparecen todas las variables declaradas en el programa y que es posible elegir. En este caso se ha elegido la variable TempACaldera ya que es una temperatura que en el sistema real será medida por sondas.

Tal y como puede observarse en la columna Type la variable aparece del tipo entero. Una vez seleccionada la variable hay que indicar a qué tipo de sonda irá conectada al canal IW0 (entrada 0 del Modulo_2). Pulsando en la pestaña I/O Configuration aparece la ventana mostrada en la figura 51.

Parameter	Type	Value	Default Value	Unit	Description
Inputs					
IW0					
Type	Enumeration of BYTE	PT100	Not used		Range mode
Scope	Enumeration of BYTE	Celsius (0.1 °C)	Not used		Unit
Minimum	INT(-32768...32767)	-2000	-32768		Minimum value
Maximum	INT(-32768...32767)	8500	32767		Maximum value
Input Filter	INT(0..1000)	0	0 * 10 ms		Input filter
Sampling	Enumeration of BYTE	100	100	ms/Channel	Input sampling selection
IW1					
IW2					
IW3					
Diagnostic					
StatusEnabled	Enumeration of BYTE	Yes	Yes		

Figura 52: Configuración de una sonda

En esta ventana para la entrada IW0 se elegirá simplemente en la columna Value el tipo de sonda o entrada que se vaya a conectar en la tarjeta. En este caso se elegirá por ejemplo una PT100. Una vez elegida el tipo de sonda, automáticamente se actualizará la venta. Este procedimiento se repetiría tantas veces como temperaturas físicas haya que monitorizar en el programa.

Pero tal y como se ha mencionado anteriormente, al no disponer del sistema real, las comprobaciones se realizaran modificando las temperaturas desde el panel táctil exceptuado la temperatura del agua de la caldera. Esta última es necesario simularla modificando su valor al mismo tiempo que se modifica la salida analógica, para poder aproximar lo máximo posible al sistema real.

De esta forma, para poder realizar las comprobaciones serán necesarias una entrada analógica, una salida analógica y una salida digital. A la entrada analógica se conectara un potenciómetro que modificara la tensión de 0-10V y simulara una temperatura (0°C a 0V y 50°C a 10V), la salida analógica será de 0-10V e ira conectada al variador y la salida digital también se conectara al variador para arrancarlo y pararlo.

IW1					
Type	Enumeration of BYTE	0 - 10 V	Not used		Range mode
Minimum	INT(-32768...9999)	0	-32768		Minimum value
Maximum	INT(1...32767)	10000	32767		Maximum value
InputFilter	INT(0..1000)	0	0 * 10 ms		Input filter
Sampling	Enumeration of BYTE	1	1	ms/Channel	Input sampling selection
IW2					
IW3					
Outputs					
QW0					
Type	Enumeration of BYTE	0 - 10 V	Not used		Range mode
Minimum	INT(-32768...9999)	0	-32768		Minimum value
Maximum	INT(1...32767)	10000	32767		Maximum value
QW1					

Figura 53: Configuración entrada analógica

Clicando dos veces sobre el módulo TM3AM6, en la pestaña I/O Mapping hay que cambiar el nombre de las entradas por el de las variables del programa tal y como se ha hecho en el ejemplo de las temperaturas. Y luego en I/O Configuration se modifica el apartado Value tanto para la entrada y la salida a 0-10V quedando como en la Figura 53.

Para configurar una salida digital del autómatas hay que pulsar en el icono  y en la pestaña I/O Mapping cambiar el nombre de la salida por el de la variable que se desee enlazar con la salida. Cuando se pasen los equipos al sistema real, las salidas digitales para circuladoras, sinfines, etc. se configurarán de esta misma forma. Para realizar las pruebas se configurará la primera salida digital (la que activa el variador de frecuencia).

Una vez configuradas todas las entradas y salidas que se utilizarán en la prueba de funcionamiento, hay que realizar pequeñas modificaciones al programa aunque después la modificación en TempACaldera habrá de eliminarse cuando el programa tenga que controlar el sistema real. Teniendo en cuenta que tanto la entrada como la salida analógica toman como mínimo valor 0 (a 0V) y como máximo valor 10000 (a 10V) hay que escalar las variables cuyo valor depende de estas entradas y salidas. En el caso de la salida analógica ya apareció en el apartado de programación del autómatas. Si el error total (ErrT) vale un 100% entonces la variable de la salida analógica (SalAnalog) valdrá 10000 y el voltaje en la salida por tanto tomará el valor de 10V.

```
61 SalAnalog:=REAL_TO_INT((10000/100)*ErrT);
```

En el caso de la entrada, cuando la entrada tome el máximo valor (10V) la variable EntAnalog valdrá 100000 y por tanto el agua de la caldera estará supuestamente a 50°C. La variable IncGrados es la variable utilizada para compensar la temperatura.

```
43 TempACaldera:=(EntAnalog*50/10000)-IncGrados;
```

7.2. Transferencia del Programa.

Para transferir el programa al autómatas y al panel hay que conectarlos al ordenador con un cable miniB-USB y en el caso del panel también hay que conectarlo a una fuente de alimentación. Antes de poder descargar el programa en el panel hay que configurar el modo de descarga y la dirección del autómatas con el que tiene que comunicarse.

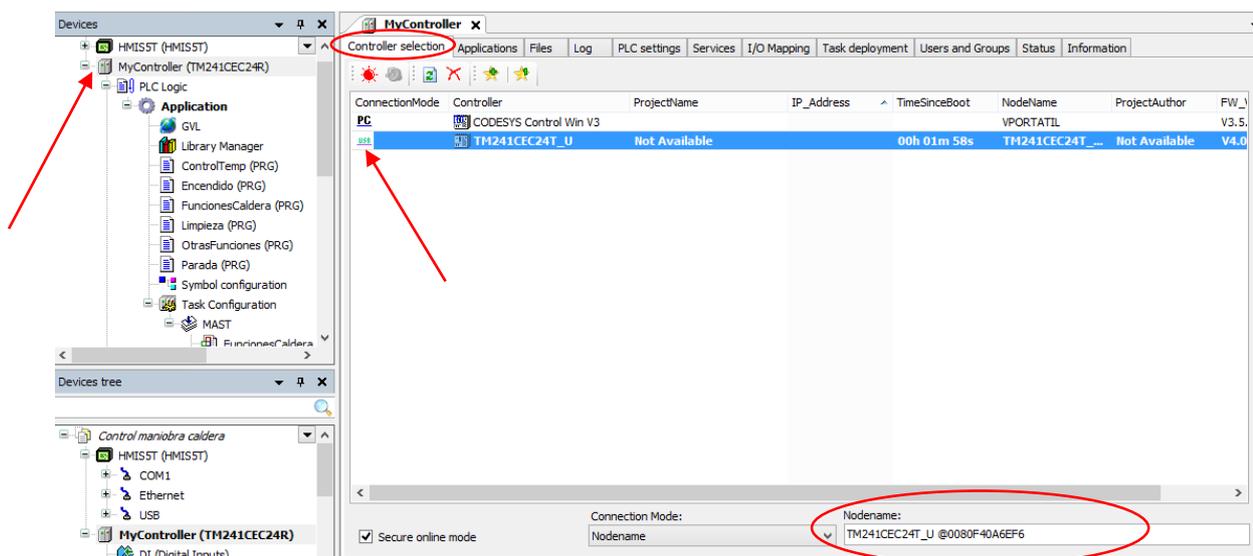


Figura 54: Dirección del autómatas

Para conocer la dirección del autómatas que hay que indicar a la pantalla hay que clicar dos veces sobre MyController en la ventana Devices. Una vez aparece una ventana central parecida a la que se observa en la Figura 54 hay que seleccionar el autómatas con un ConnectorMode USB (solo aparecerá esta opción si el autómatas está conectado por USB al ordenador). En la parte inferior derecha aparece en Nodename la dirección del autómatas. Una vez copiada esta dirección hay que pasar al Vijeo-Designer.

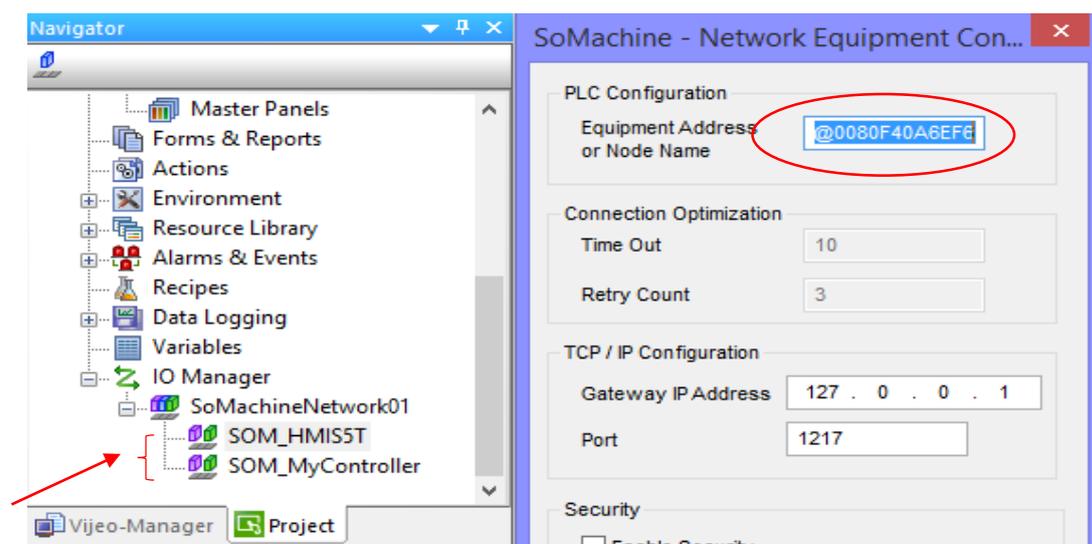


Figura 55: Ventana donde se introduce la dirección al autómatas

Dentro del Vijeo-Designer, en el menú de Navigator hay que buscar IO Manager, desplegándolo aparece otro desplegable con nombre SoMachineNetwork01. Una vez desplegado este último aparecen dos iconos con nombre SOM_HMIS5T y SOM_MyController. Clicando dos veces sobre uno de ellos aparece una ventana como la de la derecha en la Figura 55. En esta ventana, en el apartado de Equipment Address or Node Name hay que copiar la dirección del autómatas y repetir el proceso con el otro icono.

Una vez hecho esto hay que indicar el modo de descarga del programa al panel.

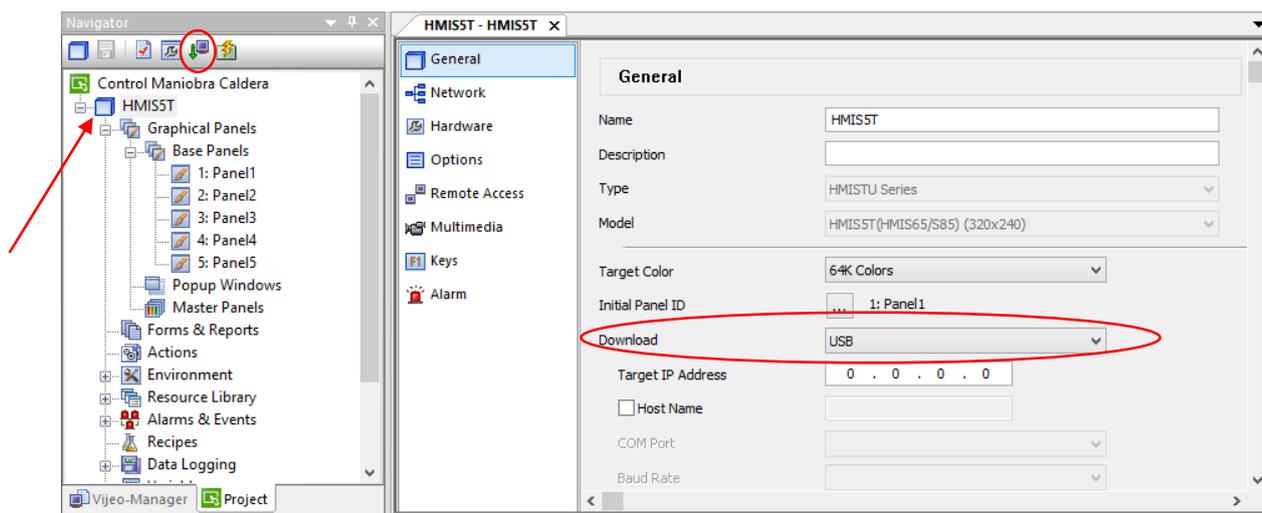


Figura 56: Modo de descarga Vijeo-Designer

Seleccionando HMIS5T en la ventana Navigator aparece en la parte central de la pantalla, en la opción Download, un desplegable con las distintas opciones posibles para descargar el programa al panel. En este caso hay que elegir USB.

Ahora ya es posible transferir el programa al panel pulsando en el icono señalado en la parte de arriba de la ventana Navigator.

En el caso del autómatas, para transferir el programa, solo es necesario conectarse al dispositivo pulsando en el icono  debajo de la barra de herramientas.

Una vez pulsado el icono aparecerán algunos cuadros de dialogo donde se preguntara si se está seguro de querer realizar la transferencia o si se quiere realizar la aplicación de arranque, etc. Esto es muy importante tenerlo en cuenta cuando se va a transferir un programa a un autómatas que ya se encuentra con todas las conexiones realizadas y su arranque podría provocar también el arranque de máquinas u otros elementos peligrosos para posibles personas que se encuentren cerca de los equipos.

Adicionalmente, para poder transferir el programa se requiere que no haya errores de compilación. Puede comprobarse si hay o no errores de compilación pulsando en el icono que se encuentra a la izquierda del icono de transferencia del programa.

7.3. Conexiones

El último paso es conectar los equipos entre ellos para que puedan intercambiar información entre ellos y puedan trabajar conjuntamente. En la figura 57 puede verse como resulta el montaje completo de todos los dispositivos. Para que esta figura sea más clara se explicaran todas las conexiones realizadas.

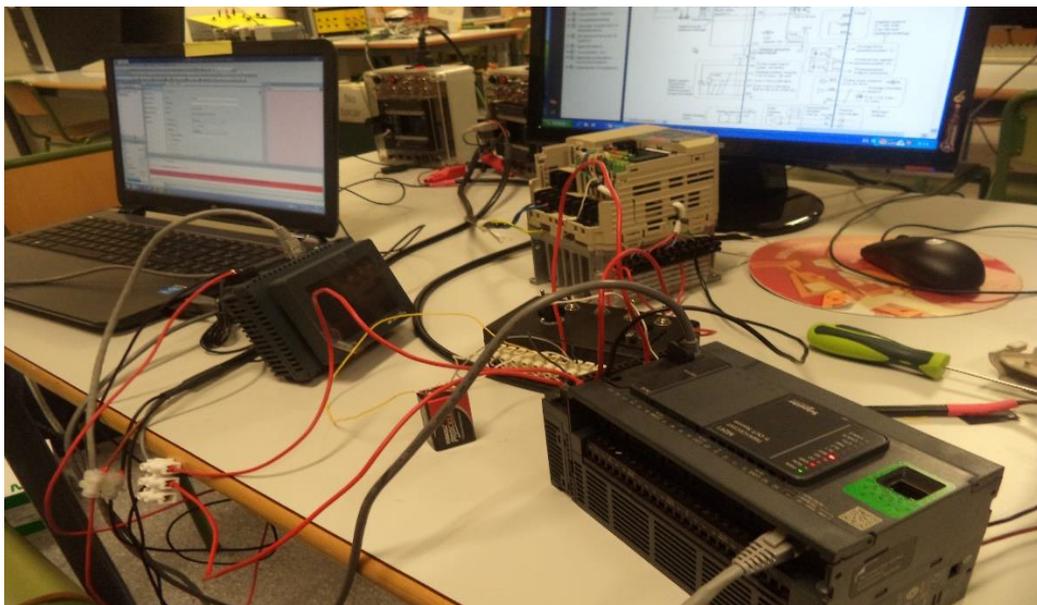


Figura 57: Montaje completo

Teniendo las entradas y salidas configuradas en el automático, habrá que conectar la salida analógica del automático con la entrada analógica del variador, la salida digital del automático con la entrada digital de marcha/paro del variador y el potenciómetro con la entrada analógica del automático. La comunicación entre el automático y el panel se realizara a través de un cable de comunicaciones entre puertos serie. A demás todos los equipos requieren de alimentación:

- El variador se conectará directamente a una tensión de 220V AC.
- Tanto el automático como el panel se conectaran a una fuente de alimentación de 24V DC.
- El módulo de E/S analógicas también requiere de alimentación a 24V DC.

-Por último, el potenciómetro se conectará una pila de 9V DC al no disponerse de fuente de alimentación de 10V.

La conexión entre el automático y el panel no requiere de más explicación ya que solo hay dos conexiones posibles para un cable de comunicaciones en los dispositivos, Ethernet y puerto serie. En este caso se conectará al puerto serie. En las imágenes se han indicado el puerto serie del automático y panel, la conexión a la alimentación de la pantalla, la conexión utilizada para conectar el panel al ordenador y aunque no se utilicen también se han indicado los puertos Ethernet.

La conexión USB entre el panel y el ordenador solo es necesaria cuando hay que volver a transferir el programa porque se ha realizado una modificación o se quiere corregir un error en el programa del Vijeo-Designer. Sin embargo la conexión por USB entre el autómata y el ordenador no se utiliza solamente para transferir el programa, también se utiliza para supervisar el valor que toman las variables del programa a través del ordenador. Por este motivo además de las conexiones que en las siguientes páginas se explican y de la conexión entre el autómata y el panel, también se mantendrá el cable USB-MiniB entre el ordenador y el autómata.

7.3.1. Imágenes panel y autómata



Serie

Ethernet

- 24V +



USB

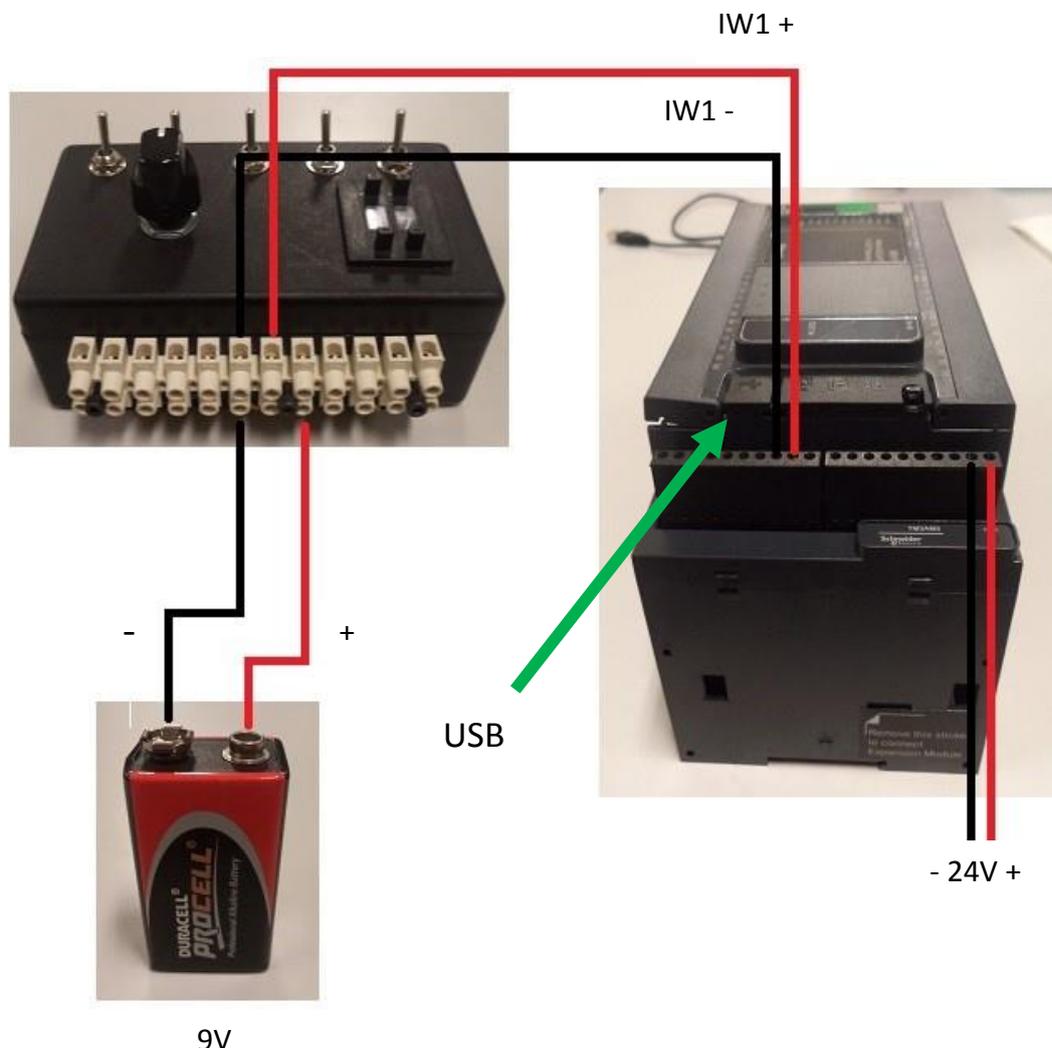


Ethernet

Serie

Las conexiones más complejas se explicarán utilizando fotografías de los equipos y por separado, ya que podría resultar confuso si se mostrara todo el equipo cableado en una misma imagen. Es evidente que las conexiones en el momento de realizar la prueba han de estar todas realizadas a la vez y no por separado.

7.3.2. Conexión del potenciómetro

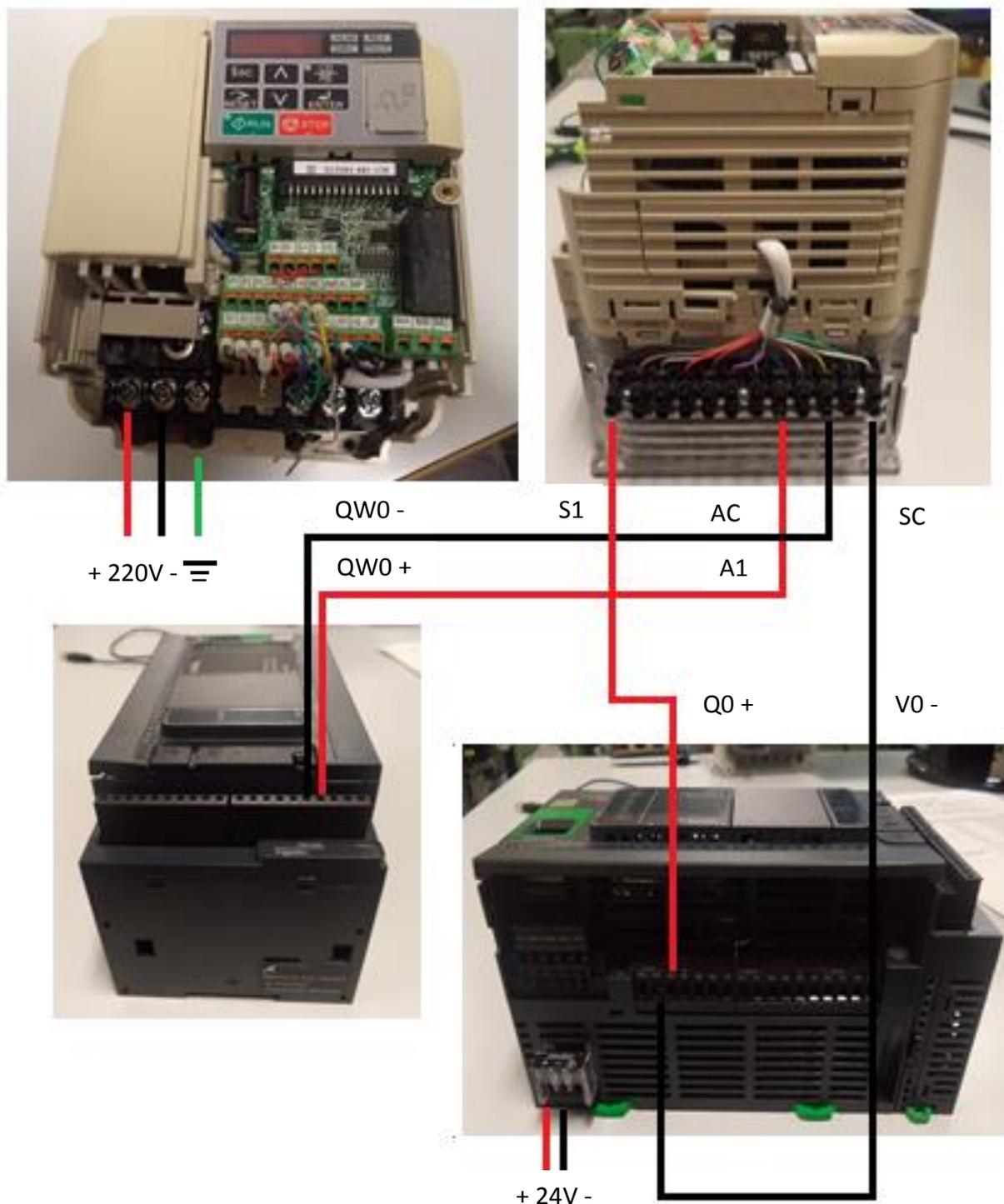


Para el conexionado del potenciómetro, hay que abrirlo y observar cuales son los tres conectores de la resistencia variable. Todos los otros bornes son para los interruptores que pueden verse en la imagen. Al positivo de la entrada analógica (IW1+) hay que llevar el borne de la resistencia variable (que puede variar entre 0 y 9V) mientras que al negativo (IW1-) hay que llevar el mismo negativo que se utiliza para generar la diferencia de potencial en el potenciómetro.

La flecha señala la conexión que permite la comunicación entre el automático y el ordenador. Esta conexión también se mantendrá durante las pruebas para poder supervisar con el ordenador el funcionamiento del programa.

Por último se han indicado los dos bornes utilizados para la alimentación del módulo de E/S.

7.3.3. *Conexión del Variador.*



En la siguiente página aparece un esquema de conexiones del variador. En este caso las conexiones a sus entradas se han llevado a un lateral con una regleta para facilitar la conexión (imagen esquina superior derecha). Si se quiere conectar a S1, por ejemplo, solo hay que mirar el color del cable que sale de S1 y conectar a la regleta que llegue ese mismo color.

Las alimentaciones indicadas en las imágenes son la del Variador (220V AC) y la del (autómata 24 DC)

Diagrama de conexiones del variador V1000.

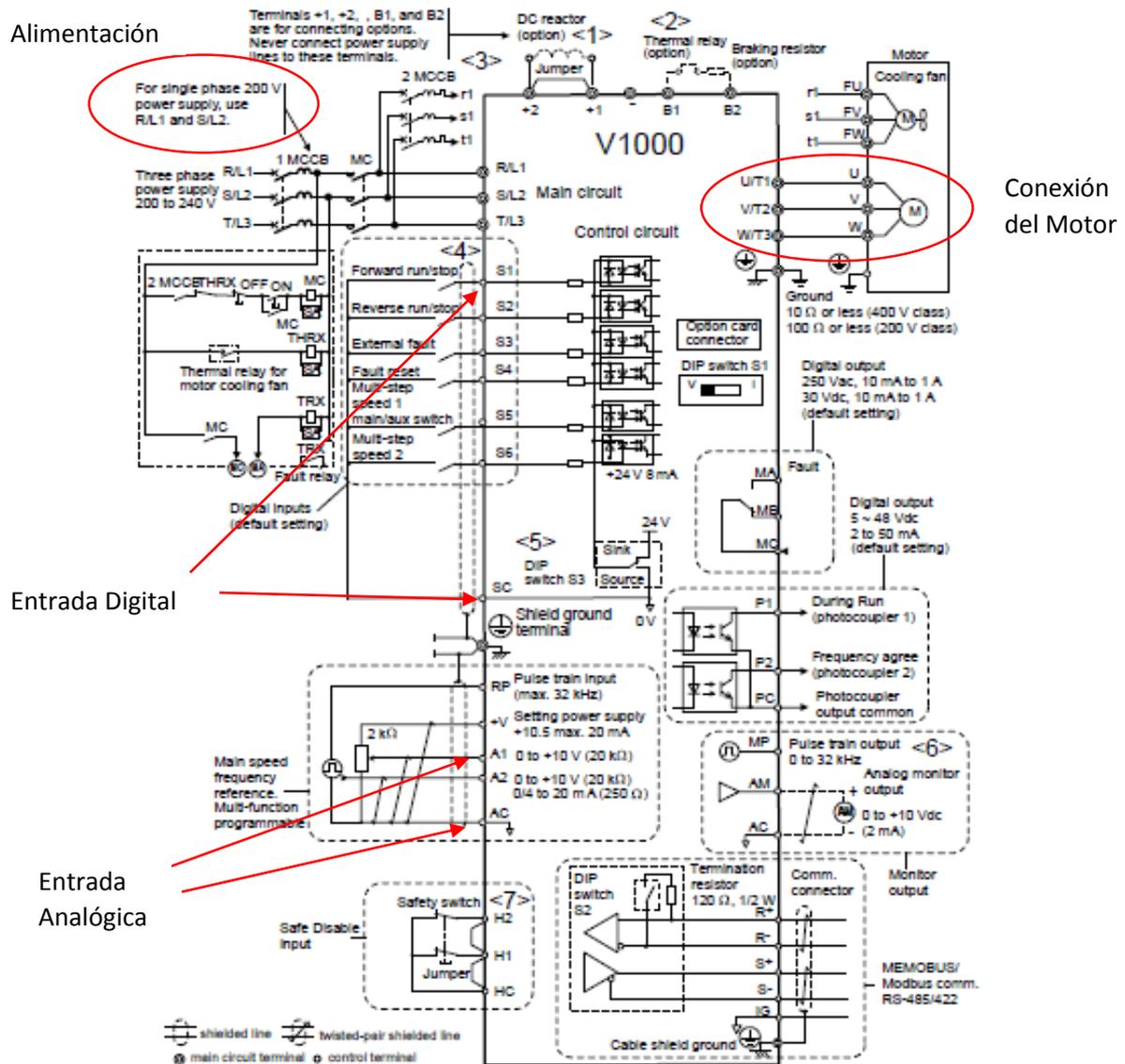


Figura 58: Esquema V1000

Para la entrada analógica hay que conectar el positivo QW0+ en A1 y el negativo QW0- en AC. Para que el variador arranque y pare con una salida digital del automático, al conector S1 hay que conectar el positivo de la salida del automático Q1+ y a SC el negativo de la salida digital V0-

El variador se alimentara con una fase y neutro a 220V, y se conectaran a R/L1 y S/L2.

8. COMPROBACIONES Y CORRECCIONES

Una vez transferidos los programas al autómatas y panel con todas las E/S configuradas y todas las conexiones establecidas ya es posible empezar con la simulación del proceso que se pretenda automatizar. Este punto del proyecto es muy importante ya que permite identificar fallos en el programa, corregirlos y evitar así que estos errores se produzcan cuando están conectados los equipos del sistema real. Si no se comprobara correctamente todas las funciones antes de colocar el control dentro del sistema real no solo se podría dañar al equipo que compone el sistema sino también a personas.

En este caso para encontrar posibles errores se han seguido los pasos que habría que seguir para la puesta en marcha de la calefacción de la nave. Al no haber sondas de temperatura reales, hay que introducir las temperaturas que supuestamente marcarían las sondas. Por ejemplo, para la temperatura del agua de la caldera (TempACaldera) hay que girar el potenciómetro hasta que en el panel aparezcan 20.0°C, mientras que por el panel se introducirán el resto: temperatura de humos (TempHumos) 20.0°C, temperatura en la primera mitad de la nave (Temp1) 22.0°C, en la segunda mitad (Temp2) 21.7°C, temperatura exterior (TempExt) 20.0°C.... También hay que introducir los valores consigna como la temperatura a la que tiene que mantenerse el agua en la caldera (TempCCaldera) 40°C, la temperatura a partir de la cual las circuladoras podrán bombear agua a la nave (TempSuf) 25°C, etc. Así como los tiempos de marcha y paro de las circuladoras (Tfuncion y Tespera), el tiempo del PID y los parámetros del mismo. Una vez ya se ha “creado el entorno” hay que poner en marcha el autómatas para que empiece a ejecutar el programa pulsando el botón  (run) en la ventana principal del So-Machine.

Clicando en cualquier POU una vez el autómatas está en marcha puede verse el valor que toma cada una de las variables a cada instante. Clicando sobre el POU que contiene la función de arranque (la primera que debería iniciarse si se pretende simular un funcionamiento real) ya puede observarse el primer error. La variable TempHumos no tiene el valor que se esperaba, su valor es de 200 y no de 20 como se había indicado por panel. El motivo es el mismo por el que las constantes del PID se habían de dividir por 100 explicado en el apartado de programación. Aunque en la pantalla aparezcan con un decimal estas variables son tipo enteras, por lo tanto todas las temperaturas que se hayan introducido con un decimal habrá que dividirlos por 10. Cuando se utilicen sondas reales también ocurrirá lo mismo porque tal y como puede verse en la Figura 52 (página 49), la sonda PT100 realiza un scope de 0,1°C y por lo tanto el valor de las temperaturas de las sondas también tendrán un decimal.

Para cambiar el programa hay que desconectarse del autómatas con el mismo icono que se utiliza para conectarse cuando no se está conectado. Una vez modificados todos los POU en los que aparecen temperaturas introducidas por sonda (hay que dividir todas estas variables por 10) y vuelto a conectarse al autómatas para actualizar el programa, hay que pulsar el botón del panel encender y automáticamente se enciende el piloto F. encender indicando que se ha activado la función de encender.

Si ahora se observa el POU de esta misma función puede observarse como muchas de las variables que antes se encontraban en estado FALSE ahora han pasado a TRUE. En las Figuras 59 y 60 se intenta mostrar como tal y como se ha explicado en el apartado de programación de este POU, cuando pasen dos minutos habiendo entrado alimentación a la caldera se detenga el sistema de alimentación.

```

MyController.Application.Encendido
1 IF Fencender TRUE THEN
2
3 IF TempHumos 200 /10 < 50 THEN
4
5     Entrada TRUE :=TRUE;
6     TONAlim (IN TRUE := Entrada TRUE, PT T#2m := T#120S);
7     Aire TRUE :=TRUE;
8     Mecheros TRUE :=TRUE;
9     SalAnalog 10000 :=10000;
10
11     IF TONAlim.Q FALSE THEN
12         SalAnalog 10000 :=0;
13     END_IF
14 END_IF
15
16 IF TempHumos 200 /10 >= 50 THEN
17     Entrada TRUE :=FALSE;
18     Mecheros TRUE :=FALSE;
19     Encender FALSE :=FALSE;
20     Calefaccion FALSE :=TRUE;
21 END_IF
22 END_IF RETURN
    
```

Figura 59: Antes de pasar 2 minutos

```

MyController.Application.Encendido
1 IF Fencender TRUE THEN
2
3 IF TempHumos 200 /10 < 50 THEN
4
5     Entrada TRUE :=TRUE;
6     TONAlim (IN TRUE := Entrada TRUE, PT T#2m := T#120S);
7     Aire TRUE :=TRUE;
8     Mecheros TRUE :=TRUE;
9     SalAnalog 0 :=10000;
10
11     IF TONAlim.Q TRUE THEN
12         SalAnalog 0 :=0;
13     END_IF
14 END_IF
15
16 IF TempHumos 200 /10 >= 50 THEN
17     Entrada TRUE :=FALSE;
18     Mecheros TRUE :=FALSE;
19     Encender FALSE :=FALSE;
20     Calefaccion FALSE :=TRUE;
21 END_IF
22 END_IF RETURN
    
```

Figura 60: Después de pasar 2 minutos

En la primera Figura de esta página la variable SalAnalog toma el valor de 10000, mientras que en la segunda (una vez transcurridos los dos minutos y se activado la salida del temporizador) la variable vale 0.

Esta función ya se ha comprobado que funciona correctamente y por tanto ya se puede pasar a comprobar la siguiente función. Siguiendo con la simulación, cuando la temperatura de humos supere los 50°C se activara la siguiente función y la más importante de este trabajo, el control de la temperatura del agua del sistema de calefacción.

Modificando por panel la temperatura de humos a 60.0°C (por ejemplo) puede verse como se apaga el piloto de la función encender y se enciende el de la función de calefacción. Esta función consta de dos partes, el control de las circuladoras y el de control de la temperatura de la caldera. Como inicialmente se había puesto el potenciómetro en una posición en la que el valor de la variable del agua de la caldera fuera de 20.0°C no se alcanza la temperatura suficiente y el control de las circuladoras no se activara. Por este motivo primero se comprobaba el funcionamiento del control de temperatura.

Todas constantes del PID se les ha introducido un valor de 1.00 y al temporizador del PID 1s. Nada más activarse la función de calefacción, el variador debería estar aumentando su frecuencia de salida ya que la temperatura del agua no llega al valor de la consigna (40°C). Sin embargo la frecuencia que marca el variador es 0Hz.

Lo que ocurre es que al primer segundo el valor del error integral es de 20% y el del error total 40%, pero al cuarto segundo, el error integral valdrá 100% y el error total 120%. En este momento la variable SalAnalog toma un valor de 12000 cuando la salida analógica solo puede recibir valores entre 0 y 10000. Al superar el valor permitido, la diferencia de potencial en la salida analógica y por lo tanto en la entrada del variador de frecuencia valdrá 0V. Para que esto no suceda hay que limitar el valor de la variable SalAnalog añadiendo las siguientes líneas al programa:

```

63 IF ErrT < 100 THEN
64   SalAnalog:=REAL_TO_INT((10000/100)*ErrT);
65 END_IF
66
67 IF ErrT>=100 THEN
68   SalAnalog:=10000;
69 END_IF

```

Figura 61: Saturación variable analógica

Mientras el error total se encuentre por debajo del 100% la variable SalAnalog tomará el valor que le corresponda en función de la variable ErrT, pero si se supera el error del 100%, entonces la variable será igual a 10000. Para poder observar mejor la variación de frecuencia en el variador la constante integral (Ki) se ha reducido su valor a 0.25 ya que el incremento del error es muy rápido.

Una vez hechas estas modificaciones, modificando el valor de la temperatura mediante el potenciómetro ya puede verse un funcionamiento correcto. Si el valor de la temperatura del potenciómetro supera los 40°C entonces el variador reduce su frecuencia de salida con cada segundo, cuando se igualan las dos temperaturas la frecuencia de salida del variador se mantiene constante y si la temperatura está por debajo de la de consigna la frecuencia aumenta también a cada segundo.

Una vez solucionado el problema de esta parte de función, se utiliza el potenciómetro para dejar la temperatura del agua de la caldera a la temperatura de consigna y se comprueba el control de las bombas de agua.

A pesar de que la temperatura del agua de la caldera está por encima de la temperatura suficiente para que las bombas puedan bombear y que las temperaturas temp1 y temp2 se encuentran por debajo de la temperatura de consigna de la nave las variables Circ1 y Circ2 se mantienen en FALSE.

El otro modo de funcionamiento de las bombas es cuando ya se hayan alcanzado las temperaturas de consigna en ambas partes de la nave y transcurra el tiempo introducido en T. Espera las circuladoras se encenderán durante un tiempo T. Funcion. En este caso una vez puestas las temperaturas Temp1 y Temp2 a 30°C (valor de la variable temperatura consigna de la nave) las los pilotos que indican la activación y desactivación de las variables sí que se encienden y apagan en los tiempos cuando transcurren los tiempos correspondientes.

Lo que causaba el error anterior era la propia función BLINK. Aunque su variable ENABLE no este activa, la función sí modifica su variable (contrariamente a lo que se creía cuando se explicó su funcionamiento en el apartado de programación) de salida al valor correspondiente. De esta forma aunque en líneas anteriores se activen las variables Circ1 y Circ2, al pasar el programa por la función BLINK, las sobrescribe a FALSE. Este problema queda solucionado escribiendo una línea que condicione la entrada en la función BLINK a que esté activa su variable ENABLE.

Siguiendo este sistema se ha comprobado el resto de funciones:

Cuando se pulsa el botón del primer panel de apagar, el piloto luminoso de entrada de aire se apaga y todas las variables toman el valor FALSE.

Aunque se pulse el botón de limpieza, esta función no se activa hasta que la temperatura de humos baja por debajo de 50°C.

El sistema de compensación de temperatura se activa cuando se introducen valores a las variables del panel 3, reduciendo el valor de la temperatura del agua de la caldera en el valor introducido en la variable GradosComp.

Y los valores del 4º panel de potencia, consumo y nivel de trabajo se ha comprobado que sean correctos en función del valor de la variable del poder calorífico y de la salida analógica que controla la frecuencia del variador.

9. CONCLUSIONES

Una vez acabadas las comprobaciones y corregidos los errores podría trasladarse ya el equipo al sistema real para empezar las verdaderas pruebas con todos los dispositivos que corresponda: Conectar los módulos de expansión necesarios para todas las sondas existentes en la instalación, realizar las conexiones entre el autómatas, los relés y las maquinas que deban ejecutar las variables del autómatas y conectar el motor del sistema de alimentación a la salida del variador de frecuencia.

Habría que eliminar la línea de programa que modifica la variable de la temperatura del agua de la caldera en función de la entrada analógica y configurarla como una entrada analógica de sonda de temperatura. De igual forma habría que configurar todas las variables de temperatura que hasta ahora se introducía su valor por pantalla para que tomen el valor correcto en función de la temperatura de las sondas tal y como se ha explicado en el apartado Configuración E/S.

Una vez todo modificado y adaptado conforme a las necesidades del sistema real solo faltaría encontrar valores para las constantes del PID que permitieran estabilizar la temperatura del agua de la caldera aunque se produjeran perturbaciones al arrancar las circuladoras ya que introducen agua fría en el circuito mientras se envía agua caliente a la granja.

Esto es lo que en esencia se ha conseguido en este trabajo: Un sistema capaz de supervisarse a sí mismo y capaz de activar actuadores que modifiquen el propio sistema para corregir errores sin necesidad de que una persona esté constantemente atenta a lo que ocurra en él.

A lo largo de todo el proyecto se ha mostrado como programar un autómatas, como configurar sus entradas y salidas digitales y analógicas, como programar una pantalla táctil que se comunique con el autómatas y como configurar un variador para que obligue a trabajar un motor en función de lo que pida el programa. Todo ello para poder integrarlos y poder ejecutar de forma automática y sin supervisión humana una función tan específica como el control de temperatura en una granja.

Pero de igual manera que se ha preparado este equipo para el control de esta función específica, aplicando los mismos conceptos que se han presentado en este trabajo, se podría haber adaptado el equipo para que realizara el control de cualquier otra función similar.

Por este motivo no solo es objetivo de este trabajo hacer entrega de la maniobra que se exigía automatizar sino también como ha sido posible alcanzar esta automatización y permitir el uso de la información presente en este trabajo para la realización de otras maniobras.

A pesar de todo, hay que tener en cuenta que por muy claros que se tengan todos los conceptos, en los sistemas reales a automatizar siempre pueden darse casos que no se habían previsto o no se creía que pudieran darse. Por eso es necesario saber no solo configurar y programar, sino también saber analizar y comprender estos sistemas reales para poder prever cualquier casuística y evitar posibles daños a componentes del sistema o personas en el peor de los casos.

En definitiva la automática es una rama de la robótica necesaria hoy en día y con una infinidad de posibilidades cuando conocimientos de ciencias, como la ingeniería, se le suman en un mismo objetivo.

10. BIBLIOGRAFÍA

La información de los distintos equipos se ha obtenido de las páginas web oficiales de sus respectivas marcas:

Autómata

Hojas de características: <http://www.schneider-electric.com/es/es/index.jsp>

<http://www.schneiderelectric.es/sites/spain/es/general/busqueda/resultados-de-busqueda.page>

Panel HMI

Hojas de características: <http://www.schneider-electric.com/es/es/index.jsp>

<http://www.schneiderelectric.es/sites/spain/es/general/busqueda/resultados-de-busqueda.page>

Variador

Hojas de características: <http://industrial.omron.es/es/home>

<http://industrial.omron.es/es/search?q=V1000>

Manual de configuración: <http://industrial.omron.es/es/home>

<http://industrial.omron.es/es/search?q=V1000>



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

VIVES JOVANÍ, VICENTE

**PROYECTO DE CONTROL Y
MONITORIZACIÓN DEL SISTEMA DE CALEFACCIÓN
DE UNA GRANJA AVÍCOLA MEDIANTE CALDERA,
AUTÓMATA PROGRAMABLE Y
PANEL DE USUARIO**

PRESUPUESTO

Curso Académico: 2014-15

PRESUPUESTO

Para la elaboración del presupuesto solo se va a tener en cuenta la mano de obra, ya que los equipos y elementos utilizados en la simulación del sistema real forman parte del equipo del laboratorio mientras que los elementos principales del control los proporciona el usuario final.

El montaje final en la instalación real debería ser presupuestado a parte ya que las conexiones entre los equipos las realizaría un técnico.

De este modo solo hay que presupuestar horas de estudio del sistema real, de programación, de comprobaciones y correcciones y de montaje.

Concepto	Cantidad (h)	Precio (€/h)	Euros (€)
Estudio del sistema	33	15	495
Visitas a la instalación	12		
Reuniones con el propietario	21		
Programación	81	25	2025
Programación Autómata	45		
Programación Panel HMI	28		
Configuración Variador	8		
Montajes	40	18	720
Configuraciones E/S	22		
Conexionados	18		
Pruebas y Correcciones	33	22	726
TOTAL			3996€
I.V.A. (21%)			832,86€
Presupuesto Base de Licitación			4828,86€

El presupuesto asciende a la cantidad de:

CUATRO MIL OCHOCIENTOS VEINTIOCHO EUROS CON OCHENTASEIS CÉNTIMOS.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

VIVES JOVANÍ, VICENTE

**PROYECTO DE CONTROL Y
MONITORIZACIÓN DEL SISTEMA DE CALEFACCIÓN
DE UNA GRANJA AVÍCOLA MEDIANTE CALDERA,
AUTÓMATA PROGRAMABLE Y
PANEL DE USUARIO**

ANEJOS

Curso Académico: 2014-15

ANEJOS

En este documento aparecen los elementos que se habían especificado en el pliego de condiciones que habría que entregar al finalizar el trabajo: Lista de instrucciones del programa, lista de variables empleadas y paneles sugeridos para realizar el control.

Listado de Instrucciones

```

IF Encender THEN
    Fencender:=TRUE;
    Fcalefaccion:=FALSE;
    Fapagar:=FALSE;
    Flimpiar:=FALSE;
END_IF

IF Calefaccion THEN
    Fencender:=FALSE;
    Fcalefaccion:=TRUE;
    Fapagar:=FALSE;
    Flimpiar:=FALSE;
END_IF

IF Apagar THEN
    Fencender:=FALSE;
    Fcalefaccion:=FALSE;
    Fapagar:=TRUE;
    Flimpiar:=FALSE;
END_IF

IF Limpiar THEN
    Fencender:=FALSE;
    Fcalefaccion:=TRUE;
    Fapagar:=FALSE;
    Flimpiar:=FALSE;
END_IF

```

1

```

IF Fencender THEN

    EncV1000:=TRUE;

    IF TempHumos/10 < 60 THEN

        Entrada:=TRUE;
        TONAlim (IN:= Entrada, PT:= T#120S);
        Aire:=TRUE;
        Mecheros:=TRUE;
        SalAnalog:=10000;

        IF TONAlim.Q THEN
            SalAnalog:=0;
        END_IF
    END_IF

    IF TempHumos/10 >= 60 THEN
        Entrada:=FALSE;
        Mecheros:=FALSE;
        Encender:=FALSE;
        Calefaccion:=TRUE;
    END_IF
END_IF

```

2

```
IF Fcalefaccion THEN
```

```
    EncV1000:=TRUE;
```

```
    IF TempACaldera > TempSuf/10 THEN
```

```
        IF Temp1/10 < TempConsig/10 THEN
```

```
            Bandera1:=TRUE;
```

```
        END_IF
```

```
        IF (Temp1/10)-2 > TempConsig/10 THEN
```

```
            Bandera1:=FALSE;
```

```
        END_IF
```

```
        IF Bandera1 THEN
```

```
            Circ1:=TRUE;
```

```
        ELSE
```

```
            Circ1:=FALSE;
```

```
        END_IF
```

```
        IF NOT Bandera1 THEN
```

```
            Blink1(enable:= NOT Bandera1, timelow:=Tespera, timehigh:=Tfuncion, out=>Circ1);
```

```
        END_IF
```

```
    IF Temp2/10 < TempConsig/10 THEN
```

```
        Bandera2:=TRUE;
```

```
    END_IF
```

```
    IF (Temp2/10)-2 > TempConsig/10 THEN
```

```
        Bandera2:=FALSE;
```

```
    END_IF
```

```
    IF Bandera2 THEN
```

```
        Circ2:=TRUE;
```

```
    ELSE
```

```
        Circ2:=FALSE;
```

```
    END_IF
```

```
    IF NOT Bandera2 THEN
```

```
        Blink2(enable:= NOT Bandera2, timelow:=Tespera, timehigh:=Tfuncion, out=>Circ2);
```

```
    END_IF
```

```
END_IF
```

```
Tespera:=DINT_TO_TIME(iTespera*1000);
```

```
Tfuncion:=DINT_TO_TIME(iTfuncion*1000);
```

```
TempACaldera:=TempACaldera-IncGrados;
```

```
TPID:=DINT_TO_TIME(iTPID*1000);
```

```
ValTempMax:=100;
```

```
Consig:=TempCCaldera*100/ValTempMax;
```

```
ValActual:=TempACaldera*100/ValTempMax;
```

```
TempACaldera:=(EntAnalog*50/10000)-IncGrados;
```

```
TONPID (IN:= NOT TONPID.Q, PT:= TPID);
```

```
IF TONPID.Q THEN
```

```
Err:= Consig-ValActual;
```

```
ErrI := ErrI+Err;
```

```
ErrD := Err-ErrAnt;
```

```
ErrAnt:= Err;
```

```
ErrT:=Err*(Kp/100)+ErrI*(Ki/100)+ErrD*(Kd/100);
```

```
END_IF
```

```
SalAnalog:=REAL_TO_INT((10000/100)*ErrT);
```

```
IF ErrT < 100 THEN
```

```
SalAnalog:=REAL_TO_INT((10000/100)*ErrT);
```

```
END_IF
```

```
IF ErrT>=100 THEN
```

```
SalAnalog:=10000;
```

```
END_IF
```

```
END_IF
```

Esta instrucción durante las pruebas era la que se indica debajo del cuadro de texto

```
IF Fapagar THEN
    SalAnalog:=0;
    EncV1000:=FALSE;
    Aire:=FALSE;
    Mecheros:=FALSE;
    Circ1:=FALSE;
    Circ2:=FALSE;
END_IF
```

4

```
IF Flimpiar THEN
    SalAnalog:=0;
    EncV1000:=FALSE;
    IF TempHumos < 50 THEN
        SinfinCeniza:=TRUE;
    ELSE
        SinfinCeniza:=FALSE;
    END_IF
END_IF

IF NOT Flimpiar THEN
    SinfinCeniza:=FALSE;
END_IF
```

5

```
//Compensación de temperatura

IF TempCCaldera-TempExt > 2*DifComp THEN
    IncGrados:=2*GradosComp;
END_IF

IF TempCCaldera-TempExt < 1.5*DifComp THEN
    IncGrados:=GradosComp;
END_IF

IF TempCCaldera-TempExt < DifComp THEN
    IncGrados:=0;
END_IF

//Variables a visualizar en el panel

Consumo:=(SalAnalog*760*60/1000)/5000;

PotCald:=Consumo*EngPellet*3600/4.18;

NivelTrabajo:= PotCald/600000*100;
```

Lista de Variables

VAR_GLOBAL	Temp1: INT;	Ki: INT;
Encender: BOOL;	TempConsig: INT;	Kd: INT;
Fencender: BOOL;	Bandera1: BOOL;	TempExt: INT;
Fcalefaccion: BOOL;	Blink1: BLINK;	DifComp: INT;
Fapagar: BOOL;	Tespera: TIME;	GradosComp: INT;
Flimpiar: BOOL;	Tfuncion: TIME;	IncGrados: INT;
Calefaccion: BOOL;	iTespera: DINT;	Consumo: REAL;
Apagar: BOOL;	iTfuncion: DINT;	PotCald: REAL;
Limpiar: BOOL;	iTPID: DINT;	NivelTrabajo: REAL;
TempHumos: INT;	TPID: TIME;	EntAnalog: INT;
Entrada: BOOL;	Consig: INT;	Temp2: INT;
TONAlim: TON;	ValActual: INT;	Bandera2: BOOL;
Aire: BOOL;	Err: INT;	Blink2: BLINK;
Mecheros: BOOL;	ValTempMax: INT;	ErrT: REAL;
SalAnalog: INT;	TempCCaldera: INT;	EncV1000: BOOL;
Circ1: BOOL;	TONPID: TON;	EngPellet: REAL;
Circ2: BOOL;	ErrI: INT;	END_VAR
SinfinCeniza: BOOL;	ErrD: INT;	
TempACaldera: INT;	ErrAnt: INT;	
TempSuf: INT;	Kp: INT;	

Variable utilizada únicamente durante las pruebas

Paneles para el Control y Monitorización



