



ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERIA TECNICA INDUSTRIAL (ELECTRICIDAD)

Título del proyecto:

**Control mediante LABVIEW de un horno de
laboratorio**

Autor: Enrique Ruiz Blanco

Tutor: Ignacio Del Villar Fernández

INDICE

1.-MEMORIA	3
1.1 Objeto y descripción del Proyecto	3
1.2 Antecedentes y datos de partida	3
1.3 Descripción de los elementos (Horno LinnVMK39 y GEFTRAN 800P)	4
1.4 Características más importantes del proyecto	4
1.5 Posibles soluciones	5
1.6 Solución adoptada	5
1.7 Descripción de lo proyectado	5
1.7.1.- HARDWARE DEL PROYECTO	7
1.7.1.1.-Introducción a la comunicación serie	7
1.7.1.2.-Interfaz serie	10
1.7.1.2.1.-RS232	10
1.7.1.2.2.-RS485: Estándar de línea balanceada	13
1.7.1.2.3.-Conversión RS232 a RS485:	15
1.7.1.2.3.1.-Fundamentos conversión	15
1.7.1.2.3.2.-Pruebas tarjetas conversoras:	21
A.-Hardware utilizado	21
B.-Ensayos software	23
1.7.2.-SOFTWARE DEL PROYECTO	27
1.7.2.1.-Protocolo bus de comunicación	27
1.7.2.2.-Protocolo Modbus	28
1.7.2.3.-Modos de transmisión Modbus serie	29
1.7.2.4.-Modbus ASCII	30
1.7.2.5.- Modbus RTU	32
1.7.2.6.- Tipos de función	35
1.7.2.7.-Implementación de las funciones en Labview	38
1.7.3.-PROGRAMACIÓN DEL SOFTWARE	39
1.7.3.1.-Introducción a LABVIEW	39
1.7.3.2.-Realización del programa	40
1.7.3.2.1.-Introducción de datos	43
1.7.3.2.1.1.- FRAME 0 CARGAR RAMPA	43
1.7.3.2.1.2.- FRAME 1 CREAR RAMPA	46
1.7.3.2.2.-Ejecución del proceso	47
1.7.3.2.3.-Procesado- envío de datos al controlador	48
1.7.3.2.4.-Visualización de la ejecución	53
1.7.3.3.-Guardado y exportación de datos	59
1.8.-Conclusiones	61
1.9.-Lineas futuras del proyecto.	62
2.-PRESUPUESTO	63
3.-ANEXOS:	
3.1.-MANUAL DE USUARIO	68
3.2.-DATASHEET MAX232	75
3.3.-DATASHEET MAX485	80
3.4.-DATASHEET CONVERTOR RS232/RS485 B&B Electronics	100
3.5.-DATASHEET GEFTRAN 800P	103
3.6.-DIRECCIONES MODBUS EN EL CONTROLADOR	117
4.-BIBLIOGRAFIA	124

1.-MEMORIA

1.1.-Objeto y descripción del proyecto:

El objeto de este proyecto de fin de carrera es realizar un programa que permita el control de forma automática desde un PC del horno Linn VMK39 que hay en el laboratorio de sensores de la Universidad Pública de Navarra, el cual tiene integrado un controlador GEFRAN 800P.

El control del horno se realiza a través del software Labview instalado en el PC, el cual comunica a través de una interfaz serie RS-232 que debe ser convertida a RS-485 para que el controlador admita la comunicación. El protocolo Modbus serie RTU será el utilizado para enviar los comandos al dispositivo.

1.2.-Antecedentes y datos de partida:

El proyecto se va a desarrollar en el Laboratorio de Sensores. El laboratorio consta de un PC y el horno con su controlador. La comunicación entre el ordenador y el controlador del horno se realiza a través del propio software Labview del ordenador y la interfaz serie. Este software ha sido utilizado previamente en la automatización de otros procesos como el control de un panel de gases dentro del Laboratorio de Sensores.

La universidad proporciona el horno Linn VMK39 (figura 1) con el regulador Gefran 800P (figura 2), un ordenador con la correspondiente licencia para el software Labview del cual se han utilizado dos versiones (8.5 y 10) y el cable serie RS232.



Figura 1. Horno Linn VMK39



Figura 2. Controlador GEFRAN 800P

1.3.-Descripción de los elementos (Horno LinnVMK39 y GEFTRAN 800P):

El Horno Linn VMK39 se trata de un horno de laboratorio con la capacidad de alcanzar una temperatura de 1200°C, tiene una carcasa de acero inoxidable lo que permite que trabaje en ambientes corrosivos. Además alberga un control por PID para regular las rampas de calentamiento y enfriamiento. Tiene una capacidad de 3.9 litros y demanda 1.3KW de potencia.

Por otra parte, el Gefran 800P se trata de un controlador de consigna que permite el control de la temperatura del horno a través de los sensores y las múltiples posibilidades de entrada (termopar, termorresistencia, termistor, entradas lineales, potenciómetro, etc). Además ofrece la posibilidad de realizar un control del controlador desde un dispositivo maestro (PC), a través de los protocolos de comunicación CENCAL y MODBUS que son las opciones que permite.

1.4.-Características más importantes del proyecto:

En el laboratorio de sensores del Departamento de Ingeniería Eléctrica y Electrónica de la Universidad Pública de Navarra se tiene un horno Linn VMK39 con un controlador GEFTRAN 800P integrado. El objetivo es conseguir manejar desde el PC con el software LABVIEW y una interfaz serie dicho controlador para dirigir el horno, para ello es necesario crear un programa en labview que comunique con el regulador. Los objetivos de este programa son:

- Poder introducir una rampa de temperatura en función del tiempo de dos maneras: mediante un archivo de texto con los datos del tiempo y la temperatura o directamente introduciendo los datos manualmente en el programa.
- Realizar la comunicación entre el PC y el regulador del horno permitiendo el envío de los datos y la ejecución de los mismos en el horno.
- Visualizar el proceso introducido en el horno a través de una gráfica en labview, sincronizada con el proceso que se estará ejecutando en el horno.
- Poder exportar los datos de la rampa introducidos en caso de necesidad de modificación de la misma dentro del programa.

1.5.-Posibles Soluciones:

Existen dos soluciones posibles para automatizar este proceso y varían en cuanto a la comunicación entre el PC y nuestro controlador. Es posible comunicar con dos interfaces diferentes y dos protocolos diferentes. Solo existen estas opciones debido a que son las únicas que admite el dispositivo.

Opción 1: CENCAL

Utilizar la interfaz por defecto del puerto serie de nuestro PC, es decir, RS-232 y utilizar el protocolo de comunicación CENCAL que es el suministrado por la marca GEFTRAN, fabricante del controlador integrado en el horno.

Opción 2: MODBUS RTU

Utilizar la misma interfaz RS232 de comunicación del PC pero realizar una conversión antes de su conexión al regulador a la interfaz RS-485 que también es admitida por el mismo. En el caso de realizar esta opción el protocolo de comunicación deberá ser MODBUS.

1.6.-Solucion Adoptada:

Finalmente la solución más adecuada para realizar este proceso es la segunda opción, la cual ha sido elegida porque permite utilizar MODBUS, un protocolo sencillo de comunicación que requiere poco trabajo de desarrollo, es público y la transmisión de datos no tiene restricciones. Dentro de este protocolo ha sido seleccionada la versión RTU (Remote Terminal Unit) que será posteriormente explicada.

1.7.-Descripción de lo proyectado:

El proyecto forma parte del campo de las de comunicaciones industriales, área de la tecnología que estudia la transmisión de información entre sistemas electrónicos utilizados para tareas tales como la domótica, la manufactura discreta (cadena de montaje), la manufactura continua (templado del acero, control de procesos biológicos), los medios de transporte o los sistemas integrados o embedded systems (automóvil).

En este caso necesitaremos introducir una serie de datos en nuestro dispositivo maestro (PC) los cuales serán enviados al esclavo (controlador Gefran 800P) el cual actuará en función de dichos datos y permitirá la visualización del proceso enviado.

Además el esclavo devolverá un archivo con los datos del proceso ejecutado. A continuación se explican brevemente las principales partes del proyecto.

INTRODUCCION DE DATOS:

Cabe la posibilidad de introducir los datos de dos maneras diferentes:

- A través de un archivo de formato .txt el cual será explicado posteriormente como debe rellenarse.
- Introduciendo los datos directamente en el programa en el apartado de crear rampa.

COMUNICACIÓN ENTRE EL PC Y EL CONTROLADOR:

La comunicación se realiza con el protocolo MODBUS RTU serie a través de una interfaz RS485 pero como el puerto serie del PC tiene salida RS232 ha sido necesaria la introducción de un adaptador para que la comunicación fuera posible.

VISUALIZACION DEL PROCESO EN EJECUCION:

La rampa introducida para ejecutar se envía al controlador separada en varios pasos y programas, existiendo diferentes modos de configuración para la ejecución que se explican posteriormente. El proceso se ira visualizando simultáneamente a la ejecución del mismo en el horno, es decir, que cada paso que se ejecute en el horno se verá en la gráfica de Labview una vez que se haya realizado.

EXPORTACION DE LOS DATOS:

Para realizar esta función se ha creado una subrutina dentro del programa que nos permitirá guardar la rampa que hemos ejecutado o introducido en el programa, exportándola en extensión .txt con el mismo formato que utilizamos para introducir los datos previamente.

El proyecto puede ser separado en dos grandes apartados:

- hardware: incluye la parte de comunicación entre el PC y el controlador a nivel físico.
- software: incluye la introducción de datos, la parte de programación con Labview de la comunicación, la visualización del proceso en ejecución y la exportación de datos.

1.7.1.- HARDWARE DEL PROYECTO:

Para entender la parte física de la comunicación hay que describir previamente algunos conceptos básicos de la comunicación a través del puerto serie que son relatados a continuación.

1.7.1.1.- INTRODUCCION A LA COMUNICACIÓN SERIE:

Actualmente las comunicaciones industriales han tomado una relevancia importante en la automatización de procesos, debido a que permiten el control y supervisión en tiempo real y a distancia.

A causa de esto el conocimiento de las redes, los códigos y protocolos que manejan resulta imprescindible para permitir automatizar un proceso.

En el caso que nos ocupa, es necesario realizar una comunicación serie entre un PC y un horno Linn VMK 39 que lleva integrado un regulador de la marca GEFRAN modelo 800P. Dicho regulador, el cual controla el horno, admite dos protocolos de comunicación, el protocolo CENCAL (suministrado por la marca GEFRAN el cual tiene un elevado coste) y el protocolo Modbus que es público y su implementación es sencilla. Para realizar la comunicación a nivel físico utilizaremos un cable serie con las interfaces rs232 y rs485 con un conversor entre ellas debido a que el regulador admite la comunicación con esta interfaz.

Las normas que especifican los patrones RS232 y RS485, sin embargo, no especifican el formato ni la secuencia de caracteres para la transmisión y recepción de datos. En este sentido, además de la interface, es necesario identificar también el protocolo utilizado para la comunicación. Entre los diversos protocolos existentes, un protocolo muy utilizado en la industria es el protocolo Modbus debido a que es público, su implementación es fácil y requiere poco desarrollo, y maneja bloques de datos sin suponer restricciones.

Modbus se basa en las algunas de las capas del modelo OSI desarrollado en 1980 por la Organización Internacional de Estándares (ISO), una federación global de organizaciones que representa aproximadamente a 130 países. Consta de siete capas:

1. Física
2. Enlace de Datos
3. Red
4. Transporte
5. Sesión
6. Presentación
7. Aplicación

1. Capa Física:

Es la que se encarga de la topología de la red y de las conexiones globales de la computadora hacia la red, tanto en lo que se refiere al medio físico como a la forma en la que se transmite la información. Se encarga de la transmisión de cadenas de bits no estructurados sobre el medio físico; está relacionada con las características mecánicas, eléctricas, funcionales y de procedimiento para acceder al medio físico.

2. Capa de Enlace de Datos:

Esta capa se ocupa del direccionamiento físico, del acceso al medio, de la detección de errores, de la distribución ordenada de tramas y del control del flujo. Proporciona un servicio de transferencia de datos seguro a través del enlace físico; envía bloques de datos (tramas) llevando a cabo la sincronización, el control de errores y de flujo necesarios.

3. Capa de Red:

Se encarga de identificar las rutas existentes entre una o más redes. Las unidades de información se denominan paquetes, y se pueden clasificar en protocolos enrutables y protocolos de enrutamiento. Proporciona independencia a los niveles superiores respecto a las técnicas de conmutación y de transmisión utilizadas para conectar los sistemas; es responsable del establecimiento, mantenimiento y cierre de las conexiones.

4. Capa de Transporte:

Encargada de efectuar el transporte de los datos, proporciona seguridad, transferencia transparente de datos entre los puntos finales; proporciona además procedimientos de recuperación de errores y control de flujo origen-destino.

5. Capa de Sesión:

Desempeña el control de la comunicación entre las aplicaciones; establece, gestiona y cierra las conexiones (sesiones) entre las aplicaciones cooperadoras.

6. Capa de Presentación :

Se encarga de la representación de la información, para que aunque distintos equipos tengan diferentes representaciones internas de caracteres los datos lleguen de manera reconocible. Trabaja con la sintaxis de los datos transmitidos.

7. Capa de Aplicación :

Proporciona el acceso a los servicios de las demás capas para los usuarios y define los protocolos que utilizan las aplicaciones para intercambiar datos, también proporciona servicios de información distribuida.



Figura 3. Pila OSI de ISO.

1.7.1.2.- INTERFAZ SERIE:

Para realizar la comunicación es preciso conectar el puerto serie del PC al regulador. Para el caso que nos ocupa el PC tiene una interfaz de salida RS-232 pero el GEFran 800P necesita una comunicación a través de una interfaz RS-485. La interfaz de comunicación ocupa el nivel físico de la pila OSI de ISO. A continuación se explican ambas interfaces y el modo de conversión de una a otra.

1.7.1.2.1.-Interfaz RS232:

El protocolo RS-232 es una norma o estándar mundial que rige los parámetros de uno de los modos de comunicación serial asíncrona y síncrona para velocidades de comunicación bajas, de no más de 20 kbps. Por medio de esta norma se estandarizan las velocidades de transferencia de datos, el modo de control de flujo que utiliza dicha transferencia, los niveles de voltajes utilizados, el tipo de cable permitido, las distancias entre equipos, los conectores, etc...

En los inicios esta norma fue diseñada para la comunicación con un modem por lo que dispone de 25 líneas para la transmisión pero para nuestro caso son necesarias menos.

En concreto las líneas de transmisión (Tx), recepción (Rx), y tierra (GND) son las más importantes. A continuación se explican el resto de líneas utilizadas en nuestro caso (figuras 4 y 5).

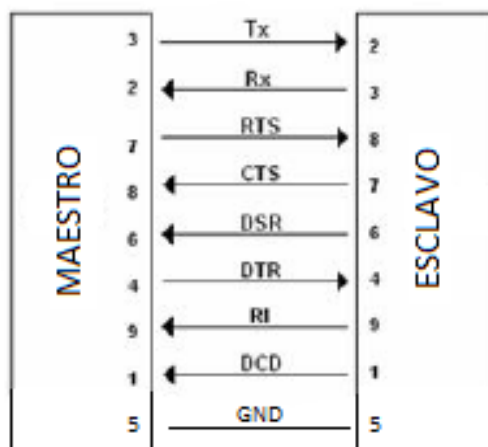


Figura 4. Conexión de líneas de interfaz RS232 maestro/esclavo (imagen descargada de <http://www.puntofotante.net/RS485.htm>)

Señal	Función
GND	Esta línea proporciona la referencia de tierra para todas las demás señales.
Tx <i>Transmitted data</i>	Por esta línea se transmiten los datos. Cuando no hay transmisión de datos, la señal se mantiene en la condición de marca (1 lógico, voltaje negativo).
Rx <i>Received data</i>	Por esta línea se reciben los datos. Cuando no se están recibiendo datos, la señal se mantiene en la condición de marca (1 lógico, voltaje negativo).
DSR <i>Data Set Ready</i>	Esta señal es sostenida en 0 lógico (voltaje positivo) por el DCE para indicar al DTE que está conectado a la línea
CTS <i>Clear to Send</i>	Esta señal es sostenida en 0 lógico (voltaje positivo) por el DCE para informar al dispositivo DTE que la transmisión puede comenzar. RTS y CTS son usados comúnmente como señales para controlar el flujo de datos dentro del dispositivo DCE.
RTS <i>Request to Send</i>	Esta señal es sostenida en 0 lógico (voltaje positivo) por el DTE para que el DCE se prepare a recibir el dato enviado. Tal preparación podría incluir la habilitación de los circuitos de recepción, o la colocación de la dirección del canal en aplicaciones half-duplex. Cuando el DCE está listo, éste acepta por medio de la línea CTS.
DTR <i>Data Terminal Ready</i>	Esta señal es sostenida en 0 lógico (voltaje positivo) por el dispositivo DTE cuando éste quiere comenzar la comunicación.
RI <i>Indicador Ring</i>	Esta señal es relevante cuando el dispositivo DCE es un módem, y es mantenido en 0 lógico (voltaje positivo) cuando una llamada está siendo recibida de la línea telefónica
DCD <i>Data Carrier Detect</i>	Esta señal es sostenida en 0 lógico (voltaje positivo) por el DCE para avisar al DTE que está recibiendo una señal portadora con información.

Figura 5. Líneas de señal interfaz RS-232 conector DB9.

A nivel de software, la configuración principal que se debe dar a una conexión a través de puertos seriales RS-232 es básicamente la dirección del esclavo con el que se quiere comunicar, la selección de la velocidad en baudios (1200, 2400, 4800... hasta un límite de 19200), la verificación de datos o paridad (paridad par, paridad impar o sin paridad), los bits de parada de cada dato (1 ó 2), y la cantidad de bits por dato (7 u 8), que se utiliza para cada símbolo o carácter enviado.

- Características Eléctricas:

La Norma RS-232 fue definida para conectar un ordenador a un modem. Además de transmitirse los datos de una forma serie asíncrona son necesarias una serie de señales adicionales, que se definen en la norma. Las tensiones empleadas están comprendidas entre +15/-15 voltios. Los rangos de voltaje están entre -12 y -3 V para el 1 lógico, entre 12 y 3 V para el 0 lógico y entre -3 y 3 V es el rango conocido como zona muerta o región de transición.

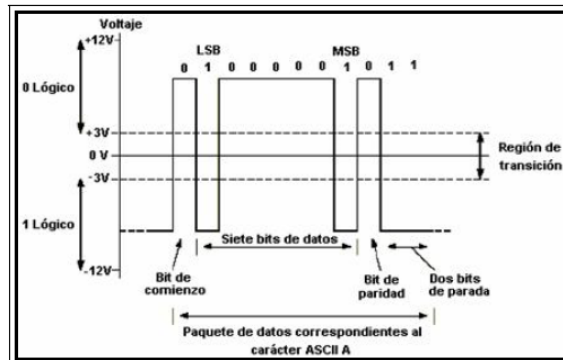


Figura 6. Transmisión de un carácter ASCII en interfaz RS-232.
 (imagen descargada de <http://www.puntofotante.net/RS485.htm>)

- Transmisión de información:

En la figura anterior se puede observar la transmisión para un carácter ASCII, la estructura del mensaje sería 1 bit de inicio, 8 bits de datos, 1 bit de paridad y 2 bits de parada.

Permite comunicaciones simplex, half dúplex y full dúplex; esto significa que es posible enviar solamente en una dirección (simplex), enviar y recibir datos en ambas direcciones pero no simultáneamente (half dúplex) y enviar y recibir simultáneamente en ambas direcciones (full dúplex).

- Características físicas:

La norma sugiere un cable de 15 m de cobre y una velocidad máxima de transmisión de 19200 baudios. El conector más habitual para esta interfaz es el DB9 pero admite también conexiones con DB25, EIA/TIA 561, Host, RJ50 y MMJ (figura 7).



Figura 7. Conectores DB25, DB9 y EIA/TIA 561

- Inconvenientes rs232:

Los inconvenientes de esta norma es que solo permite la comunicación entre dos dispositivos (comunicación punto-punto), la distancia entre ambos dispositivos debe ser inferior a 15 m para que la comunicación sea efectiva, no es inmune al ruido debido a que a una línea referenciada a tierra es imposible bloquearle efectivamente el ruido, al aumentar distancia y velocidad de transmisión la capacidad de la línea introduce fuertes interferencias que pueden llegar a distorsionar el dato hasta el límite de no permitir su lectura.

1.7.1.2.2.-Interfaz rs485:

La interfaz RS-485 utiliza el sistema de comunicación de línea balanceada:

ESTANDAR DE LINEA BALANCEADA:

Un estándar de línea balanceada está fundamentado en el uso de un par de alambres para llevar cada señal. Los datos se envían y reciben por cada par, como un voltaje diferencial entre ambos, llamados A (voltaje negativo) y B (voltaje positivo).

Los estados lógicos se definen de la siguiente manera:

-cuando B tiene voltaje positivo respecto de A se transmite un 0 lógico.

-cuando B tiene voltaje negativo respecto de A se transmite un 1 lógico.

Los dispositivos que envían información por voltajes diferenciales reciben el nombre de “driver” y aquellos que reciben la información de esta manera son definidos como “receiver”. Aunque las líneas balanceadas no necesitan una señal de tierra para realizar el envío de datos, sí que es necesario para mantener el voltaje en modo común dentro de un rango seguro. El voltaje en modo común se define como la suma de tensiones de A y B dividido entre 2, es la media de tensiones respecto a tierra.

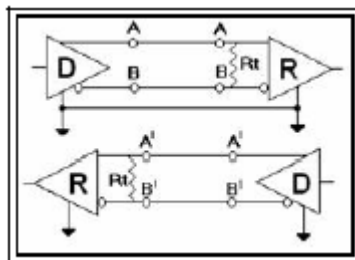


Figura 8. Conexión punto-punto

Como trabajan con voltaje diferencial no les afecta la diferencia de tensiones entre tierra-emisor y tierra-receptor.

Cuando la impedancia de carga, línea y fuente son diferentes se crea una reflexión de la señal en la línea de transmisión lo que puede producir distorsión en el envío a largas distancias y altas velocidades. Por ello se coloca una impedancia al final del bus que suele ser una resistencia (Rt) para que esto no ocurra.

ESTANDAR RS485:

Es una variante de RS-422 denominada bus de transmisión RS-485 (también conocido como EIA-485), que se implementó para permitir conexiones multipunto entre dispositivos a mayor velocidad y distancia. Está definido como un sistema en bus de transmisión multipunto diferencial. La diferencia principal que existe entre el bus RS-232 y RS-485 es que con el primer bus la comunicación es entre dos equipos, mientras que en el segundo la comunicación puede ser hasta con 31 dispositivos. Además la distancia que separa los dispositivos es mayor para el puerto RS-485: 1200 metros. Se puede decir que este puerto es ideal para transmitir a altas velocidades (permite velocidades entre 9.6 kbit/s y 12 Mbit/s) sobre largas distancias y reduciendo los ruidos que aparecen en la línea de transmisión.

La comunicación es half-duplex (semiduplex) y la transmisión diferencial permite múltiples drivers dando la posibilidad de una configuración multipunto. Al tratarse de un estándar bastante abierto permite muchas y muy diferentes configuraciones y usos. Los niveles de tensión que maneja para las líneas A y B se pueden ver en la figura 9.

Driver		Receiver	
1 Lógico	0 Lógico	1 Lógico	0 Lógico
$V_B - V_A < -1.5 V$	$V_B - V_A > 1.5 V$	$V_B - V_A < -0.2 V$	$V_B - V_A > 0.2 V$

Figura 9. Valores de tensión rs485.

Para nuestro caso se utiliza una conexión punto-punto ya que solo debemos conectar dos dispositivos entre sí (figura 8).

1.7.1.2.3.-Conversión rs232 a rs485:

En el caso que nos ocupa debemos utilizar la norma rs232 que es la que nos proporciona nuestro PC y realizar una conversión a la norma rs485 que es la que admite el dispositivo implementado en el horno. Por tanto deberemos utilizar una tarjeta conversora entre dichas interfaces, la cual adecua las señales de transmisión y recepción a las tensiones que el controlador es capaz de analizar. En la figura de la parte inferior se puede ver un ejemplo de este tipo de circuito.

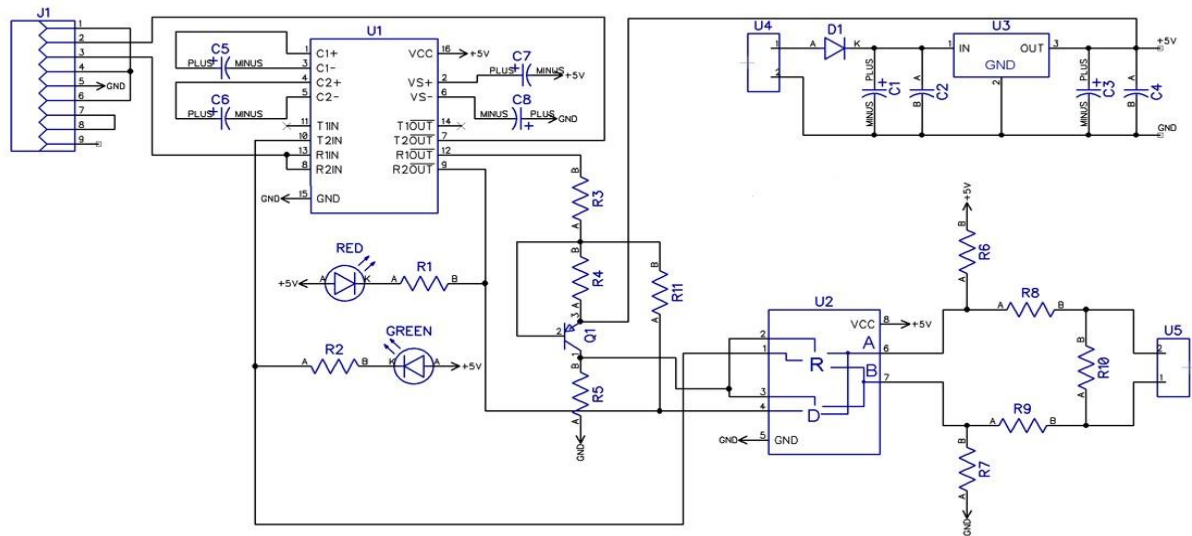


Figura 10. Circuito conversor de RS232 a RS485. (imagen descargada de http://www.eeweb.com/project/circuit_projects/rs232-rs485-converter-with-automatic-rx-tx-control)

1.7.1.2.3.1.-Fundamentos de la conversión:

Para realizar esta conversión de señal, se debe adecuar los niveles de tensión generados por el puerto serie los cuales oscilan de -3 a -12 V para el 1 lógico y de 3 a 12V para el 0 lógico. Para ello el pin 3 de transmisión o Tx del conector DB9 de la entrada RS232 debemos conectarlo a un dispositivo integrado denominado MAX232 el cual adecua la señal a niveles TTL/CMOS los cuales oscilan entre 0 y 5V. La salida de este integrado se conecta después de pasar por un circuito formado por divisores de tensión y un transistor que actúa de interruptor a otro integrado denominado MAX485 el cual es un transductor de baja tensión para comunicación RS485 y RS422. Cada parte del integrado contiene un emisor y un receptor. De las salidas de este integrado sale una señal que solamente debe pasar por un circuito de resistencias para que finalmente la señal salga adecuada para que la interfaz RS485 del receptor sea capaz de recibirla.

De la misma manera el camino de la recepción será similar pero a la inversa. Serán emitidos unos datos desde la interfaz RS485, que entraran en el MAX 485 en su parte de receiver, el cual adecuará la señal a niveles TTL/CMOS para que puedan ser identificados por el MAX232 y este vuelva a realizar otra conversión a los niveles de comunicación del puerto serie con la norma RS232. En las figuras se puede ver los valores de tensiones en la conversión en ambos sentidos.

Estados	Niveles de Tensión		
	RS-232	TTL/CMOS	RS-485
0 Lógico	$3V < L < 12V$	$0V < L < 0.8V$	$V_B - V_A > 2V$
1 Lógico	$-12V < H < -3V$	$3.5V < H < 5V$	$V_B - V_A < -2V$

Figura 11. Valores de tensión de rs232 a rs485

Estados	Niveles de Tensión		
	RS-485	TTL/CMOS	RS-232
0 Lógico	$V_B - V_A > 0.2V$	$0V < L < 0.8V$	$3V < L < 12V$
1 Lógico	$V_B - V_A < -0.2V$	$3.5V < H < 5V$	$-12V < H < -3V$

Figura 12. Valores de tensión de rs485 a rs232

A continuación se explica más detalladamente el funcionamiento de ambos integrados como las diferentes conexiones de los pines de ambas normas de comunicación, la cual sigue el siguiente esquema (figura 13):

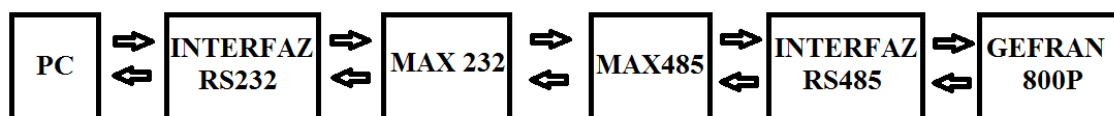


Figura 13. Esquema conversión de interfaces.

- Conector DB9 norma RS232.
- Circuito Integrado MAX232.
- Circuito interruptor.
- Circuito Integrado MAX485.
- Conector norma RS485.

-CONECTOR DB9 RS232:

El conector DB9 es la salida más común para el puerto serie de un PC, es un conector que tiene 9 pines (figura 14) de los cuales 2 de ellos son los encargados de la transmisión y recepción de información, tiene también un pin para la conexión a tierra (GND pin 5). Estos pines son el Rx y Tx (pines 2 y 3). El resto de pines tienen una función de control del flujo de la comunicación, es decir, habilitan o deshabilitan al dispositivo maestro para que transmita o no y ocurre lo mismo con el esclavo. En la siguiente figura podemos observar la configuración del conector.

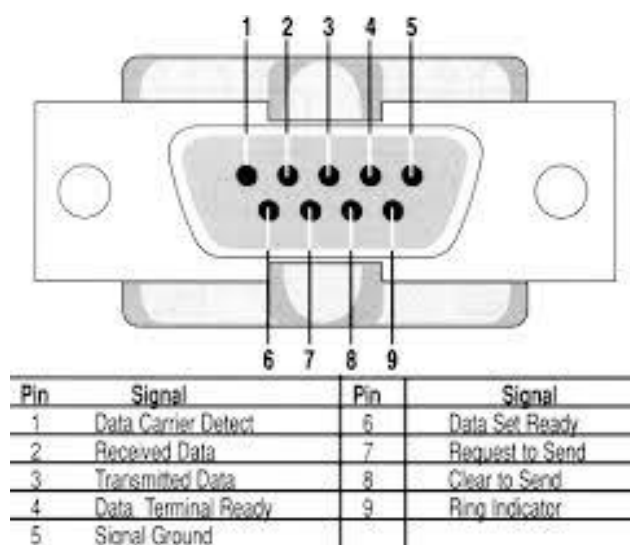


Figura 14. Denominación de los pines de conector DB9 para interfaz RS232

-CIRCUITO INTEGRADO MAX232:

Este circuito integrado creado por la marca Maxim (figura 15) convierte la señal de un puerto serie RS232 como el de nuestro PC a una señal compatible con valores de tecnología TTL/CMOS para los circuitos lógicos y viceversa. El MAX232 sirve como interfaz de transmisión y recepción de datos.

Las salidas de este componente están preparadas para manejar valores de tensión semejantes a los de la norma RS232 gracias a los multiplicadores de voltaje que llevan internamente, a los condensadores que se deben conectar a sus patillas y al voltaje de alimentación de +5V que debe ser conectado.

Las entradas de recepción de RS-232 (las cuales pueden llegar a ± 25 V), se convierten al nivel estándar de 5 V de la lógica TTL/CMOS, estos receptores tienen un umbral típico de 1.3 V, y una histéresis de 0.5 V.

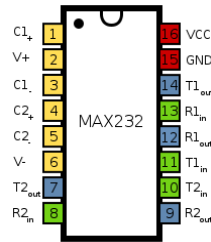


Figura 15. Configuración de pines Circuito Integrado MAX232

Cuando un circuito integrado MAX232 recibe un nivel TTL/CMOS (este valor oscila entre 0 y 5 V para los niveles lógicos 0 o 1 respectivamente) lo transforma, es decir, cambia el nivel lógico TTL/CMOS de 0 a un nivel comprendido entre +3 y +15 V, y cambia un nivel lógico TTL/CMOS 1 a un nivel comprendido entre -3 a -15 V, y viceversa, para convertir niveles de RS232 a TTL/CMOS.

Para más información al respecto del integrado se adjunta el datasheet del mismo en el anexo 1.

-CIRCUITO INTERRUPTOR(figura 16):

Esta parte del circuito tiene la función de hacer llegar la señal TTL/CMOS (datos) generada por el max232 al integrado max485 y habilitar o deshabilitar el funcionamiento del mismo. Como más adelante se explica el integrado max485 tiene dos patillas de habilitación de sus circuitos internos, el circuito interruptor esta constituido por un transistor PNP BC327 que actúa de interruptor y por varias resistencias necesarias para el funcionamiento de este. Dependiendo de los pulsos que lleguen del max232 este circuito mandará un pulso high o low a las patillas del max485 lo cual permitirá la comunicación en uno de los dos sentidos pero no así en el otro.

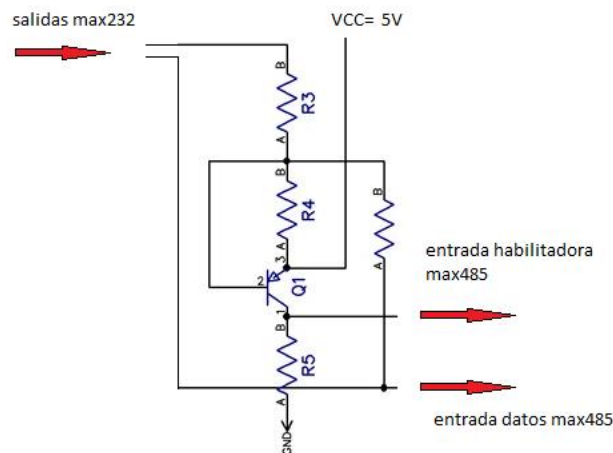


Figura 16. Circuito interruptor del esquema adquirido de internet.

-CIRCUITO INTEGRADO MAX485:

El MAX485 (figura 17) es un integrado que adecua una señal TTL/CMOS a una señal de RS485 o RS422 y viceversa. Se trata de un transductor de baja potencia para esas interfaces. Cada parte está constituida por un driver (conductor) y un receiver (receptor).

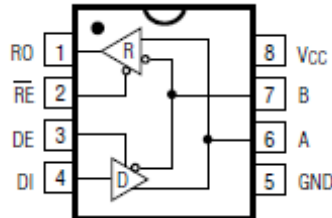


Figura 17. Configuración de pines Circuito Integrado MAX485(anexo 3.3).

La velocidad de respuesta de este transductor le permite transmitir hasta 2.5Mbps. Este integrado absorbe unas intensidades entre $120\mu\text{A}$ y $500\mu\text{A}$ de la corriente de suministro cuando están descargados o cuando están totalmente cargados con los conductores deshabilitados. Este integrado permite una comunicación half-duplex entre dos dispositivos. El integrado tiene dos patillas a través de las cuales es alimentado, Vcc (8) estará conectado a un voltaje de 5 V y GND (5) la cual estará conectada a la tierra del circuito.

Para que una señal TTL/CMOS sea transmitida a la interfaz actúan las patillas 3 y 4 del integrado. La patilla 3 habilita el funcionamiento del driver(conductor) y por la 4 entran los paquetes de datos. En el caso que entre un 1 lógico por DI obtendremos un 1 en A y un 0 en B; y en el caso de que entre un 0 lógico A tendrá un 0 y B un 1.

De la misma manera pero realizando la comunicación en el sentido opuesto, los datos enviados desde la interfaz rs485 o rs422 se adecuarán a una señal TTL/CMOS. Para ello es necesario que en las patillas 6 y 7 (A y B) deben tener una diferencia de potencial de $+200\text{mV}$ para el 1 lógico y de -200mV para el 0 lógico entre ellas, realizando la comparación de estos dos valores de voltaje obtendremos los datos en la patilla RO (1).

Para que esta dirección de comunicación sea posible la patilla RE (2) debe tener un 0 lógico conectado. Se adjunta el datasheet del integrado donde viene explicado todo en detalle y se pueden observar todas las características del mismo (anexo 3.2).

-CONECTOR NORMA RS485:

Este conector puede variar dependiendo del dispositivo al que queramos conectar la salida de la tarjeta conversora. Es posible utilizar un conector DB9 como el de la entrada. Sin embargo existe la posibilidad de utilizar otros conectores indicados en el apartado de ejemplos de conector para la norma RS-232.

En el caso que nos ocupa en un momento inicial al tener un conector DB9 en la entrada de nuestro regulador parecía la mejor opción utilizar uno similar a la salida de la tarjeta. Pero una vez abierto el horno y observado las conexiones del conector al regulador, se puede ver que solamente utiliza dos pines de los 9 existentes. Además la configuración de conexiones recomendada por el fabricante (GEFRAN) ilustra en una imagen, a continuación adjuntada, cómo deben conectarse dichos pines.



Figura 18. Configuración conexionado (imagen extraída del datasheet del Gefran 800P, anexo3.5)

El conector utilizado para conectar al controlador GEFTRAN 800P es un DB9 al que solo se conectan los pines A y B de la salida de la interfaz RS485. Estos pines se conectan a las patillas 4 y 5 del conector DB9, los cuales son conectados de la manera indicada en la figura 18.

1.7.1.2.3.2.-PRUEBAS TARJETAS CONVERSoras:

Para conseguir que esta parte del proyecto funcionara han sido necesarios varios ensayos variando partes de software y hardware. A continuación se explican todos ellos.

A.-Hardware utilizado (tarjetas conversoras):

Se ha precisado realizar ensayos con 4 tipos de tarjetas conversoras diferentes hasta encontrar la adecuada. Una vez delimitada la configuración necesaria para la comunicación entre nuestro pc y el dispositivo, sabiendo que era necesaria una conversión de los datos de salida del pc a la norma rs485 se procedió a adquirir el circuito necesario para este propósito.

La primera tarjeta (figura 19 izquierda) fue comprada en la web [www.todoelectronica.com](http://todoelectronica.com):<http://todoelectronica.com/conversor-interfaces-rs232-rs485-p-6831.html> , se realizaron los ensayos necesarios (explicados en el apartado siguiente) para ver si la comunicación funcionaba de manera adecuada lo cual no fue así. Se adquirió la segunda después de realizar las medidas oportunas con un osciloscopio en el circuito de la primera tarjeta, viendo que las medidas eran correctas pero el funcionamiento de la comunicación no.

Con la segunda tarjeta (figura 19 derecha) adquirida en la web www.amazon.es:

http://www.amazon.es/gp/product/B00CL5B902/ref=oh_details_o00_s00_i00?ie=UTF8&psc=1 el proceso fue el mismo con similar resultado.

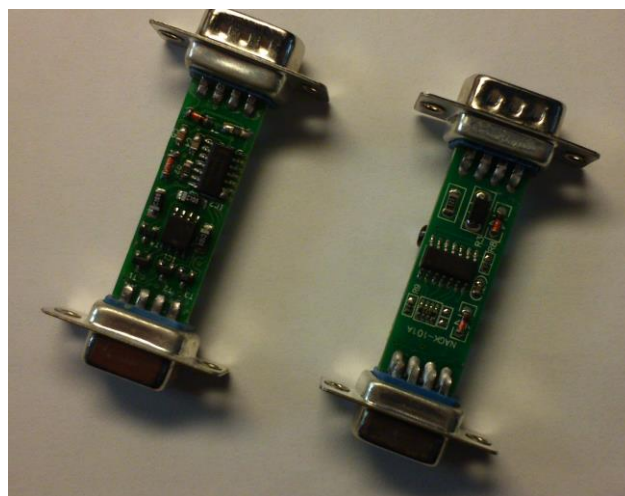


Figura 19. Tarjetas para conversión interfaces RS232/RS485.

También se recurrió a fabricar una tarjeta propia con un diseño descargado de internet (figura 10). Este diseño fue simulado previamente con el software “Proteus” con resultado satisfactorio. Una vez montado se realizó el mismo proceso que para las tarjetas anteriores sin obtener ningún avance en cuanto a la comunicación.

Algo que tenían en común los tres tipos de circuito utilizados para la comunicación era que no tenían el control de flujo de comunicación automático. Esto provocaba que se pudieran transmitir los datos desde el PC al dispositivo pero no a la inversa, debido a que el canal de comunicación quedaba bloqueado y no permitía la recepción de los datos.

Una vez adquirida la última tarjeta (figura 20) con el control de flujo automático se realizó el proceso de ensayos con resultado positivo.

Esta tarjeta era la recomendada por el fabricante del regulador para este tipo de comunicaciones, la empresa fabricante es B&B Electronics. <http://www.bb-elec.com/Products/Serial-Connectivity/Serial-Converters/Port-Powered-RS-232-to-RS-485-Converters/485SD9TB.aspx>

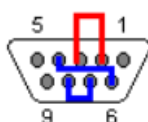


Figura 20. Tarjeta RS232/RS485 B&B Electronics.

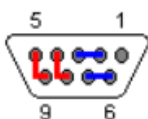
Con este control automático del flujo se simplifica todo, de modo que una vez ha dejado de transmitir el circuito deshabilita esa función y permite la de recepción de los datos, con lo cual el canal se abre en un sentido u otro dependiendo del momento.

B.-Ensayos software (loopback test) con diferentes programas:

Para realizar los ensayos se han utilizado varios programas diferentes, el software Labview con un programa ejemplo descargado de la página web oficial de National Instruments (fabricante), Docklight, Hiperterminal y GF_express. Para ver el funcionamiento de las tarjetas se recurrió a un ensayo denominado “loopback test” (figura 21). Este consiste en puentear los pines de salida de la interfaz que se conecta al dispositivo, de manera que los pines de transmisión y recepción queden conectados (puenteados en rojo) al igual que los pines de control de flujo (puenteados en azul). Con esto conseguimos que los paquetes de datos hagan un circuito cerrado y podamos averiguar en que punto falla la comunicación. En la figura continuación se puede ver el puenteado para RS232 y para RS485.



RS-232 female DE-9 plug with connections required for loopback test



RS-422/485 female DE-9 plug with connections required for loopback test

Figura 21. Conexión para realizar el loopback test en RS232 y RS485.

A continuación se explican los programas utilizados para ver el funcionamiento de las tarjetas.

Labview:

Para realizar estas pruebas se ha utilizado un programa ejemplo descargado de la página web de National Instruments para ensayos de comunicación serial con Modbus y modificado ligeramente para este caso concreto (figura 22). De estos ensayos fue posible averiguar que el principal problema estaba en la lectura de datos del puerto serie, ya que el programa transmitía correctamente pero a la hora de realizar una lectura el programa fallaba en el bloque indicado en rojo en la figura 21. Fue posible averiguar esto gracias al “property node” de lectura de bytes en el puerto serie el cual devolvía constantemente un valor de 0. Para confirmar si el error se encontraba concretamente ahí se continuó probando con otro tipo de software para confirmar las sospechas.

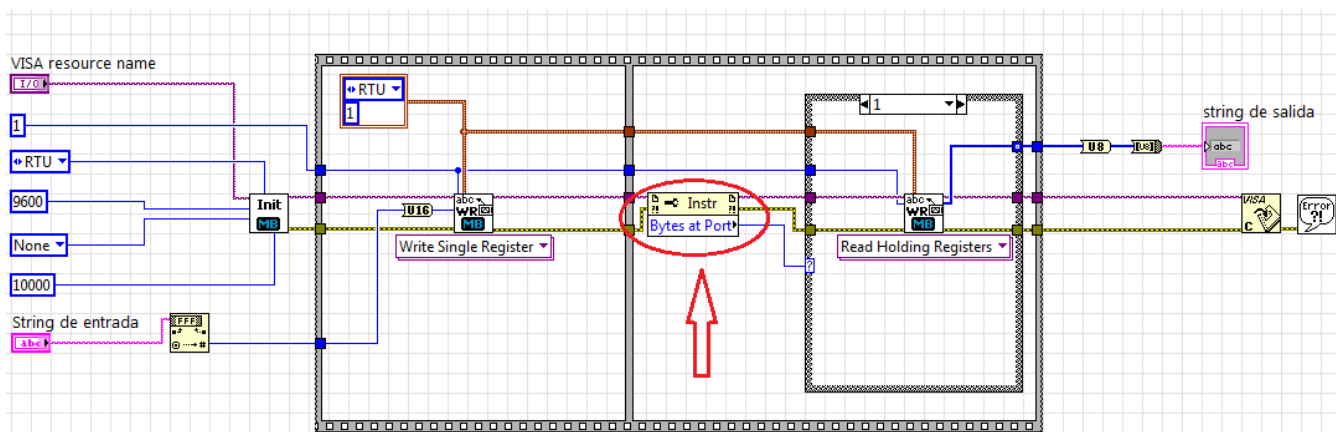


Figura 22. Programa labview ensayos comunicación.

Docklight:

Se trata de una herramienta (figura 23) que permite realizar simulaciones de comunicación serie entre dos dispositivos o solamente con uno de ellos, con los protocolos RS232, RS422 o RS485. Este programa fue descargado del siguiente link: http://download.cnet.com/Docklight/3000-2651_4-196405.html

Para realizar los ensayos se procedió a mandar una trama de datos a ver si la transmisión era correcta. Después de realizar numerosas pruebas, se obtuvo la misma conclusión que con el software anterior, la transmisión era correcta pero no así la recepción de los datos ya que no se recibían las tramas enviadas.

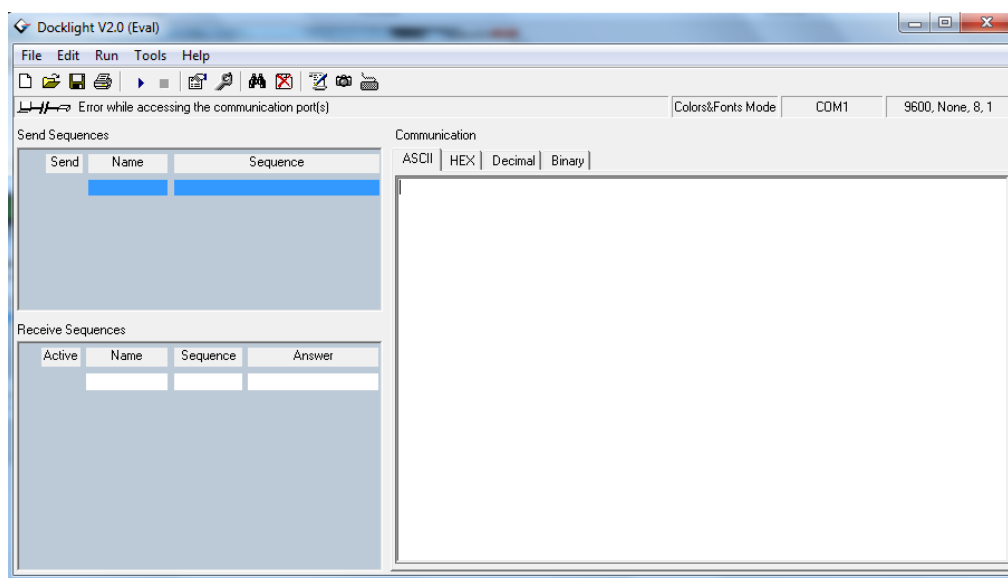


Figura 23. Imagen del programa Docklight para simulaciones de comunicación.

Hiperterminal:

Es una herramienta (figura 24) proporcionada por el sistema operativo Windows que permite realizar comunicaciones y conexiones con otros equipos desde nuestro pc. En este caso lo utilizamos para ver si la comunicación de las tarjetas conversoras es correcta. En este programa como en los anteriores será necesario configurar la comunicación, velocidad de comunicación (baudios), bits de parada, paridad, etc.

Una vez configurado y listo para funcionar se escribe por teclado la trama de datos a enviar, la cual debe aparecer por pantalla. Si esto no ocurre el funcionamiento no es el adecuado. Este programa no muestra demasiada información del proceso de comunicación por lo que no es posible analizar detalladamente dónde se produce el fallo.

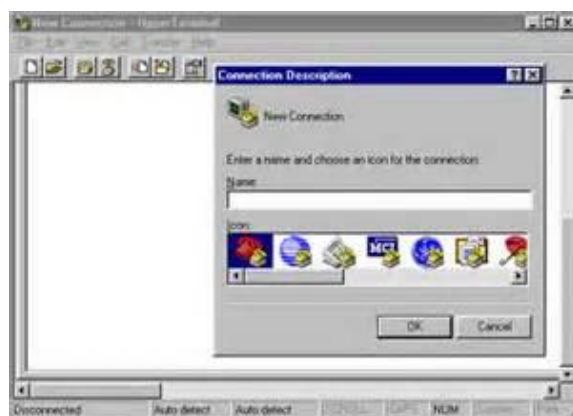


Figura 24. Imagen del programa Hyperterminal para simulaciones de comunicación.

GF_express:

Se trata de una herramienta para realizar ensayos con dispositivos GEFAN (figura 25) suministrado por la propia marca. Se decidió utilizar este programa después de hablar con el fabricante de nuestro dispositivo. Con este programa no se ha recurrido al uso del loopback test, debido a que al ser creado por la propia marca las compatibilidades a nivel de software son mayores que con los otros programas.

Este software permite realizar igual que los programas anteriores ensayos de comunicación serie, con la diferencia que permite visualizar y modificar los registros del dispositivo en cuestión de manera más sencilla.

Para empezar a trabajar con él lo primero que se pide es seleccionar el modelo de controlador que vamos a utilizar en la comunicación, y es muy importante que la configuración del puerto y del dispositivo sea idéntica para que la comunicación funcione adecuadamente.



Figura 25. Imagen del programa GF_Express para simulaciones de comunicación.

En los otros programas era necesario crear un mensaje Modbus para que el dispositivo reaccionara ante él, mientras que con GF_express después de configurar las características de la comunicación y el protocolo utilizado, basta con elegir el registro y modificarlo a gusto del usuario (figura 26).

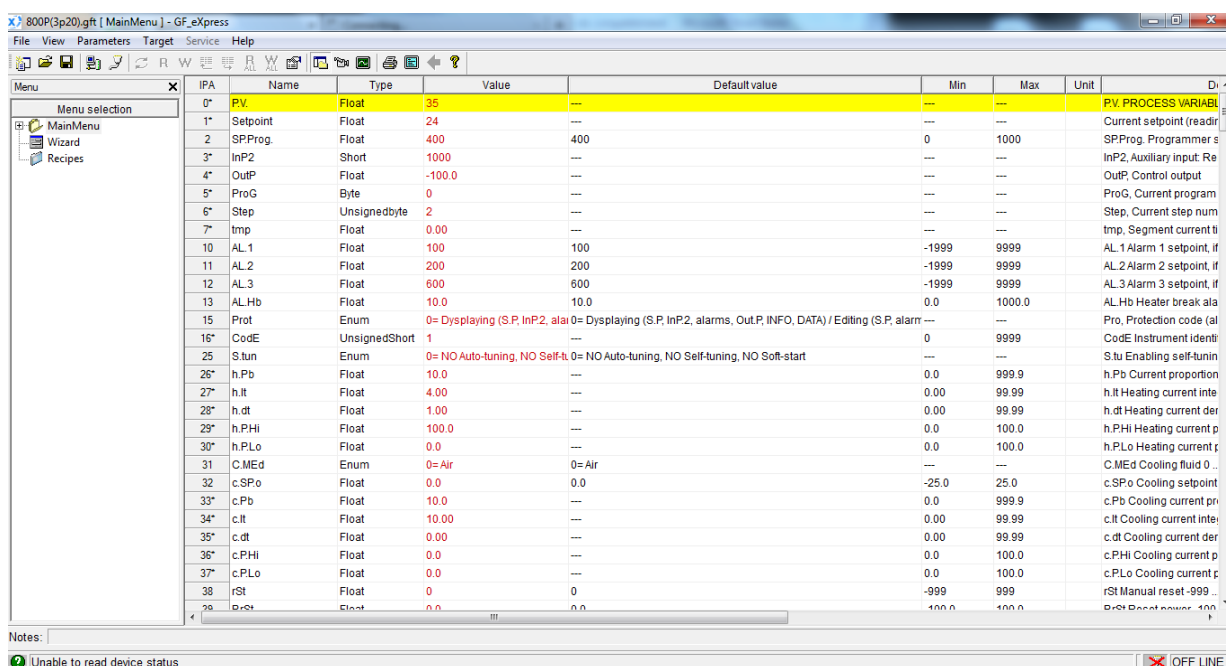


Figura 26. Imagen del programa GF_Express mostrando los registros del 800P.

Fue probado con todos los circuitos convertidores disponibles pero el funcionamiento era similar en ellos, permitían la escritura o transmisión de los mensajes pero la respuesta o lectura de los registros no funcionaba de manera correcta. Esto fue solucionado al adquirir la última tarjeta convertora, que permite un funcionamiento correcto de la comunicación.

1.7.2.-SOFTWARE DEL PROYECTO:

1.7.2.1.-INTRODUCCION A LOS PROTOCOLOS DE COMUNICACIÓN:

Un protocolo de comunicación está constituido por un conjunto de reglas y convenciones entre los dispositivos a comunicar. El objetivo por el que se utilizan protocolos es establecer la conexión y permitir asegurar la transferencia de datos entre emisor y receptor con la certeza de que todos los datos lleguen correctamente. Además de delimitar adecuadamente las funciones de los dispositivos en cada caso como receptor o emisor.

Las características principales que definen a un protocolo son las siguientes:

- Modos de operación , estructura de los mensajes y los tipos de órdenes de pregunta y respuesta.
- Las diferentes fases de la comunicación dependiendo de si hay confirmación de las ordenes enviadas:

RESQUEST – un servicio es solicitado por un usuario

INDICATION – un dispositivo es notificado de un evento

RESPONSE – respuesta del dispositivo al evento

CONFIRM – un dispositivo informa de una solicitud anterior

- Los diferentes niveles que solicitan servicios de niveles superiores y/o inferiores.

La estructura de los mensajes es una de las características más importante a tener en cuenta a la hora de realizar la comunicación con un protocolo. En la figura 27 se puede observar una estructura común a muchos protocolos.

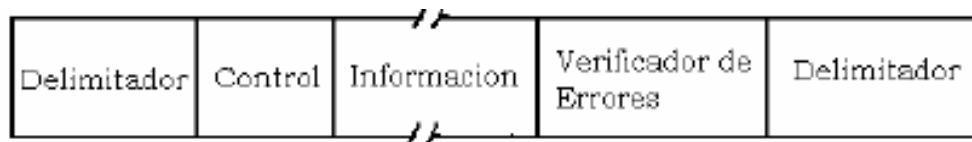


Figura 27. Estructura del mensaje de comunicación

Las partes del mensaje se explican a continuación:

- Delimitador: especifica el inicio y fin del mensaje
- Control: indica el tipo de mensaje, destino del mensaje, numero de secuencia o código de petición o confirmación.
- Información: representa los datos que se quieren enviar, la longitud de este es variable.
- Verificador de errores: este campo sirve como control para ver si la información llega al receptor de manera adecuada y sin ningún error. Se trata de un detector de errores que puede emplear métodos como el CRC (control de redundancia cíclica), CRH (control de redundancia horizontal) y el VRC (control de redundancia vertical).

1.7.2.2.-PROTOCOLO MODBUS:

Modbus se enmarca dentro del concepto de bus de campo de control, emplea los niveles 1(Físico), 2(Enlace de datos) y 7(Aplicación) del modelo OSI. Es un protocolo desarrollado por la empresa Modicon en 1979 para realizar comunicaciones serie. En sus inicios estaba pensado para el uso exclusivo de controladores lógico programables del propio fabricante, pero con el tiempo se ha convertido en uno de los protocolos más comunes de uso industrial, sistemas de telecontrol y monitorización. La idea de este protocolo estaba indicada para comunicaciones seriales pero poco a poco se han creado variantes como la Modbus TCP que permite el encapsulamiento de las tramas serie en tramas Ethernet TCP/IP de manera sencilla. Es posible este encapsulamiento debido a que este protocolo se ubica en la capa de aplicación de la pila OSI.

El objetivo de este protocolo es la transmisión de información entre dos o más dispositivos electrónicos conectados entre sí por un bus (figura 28). En el caso de Modbus TCP/IP la nomenclatura de los dispositivos es de cliente y servidor, la conexión se realiza mediante Ethernet y la transmisión de información debe ser encapsulada en paquetes TCP/IP. En el caso de Modbus serie uno de estos dispositivos deberá tener el rol de maestro y los demás de esclavo pudiendose conectar hasta un máximo de 247 esclavos, dependiendo del modo (en Modbus TCP es mayor).

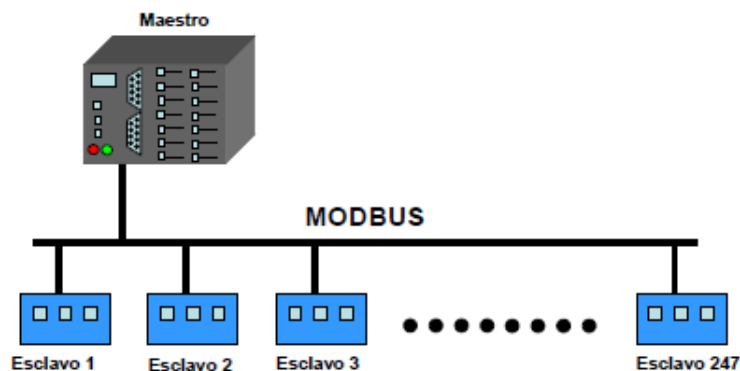


Figura 28. Estructura de una red maestro/esclavo.

Las razones de su popularidad para dentro del sector industrial respecto a otros protocolos son algunas de las siguientes:

- Se trata de un protocolo público que permite a los usuarios utilizarlo libremente sin restricciones, lo que simplifica en gran medida el acceso a la información y a la estructura del mismo.
- Su implementación es muy simple lo que reduce el tiempo de desarrollo. Esto significa que la construcción de las tramas y su posterior acceso a ellas es mucho más fácil debido a que no es necesario el tratamiento de las mismas para su envío, ni su posterior acceso a los datos.
- No existen restricciones a la hora de enviar un tipo de datos u otro, lo que quiere decir que es posible enviar diferentes tipos de datos con este protocolo, ya sean Word con signo, un entero sin signo de 16bits o la parte alta de una representación tipo Float de 32bits, etc.... La única restricción será la correlación que deberá existir entre el tipo de datos enviados por el maestro y el tipo de datos que admita el esclavo.

1.7.2.3.-MODOS DE TRANSMISION MODBUS SERIE:

Se pueden emplear las interfaces RS-232 y RS-485, pudiendo existir dos modos diferentes dentro este tipo de comunicación::

- Modbus RTU: representación en código binario compacto de los datos.
- Modbus ASCII: representación en código ASCII del protocolo.

En la figura 29 se muestra un ejemplo de envío de un intercambio de tramas de datos entre un maestro y un esclavo por Modbus serie (en los siguientes apartados se mostrará la diferencia entre la versión ASCII y RTU).

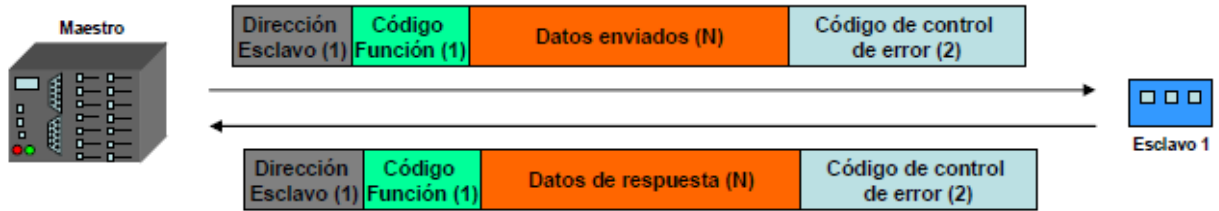
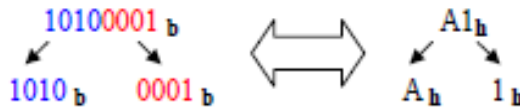


Figura 29. Modelo de envío y recepción de datos en una red maestro/esclavo.

1.7.2.4.-MODBUS ASCII:

Este modo de transmisión se utiliza cuando la comunicación física o las capacidades de los elementos no permiten el uso del modo RTU, ya que es menos eficiente debido a la necesidad de utilizar dos caracteres para enviar un byte. Las siglas ASCII se traducen a Código Estándar Americano para Intercambio de Información. Se trata de un código alfanumérico utilizado por la mayoría de equipos electrónicos y ordenadores. La comunicación debe iniciarse con una solicitud del maestro al esclavo y una respuesta de este. En todos los mensajes la estructura es la misma, solo varía el tamaño del mensaje dependiendo del campo de datos. Se envían dos caracteres ASCII que representan cada uno de ellos un byte en hexadecimal (los caracteres pueden ser '0-9' y 'A-F'). La información es codificada de la siguiente manera:

-cada byte de información se divide en dos caracteres:



-cada carácter se transforma en un carácter ASCII sumándole un valor:

-30_h si el carácter esta entre 0_h y 9_h

-37_h si el carácter esta entre A_h y F_h

$$\begin{array}{l}
 1010 = A_h \xrightarrow{+37_h} 41_h \text{ Código ASCII} \\
 0001 = 1_h \xrightarrow{+30_h} 31_h \text{ Código ASCII}
 \end{array}$$

De esta manera se obtiene que 10100001 = 41_h31_h ASCII

Posteriormente cada carácter en código ASCII se empaqueta en un carácter serie ASCII (figura 30), el empaquetado viene explicado en la siguiente figura. Está delimitado por 1 bit de inicio, 7 u 8 bits de datos (carácter ASCII), 1 bit de paridad el cual es opcional y 1 bit de fin, 2 en el caso que la paridad no se calcule.

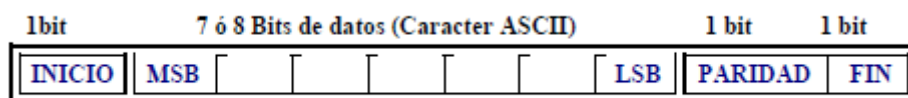


Figura 30.Estructura del empaquetado de un carácter serie ASCII.

Este empaquetado del código ASCII será el campo de datos en el mensaje Modbus que se enviará, el cual es de tamaño variable dependiendo de los datos que deban ser enviados.

Descripción de la estructura del mensaje:

INICIO	DIRECCIÓN	FUNCION	DATOS	CHEQUEO	FIN
1 CAR : 3A hex	2 CAR	2 CAR	N CAR	2 CAR	2 CAR CRLF D Y A hex

Donde CAR representa un carácter ASCII.

Figura 31.Estructura de un mensaje ASCII.

Esta es la estructura (figura 31)más común utilizada, y su tamaño máximo es de 513 caracteres. En ella se pueden ver los diferentes campos:

-INICIO: se utiliza el carácter “:” en código ASCII equiparable a 3A en hexadecimal.

-DIRECCION: se trata de la dirección de esclavo, es el número asignado al esclavo al cual se quiere enviar o demandar la información. Tiene un tamaño de 2 caracteres ASCII.

-FUNCION: se trata de la acción que debe realizar el esclavo, Modbus permite varias funciones que son explicadas a posteriori. El tamaño de este campo es de 2 caracteres ASCII.

-DATOS: este campo depende en gran medida de la función, ya que condiciona el formato y tamaño del mismo. Su longitud es variable con un máximo de 2x252 caracteres ASCII, lo que sería el doble que en el modo RTU.

-CHEQUEO: es el encargado del chequeo de errores, y utiliza el LRC o control de redundancia longitudinal. Este método consiste en verificar la integridad del bit de paridad de un grupo de caracteres.

-FIN: se utilizan los caracteres D y A del código hexadecimal para delimitar el fin del mensaje. El tamaño es de dos caracteres ASCII.

1.7.2.5.-MODBUS RTU (Unidad de Terminal Remota):

En este modo de transmisión por cada byte enviado se envían dos caracteres hexadecimales de 4 bit. La ventaja de utilizar este modo respecto al ASCII es que aumenta el rendimiento de la velocidad de transmisión debido a la gran densidad de carácter. La comunicación debe iniciarse con una solicitud del maestro al esclavo y una respuesta de este. En todos los mensajes la estructura es la misma, solo varía el tamaño del mensaje dependiendo del campo de datos.

Los datos se deben enviar dentro del campo de información y deben ser enviados como se especifica continuación. Cada carácter se empaqueta en un carácter serie RTU. El empaquetado viene explicado en la figura 32. La estructura que sigue es 1 bit de inicio, 8 bits de datos (1 byte de información), 1 bit de paridad (opcional) y 1 o 2 bits de fin dependiendo de si hay o no paridad.

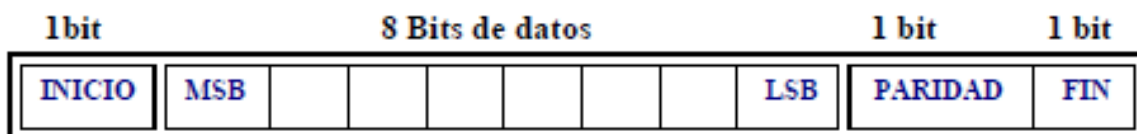


Figura 32. Estructura del empaquetado de un carácter serie RTU.

Este empaquetado del código será el campo de datos en el mensaje Modbus que se enviará, el cual es de tamaño variable dependiendo de los datos que deban ser enviados.

Descripción de la estructura del mensaje:



Figura 33. Estructura del mensaje RTU.

INICIO: El mensaje Modbus RTU inicia con un delimitador de inicio de trama. En este intervalo no hay transmisión alguna. Se trata de un silencio de duración similar a 3.5 veces el tiempo de transmisión de un carácter en Modbus RTU. El cálculo de este intervalo se consigue esperando el tiempo de transmisión de un carácter que es inversamente proporcional a la velocidad en baudios y multiplicando ese valor por 3.5.

DIRECCION: Este campo es similar al del modo ASCII, delimita el número del dispositivo esclavo con el que se quiere comunicar. En caso de ser 0 lo enviará a todos los dispositivos de la red. Varía respecto al otro modelo ASCII en lo referente al tamaño que en este caso será de 1 byte, mientras que en el otro era de 2.

FUNCION: se trata de la acción que debe realizar el esclavo, Modbus permite varias funciones que son explicadas a posteriori. El funcionamiento es similar al del modo ASCII. El tamaño de este campo es de 1 byte.

DATOS: Dentro de esta parte del mensaje se adjuntan los datos a enviar al esclavo o esclavos. Estos suelen ser los registros internos del esclavo que se quieren modificar o leer. El tamaño de este campo es variable dependiendo de la función utilizada en el campo anterior. Será de N* bytes.

CHEQUEO: Se trata de un control de errores para que la información se transmita de manera adecuada. En este caso se utiliza el método CRC-16 o control de redundancia cíclica (figura 34). El tamaño de este campo es de 2 bytes, de los cuales el primero es el byte menos significativo CRC- y el segundo el más significativo CRC+.

Para el cálculo del CRC se sigue el proceso a continuación explicado(figura 33):

- 1.- Se coge el primer byte del mensaje (solo el campo de datos) conectado a una puerta lógica XOR con los 8 bits menos significativos del CRC (cuyo valor inicialmente es de $CRC=FFFF_h$) llevando el resultado de la XOR a unir con el valor más significativo del CRC.
- 2.- Se desplaza el valor de esta unión una posición a la derecha en dirección al bit menos significativo y se rellena el más significativo con un valor de 0.
- 3.- A posteriori se compara el valor que ha sido desplazado. En el caso que sea 0 no se ha hecho el cálculo del CRC. Si por el contrario el valor es de 1, el valor del CRC entra a una puerta lógica XOR con un valor $A001_h$. Este proceso debe repetirse hasta 8 veces.

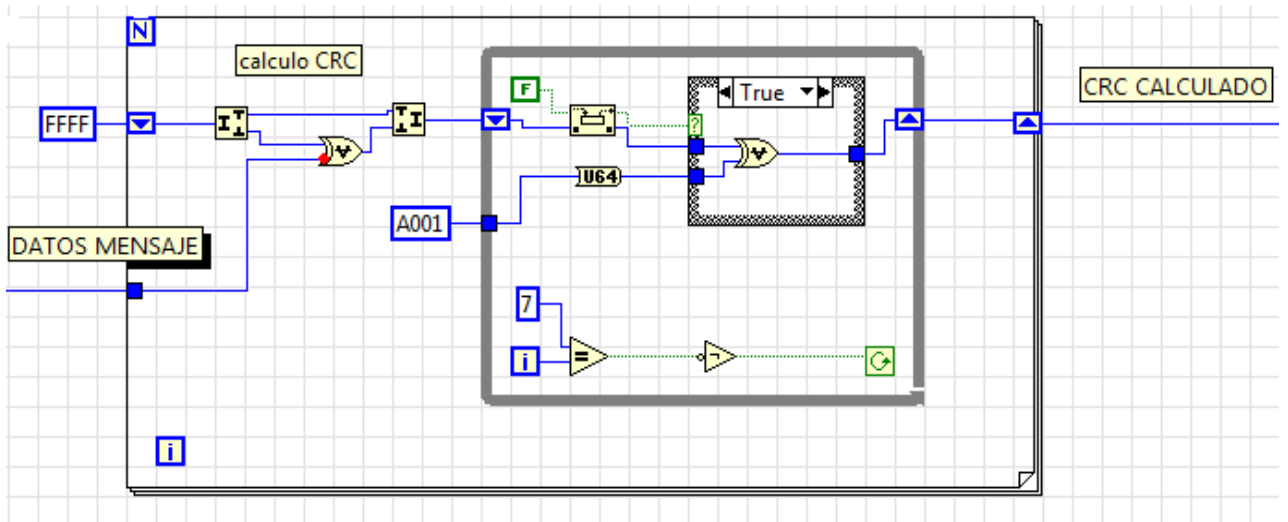


Figura 34. Programa para calcular el CRC en Labview.

4.- Este proceso debe repetirse hasta que todos los datos del mensaje hayan pasado por el proceso. Una vez que ha ocurrido esto el CRC calculado pasará al campo de chequeo y será posible enviar el mensaje.

FIN: este campo es similar al del inicio, se trata de un tiempo de silencio cuya duración depende del tiempo de transmisión de un carácter. Tiene la misma duración que el campo de inicio, 3.5 veces el tiempo de transmisión de un carácter.

1.7.2.6.-FUNCIONES Y REGISTROS MODBUS:

Existen cuatro tipos diferentes de datos a los que llamaremos registros que manejamos en Modbus, sobre los cuales escribiremos otro valor o leeremos el que tengan dependiendo de las necesidades:

- Discrete Inputs - entradas tipo bit.
- Coils (Outputs) - salidas tipo bit.
- Input Registers (Input Data) – entradas tipo palabra (2 bytes).
- Holding Registers (Output Data) – salidas tipo palabra (2 bytes).

Para comunicar con el dispositivo debemos utilizar las diferentes funciones que alberga el protocolo Modbus. Cada una de ellas tiene un propósito concreto, albergando algunas el objetivo de leer el valor de un registro y otras el de modificarlo. A continuación se pueden ver las funciones más utilizadas con el código que les hace referencia:

CODE	FUNCTION
01(01H)	READ COIL
03(03H)	READ HOLDING REGISTERS
04(04H)	READ INPUT REGISTERS
05(05H)	WRITE SINGLE COIL
06(06H)	WRITE SINGLE REGISTER
15(0FH)	WRITE SINGLE COIL
16(10H)	WRITE MULTIPLE REGISTERS
17(11H)	REPORT SLAVE ID

Figura 35.Funciones más comunes utilizadas del protocolo MODBUS

Del amplio rango de funciones han sido seleccionadas varias de ellas que son las utilizadas para acceder a los registros de nuestro dispositivo. Para el proceso de transmisión o escritura en el dispositivo se ha elegido la función “write single register” y “write single coil”, y para el de lectura o recepción de los datos “read holding registers” y “read coil”.A continuación se expone un ejemplo de uso de dichas funciones:

WRITE SINGLE REGISTER :

Esta función está diseñada para escribir en un registro de nuestro dispositivo el valor que se introduzca en el campo de información del mensaje. A continuación se delimita como debe ser la estructura del mensaje “query” (pregunta) que hace el maestro al dispositivo esclavo y el mensaje “response”(respuesta) que devuelve el esclavo para esta función.

PREGUNTA (MAESTRO)	RESPUESTA (ESCLAVO)
Dirección del esclavo	Dirección del esclavo
Función	Función
Dirección del registro(byte high)	Dirección del registro(byte high)
Dirección del registro(byte low)	Dirección del registro(byte low)
Valor para el registro (byte high)	Valor para el registro (byte high)
Valor para el registro (byte low)	Valor para el registro (byte low)
CRC-	CRC-
CRC+	CRC+

Ejemplo de mensaje para escribir en el registro 243 del dispositivo encargado de la configuración de la base de tiempos que utilizará el programa si HH:MM o MM:SS:

- dirección esclavo: 1 = 01 h (1 byte)
- código de función: 6 =06 h (1 byte)
- dirección registro: 243 = 00F3 h (2 byte)
- valor asignado al registro: 2 = 0002 h (2 byte)
- CRC (comprobación de error) : 38F8 h (2 byte)

PREGUNTA (MAESTRO)	VALORES	RESPUESTA (ESCLAVO)	VALORES
Dirección del esclavo	01H	Dirección del esclavo	01H
Función	06H	Función	06H
Dirección del registro(byte high)	00H	Dirección del registro(byte high)	00H
Dirección del registro(byte low)	F3H	Dirección del registro(byte low)	F3H
Valor para el registro (byte high)	00H	Valor para el registro (byte high)	00H
Valor para el registro (byte low)	02H	Valor para el registro (byte low)	02H
CRC-	38	CRC-	38
CRC+	F8	CRC+	F8

READ HOLDING REGISTERS :

Esta función está diseñada para leer en un registro de nuestro dispositivo el valor queeste tenga asignado o se haya escrito en él. A continuación se delimita como debe ser la estructura del mensaje “query” (pregunta) que hace el maestro al dispositivo esclavo y el mensaje “response”(respuesta) que devuelve el esclavo para esta función.

PREGUNTA (MAESTRO)	RESPUESTA (ESCLAVO)
Dirección del esclavo	Dirección del esclavo
Función	Función
Dirección del registro(byte high)	Campo Byte Count
Dirección del registro(byte low)	Dato 1(byte low)
Número de registros a leer(byte high)	Dato 1 (byte high)
Número de registros a leer (byte low)	Dato 2 (byte low)
CRC-	Dato 2 (byte high)
CRC+	etc...
	CRC-
	CRC+

Ejemplo de mensaje para leer el registro 253 del dispositivo encargado de la configuración del número de pasos que tendrá el programa que ejecutará el horno:

- dirección de esclavo: 1 = 01h (1 byte)
- código de función: 3 = 03 h (1 byte)
- dirección del registro: 253 = 00FD h (2 bytes)
- número de registros a leer: 1 = 0001h (2 bytes)
- CRC (comprobación de error) : FA15 h (2 bytes)

PREGUNTA (MAESTRO)	VALORES	RESPUESTA (ESCLAVO)	VALORES
Direccion del esclavo	01H	Direccion del esclavo	01H
Función	03H	Función	03H
Direccion del registro(byte high)	00H	Campo Byte Count	02H
Direccion del registro(byte low)	FDH	Dato 1(byte low)	00H
Número de registros a leer(byte high)	00H	Dato 1 (byte high)	01H
Número de registros a leer (byte low)	01H	CRC-	84
CRC-	FA	CRC+	79
CRC+	15		

1.7.2.7.-IMPLEMENTACION DE LAS FUNCIONES EN LABVIEW:

Para utilizar las funciones de Modbus explicadas anteriormente en el software Labview se descargó una librería desarrollada por National Instruments.

Esta librería no permitía la modificación ni la visualización del funcionamiento de sus subprogramas cosa necesaria para el desarrollo de nuestro programa, por ello se desarrollaron diferentes subprogramas casi idénticos o muy similares.

A continuación se muestra un programa básico de comunicación con el protocolo Modbus desarrollado para realizar pruebas de lectura y escritura de registros en nuestro dispositivo (figura 36). Para iniciar la comunicación es necesario inicializar el puerto serie, lo cual se consigue con la VI “Init” la cual delimita las características de la comunicación, parámetros serie, velocidad de transmisión, puerto utilizado, etc... Las VI’s señaladas con una flecha roja son los subprogramas diseñados para las funciones de Modbus en el programa que serán explicadas posteriormente en la parte de software del proyecto.

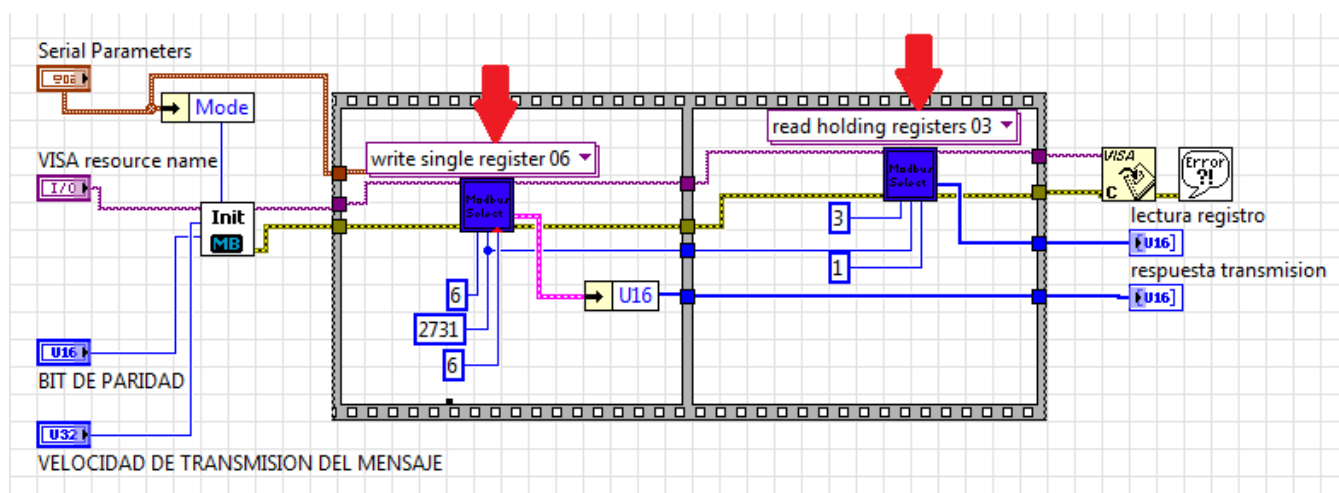


Figura 36. Programa básico de comunicación serie con Modbus.

Para la función “write single register” se debe delimitar la dirección del registro a escribir (en este caso 2731) y el dato a escribir (6). Para la función “read holding registers” se delimita la dirección también la cual será la misma que antes y el número de datos que se quiere leer (1 en este caso).

Este programa ha sido probado con éxito y es en el que está fundamentada la comunicación del software del programa principal del proyecto que esta explicado a continuación.

1.7.3.-PROGRAMACIÓN DEL SOFTWARE



1.7.3.1.-INTRODUCCION A LABVIEW:

LABVIEW es un software de programación gráfica para procesos que incluyan adquisición, control, análisis y presentación de información. Este programa utiliza lenguaje gráfico para implementar los códigos del mismo.

Los programas implementados con este software reciben el nombre de Instrumentos Virtuales o VI's. Dicho nombre proviene del ámbito de control de instrumentos, aunque actualmente se ha expandido y ya no solo recibe ese nombre el control de electrónica sino también su programación embebida.

El rasgo principal es la facilidad y sencillez de uso, es posible utilizarlo tanto por programadores profesionales como por personas con pocas nociones de programación. Permite realizar programas relativamente complicados que en el caso de utilizar lenguajes tradicionales sería de una gran dificultad para ellos. Además permite una gran ligereza a la hora de realizar un programa por lo que también es utilizado por programadores experimentados.

Uno de sus principales objetivos es aligerar el tiempo de desarrollo de aplicaciones y favorecer la entrada a la informática de profesionales de otros campos. Este software está diseñado para poder utilizarlo con cualquier otro software y hardware, independientemente del fabricante.

LABVIEW permite crear programas de multitud de VI's relacionadas entre sí, lo que correspondería a un gran número de páginas de código texto. Esto es muy útil para crear programas complejos con muchos puntos de entrada y salida, combinar VI's entre sí, etc.

Además LABVIEW tiene un gran número de librerías y subprogramas. A parte de las funciones típicas de todos los lenguajes de programación, incluye librerías para la adquisición de datos, control de instrumentación, GPIB y comunicación serie, análisis, presentación y guardado de datos.

Todos los programas creados o VI's tienen dos ventanas diferentes. Un diagrama de bloques y un panel frontal. Dentro de estas ventanas hay paletas que contienen las opciones para crear y modificar los VI's.

PANEL FRONTAL:

Se podría denominar como la pantalla de control del programa, es la interfaz gráfica que utiliza el usuario para interactuar con el programa. Dentro de ella se introducen la información que manejará el programa, y mostrará las salidas del programa. Suele estar formado por una serie de botones, pulsadores, potenciómetros, gráficos, etc...

Suelen estar definidos como controles o indicadores, dependiendo de su función. Los primeros sirven para introducir los parámetros al programa, mientras que los segundos muestran los resultados generados por el VI independientemente de si son datos adquiridos o resultados de alguna operación.

DIAGRAMA DE BLOQUES:

El diagrama de bloques constituye el código fuente del VI. En él es donde se realiza la programación del VI para realizar los procesos concretos utilizando las entradas y salidas del panel frontal.

Alberga en la paleta las funciones y librerías que tiene LABVIEW. En el lenguaje G (gráfico) las funciones y las estructuras son nodos elementales. Equiparables a funciones y estructuras de los lenguajes convencionales de programación.

1.7.3.2.-REALIZACION DEL PROGRAMA:

El archivo principal del proyecto es el llamado PROYECTO.VI (figura 36), una vez abierto nos permite configurar y ejecutar los programas adecuados para la rampa de temperatura/tiempo que el usuario desee ejecutar; a partir de los subprogramas que posteriormente se enumeran y más adelante serán explicados. Los subtareas son:

- **TRATAMIENTO DE DATOS:**

- Introducción de datos, **CARGAR** o **CREAR RAMPA:**

- Se introducen los datos a partir de un archivo .txt o son introducidos directamente por teclado en un control adecuado para ello dentro del programa.

-Guardado rampa, **GUARDAR RAMPA:**

Crea un archivo .txt con los datos ejecutados independientemente de si han sido introducidos por archivo o por teclado.

• **COMUNICACIÓN Y EJECUCION DEL PROCESO:**

-CONFIGURACIÓN COMUNICACIÓN con el horno:

Esta subtarea se encarga de configurar las características de la comunicación. Se elige la velocidad, puerto de salida, etc.

-Ejecución de rampa, **EJECUTAR RAMPA:**

Permite seleccionar el tipo de configuración para la ejecución de los datos introducidos en diferentes programas y pasos, además visualiza la ejecución de los datos en una gráfica en tiempo real.

Como se puede observar en la pantalla principal del programa (figura 37) existen diferentes recuadros, que nos permitirán seleccionar la configuración del programa.

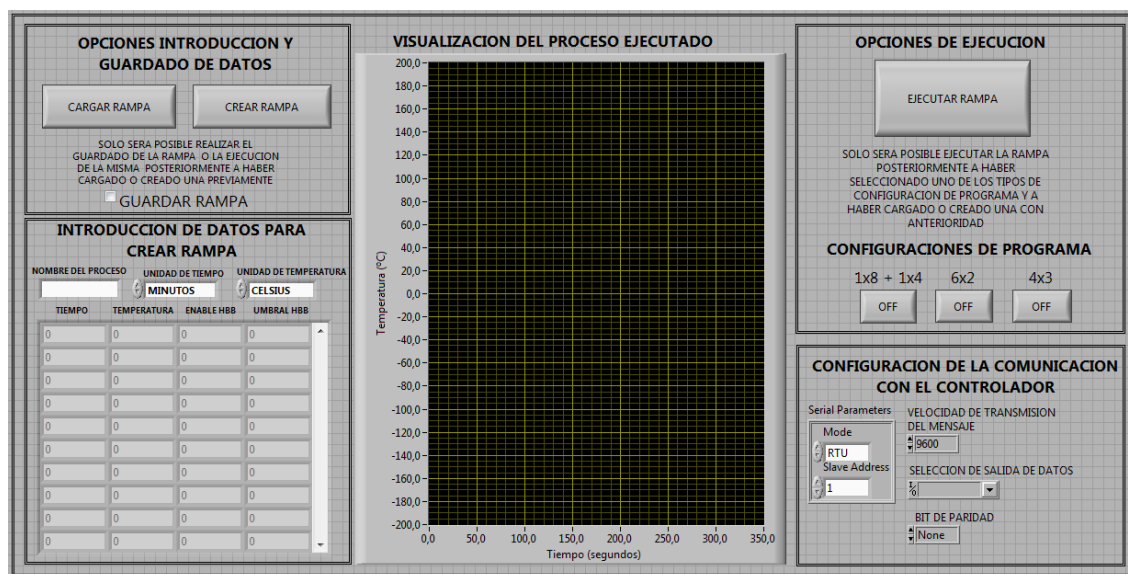


Figura 37. Pantalla inicial del programa.

Lo primero que debemos hacer para realizar un proceso con este programa es decidir previamente si queremos ejecutar y guardar la rampa que introduzcamos, ya que es necesario delimitar estas funciones antes de introducir los datos.

Esto se debe a que una vez iniciada la introducción de los datos el programa realizará la ejecución y guardado automáticamente si así lo hemos estipulado en la pantalla inicial.

El programa está estructurado con una Flat Sequence Structure la cual implica que se ejecuten de izquierda a derecha los frames que formen parte de esta estructura.

Para iniciar el programa, una vez que hemos pulsado “run” en labview el programa no comienza su ejecución hasta que no seleccionamos uno de los pulsadores “cargar rampa” o “crear rampa”. Los cuales activan una estructura de eventos dentro de un bucle while que se puede ver en la figura 39. Una vez terminado el bucle while, se dará paso al segundo frame de la Flat Sequence Structure que engloba todo el desarrollo del programa (recuadrado en verde figura 38).

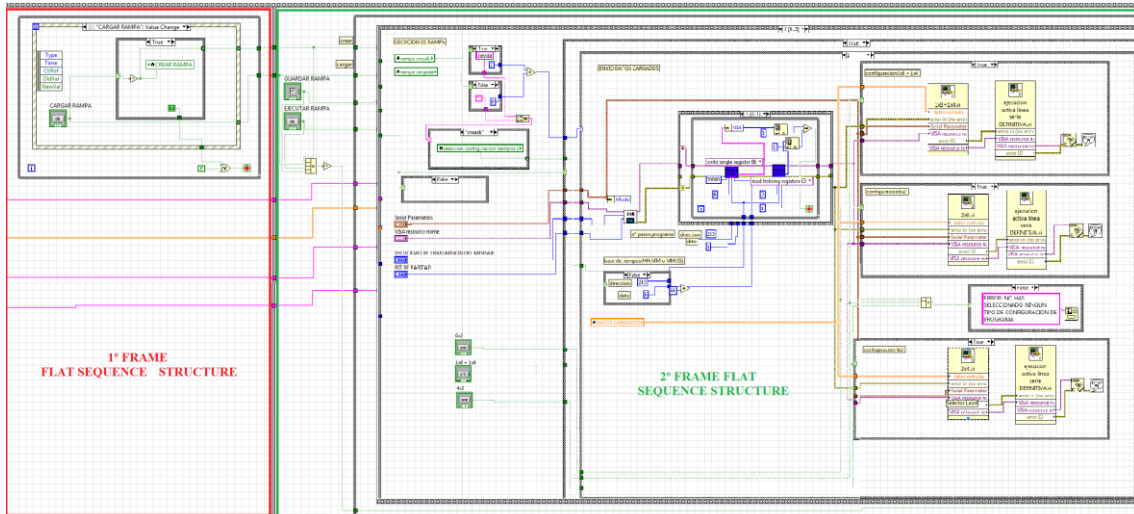


Figura 38. Imagen programa principal Flat Sequence Structure

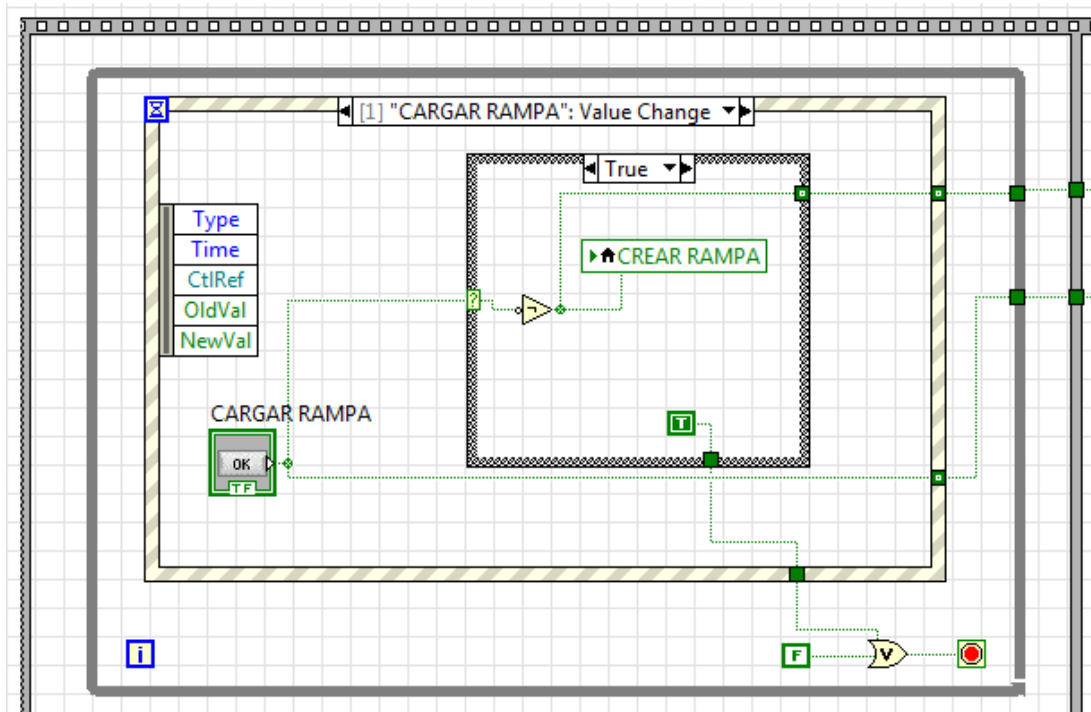


Figura 39. Event Structure dentro del primer frame de la Flat Sequence Structure.

En el segundo frame (recuadrado en verde figura 38) de la Flat Sequence Structure se encuentra un bucle “while” que alberga una Stacked Sequence Structure.

Esta estructura consiste en uno o varios subprogramas que se irán ejecutando secuencialmente dependiendo del orden en el que los hayamos introducido.

En la de nuestro programa tenemos 4 frames:

- FRAME 0: contiene el subprograma “cargar rampa”.
- FRAME 1: contiene el subprograma “crear rampa”.
- FRAME 2: contiene el subprograma “ejecución de rampa”.
- FRAME 3: contiene el subprograma “guardado de rampa”.

Estos frames se explican en apartados posteriores.

1.7.3.2.1.-INTRODUCCION DE DATOS:

Esta función se puede realizar de dos modos diferentes en este programa, cabe la posibilidad de cargar un archivo de texto (.txt) o introducir por teclado los datos en la tabla de la pantalla inicial.

Estos modos se pueden seleccionar a través de los pulsadores para ello habilitados, los cuales activarán el frame 0 o 1 dependiendo el que sea elegido.

1.7.3.2.1.1.- FRAME 0 – CARGAR RAMPAS

Este subprograma será el primero en ejecutarse en caso de seleccionarlo y tiene la función de importar un archivo .txt del PC y adquirir los datos del mismo para su uso posterior.

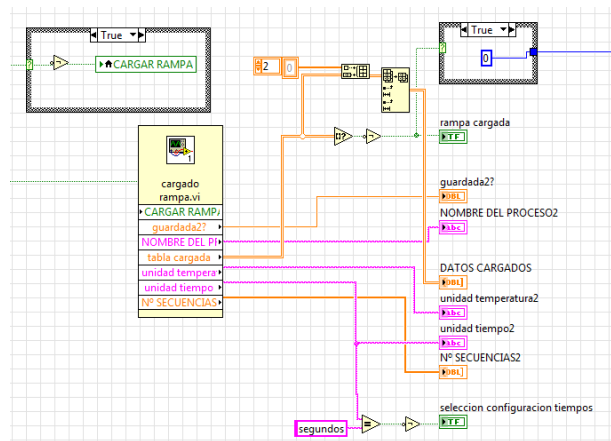


Figura 40. Contenido de frame 0 “cargar rampa”.

En este frame hay creada una subvi (subprograma) llamada “cargado de rampa” la cual es la que realiza el proceso de adquisición de datos a partir del archivo (figura 40).

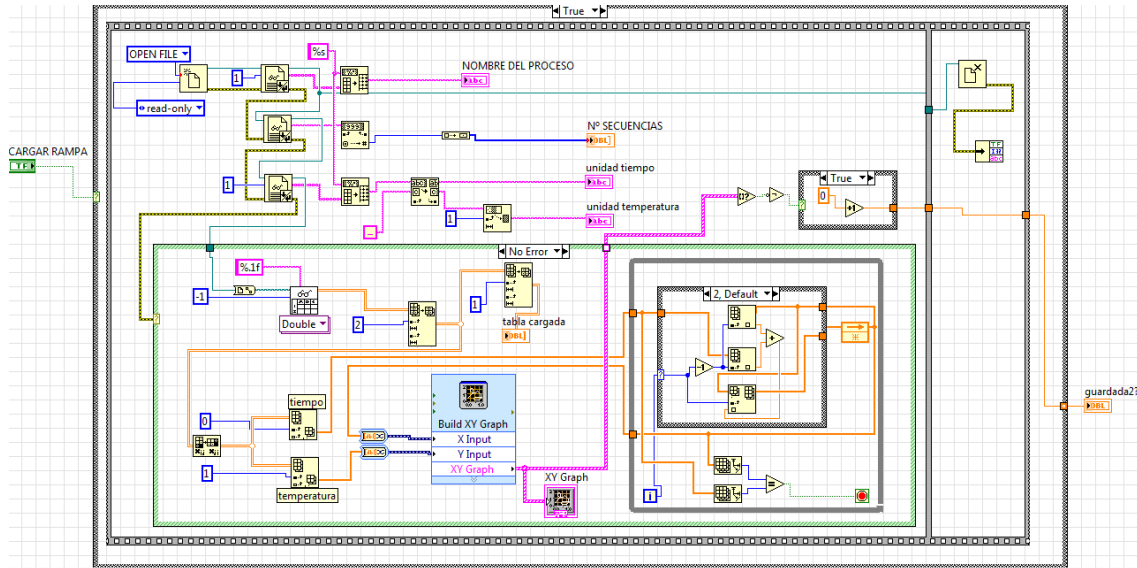




Figura 41. Subvi “cargado de rampa”

Este subprograma está contenido en otra Flat Sequence Structure (figura 41), se inicia con la función “open file”  la cual nos permite como su nombre indica abrir un archivo del disco duro del pc.

Posteriormente a través de una línea de dialogo se reciben los datos del archivo y se van extrayendo según interesa con la función “read from text file”  las líneas o caracteres del archivo.

Para extraer los datos de la rampa del archivo se utiliza el bucle de la figura 42.

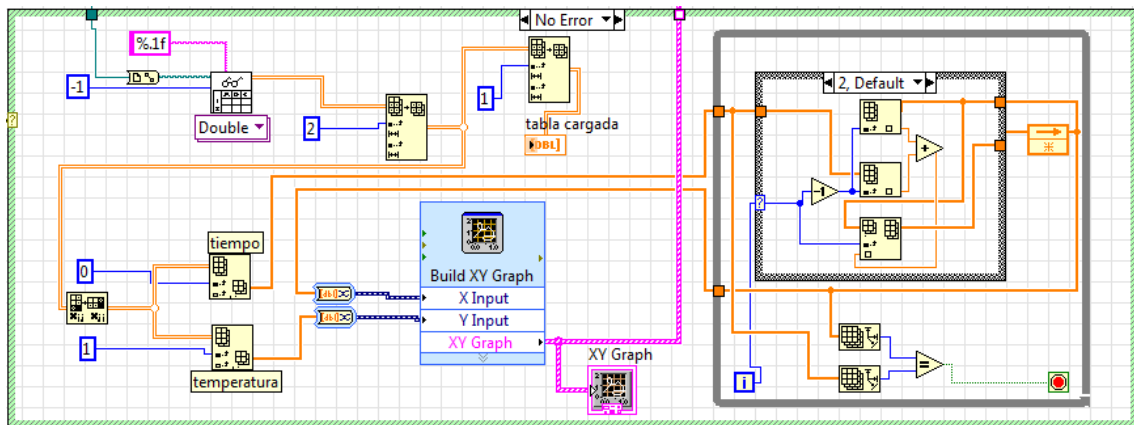



Figura 42.Extracción array de datos del archivo.

Una vez que se han extraído los datos y se han guardado en una variable (tabla cargada), el subprograma mostrará en el front panel de la subvi el proceso cargado. Esta función es desarrollada por el bucle while de la parte derecha. Posteriormente se procederá a cerrar el archivo desde el que se han cargado en el segundo frame de la estructura. Esta acción la realiza a través de la función “close file” .

Una vez realizado este paso el subprograma finaliza y vuelve al programa inicial,el cual dará paso al siguiente frame.

El formato de archivo que se ha de cargar debe tener la siguiente estructura:

- nombre del proceso.
- número de secuencias.
- unidades de tiempo y temperatura.
- datos de tiempo(1ª columna), temperatura (2ª columna), habilitación de la función Hold Back Band y umbral de la misma (3ª y 4ª columna respectivamente). (*)

(*)Los valores referentes a la función Hold Back Band deben ser introducidos en la primera fila de cada paso para que queden delimitados de manera adecuada. El uso de la función Hold Back Band es explicado en la parte de “ejecución y visualización del proceso”.

Un ejemplo de archivo se puede ver en la figura 43.

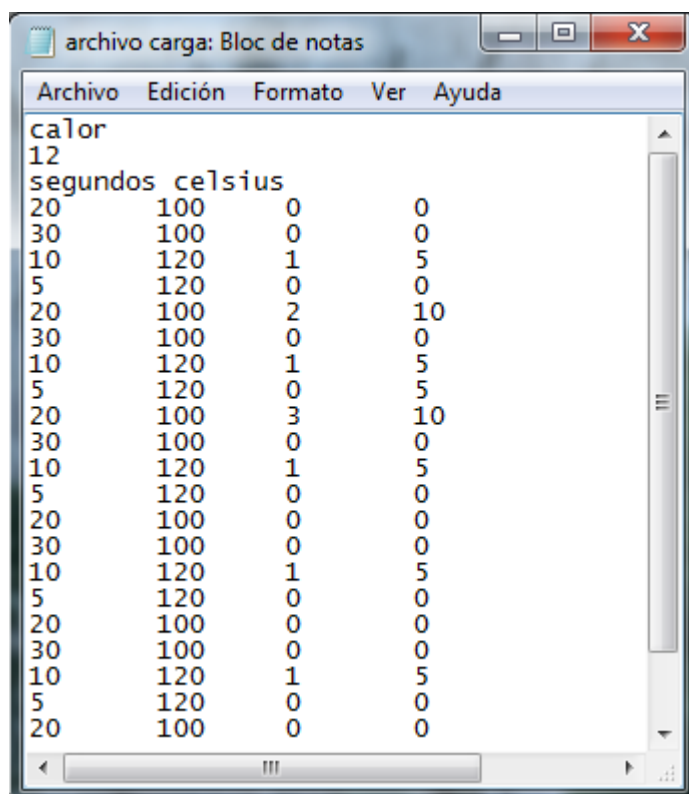


Figura 43.Ejemplo de archivo para cargar.

1.7.3.2.1.2.- FRAME 1 – CREAR RAMPA

Este subprograma es la segunda opción existente para introducir los datos de la rampa. Dentro de la Stacked Sequence Structure del programa es el frame1, y recibe el nombre de crear rampa (figura 44).

En este frame es necesario introducir los datos por teclado en la pantalla inicial del programa rellenando todos los campos existentes, entiéndase unidades de tiempo, temperatura, nombre del proceso y datos de la rampa.

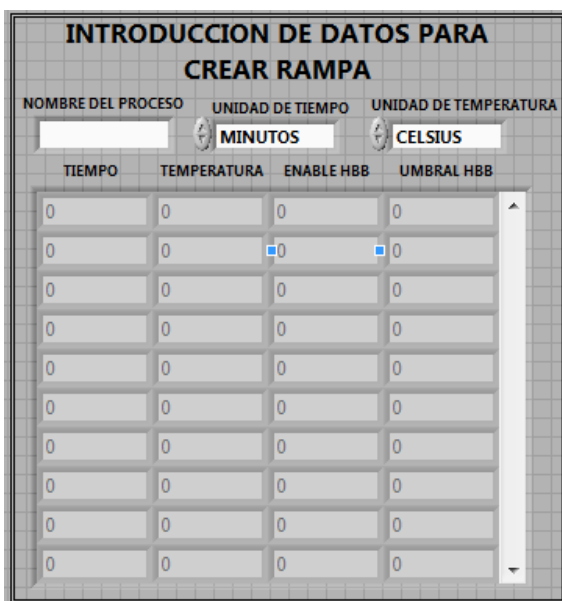


Figura 44. Panel para introducción de datos por teclado.

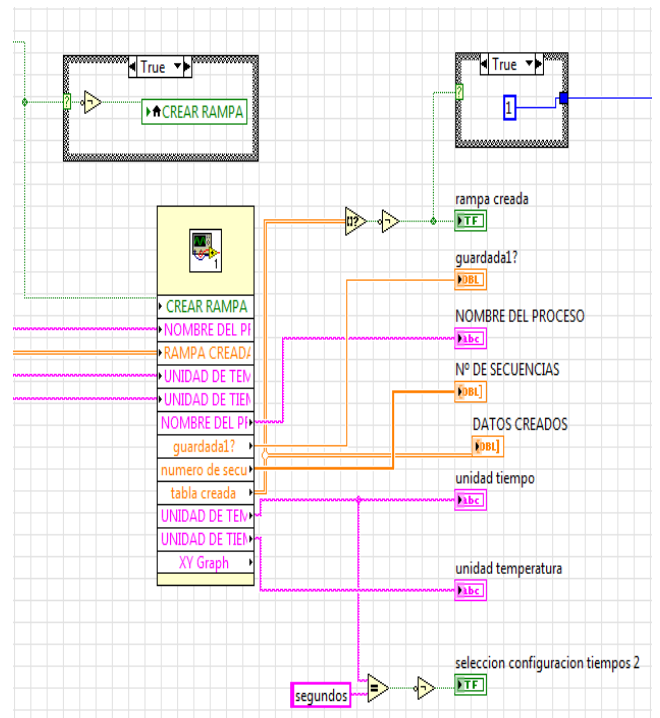


Figura 45. Contenido Frame 1 “Crear rampa”.

De la misma manera que en el frame 0 dentro de éste hay un subvi (Figura 45) que es el que desarrolla la función de adquirir los datos introducidos y almacenarlos en un array para su procesamiento posterior.

Este subprograma (figura 47) esta implementado con una estructura case la cual tiene dentro de ella otra estructura del mismo tipo. Al primer case el programa entrará con el pulsador de crear rampa. Pero para que entre al segundo case es necesario introducir todos los campos de unidades y nombre de proceso (si no se hace el subprograma no funcionará). En el caso de rellenar estos apartados pero no hacerlo en la tabla, el programa dará un mensaje de error por pantalla y parará (figura 46).

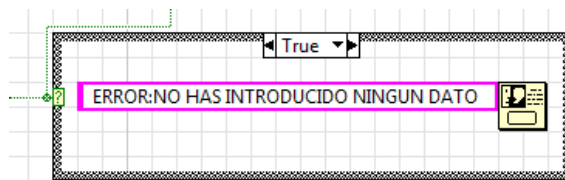


Figura 46 .Mensaje de error.

Una vez rellenados todos los campos el programa entrará al segundo case y almacenará estos datos en variables y la tabla en un array. Las variables en las cuales se almacenan los datos serán posteriormente utilizadas en el momento de ejecutar la rampa. También se mostrarán los datos en una gráfica de la subvi gracias a la estructura “while” de la segunda estructura “case”.

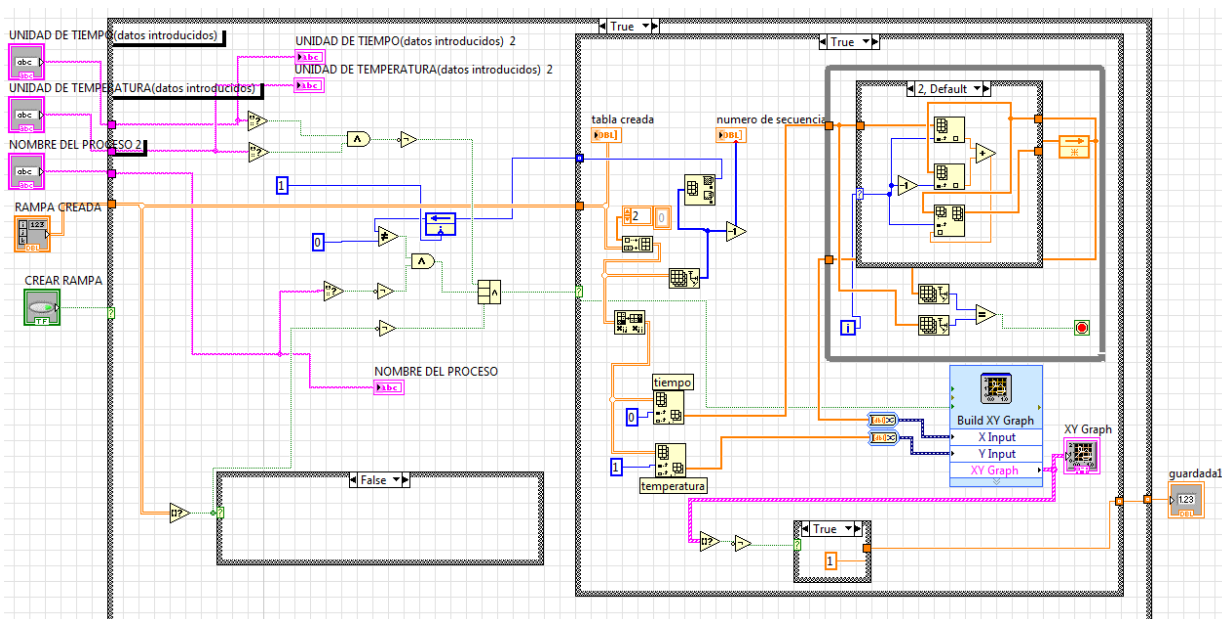


Figura 47. Subvi “crear rampa”.

1.7.3.2.2.-EJECUCION DEL PROCESO:

La ejecución del proceso esta implementada en el frame 2 de la Stacked Sequence Structure.

En este subprograma se utilizan y preparan los datos introducidos en los frames anteriores para que sean reconocidos por el regulador y este los ejecute de manera óptima.

Dentro de este frame se pueden distinguir dos partes bastante diferentes. Una de ellas es el procesado de datos del array y su posterior envío (este es el apartado de mayor relevancia en todo el programa).

La otra tiene la función de permitirnos ver la ejecución del proceso paso a paso en una gráfica a tiempo real, donde se irá visualizando todo el proceso ejecutado simultáneamente a su ejecución.

Los diferentes parámetros de la configuración del puerto y las características de la comunicación son comunes a ambas partes y se pueden observar en la figura 48 que hay a continuación.

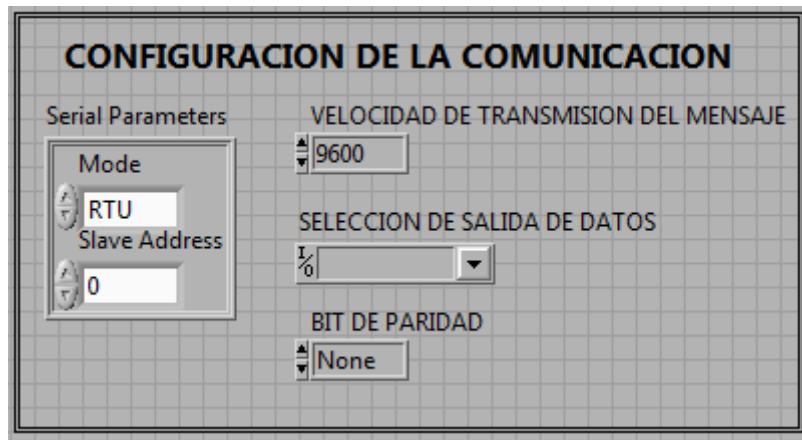


Figura 48. Parámetros de comunicación.

Los valores para estas variables tienen que ver directamente con la comunicación, en esta parte solo serán nombrados ya que posteriormente se explicará detalladamente la función de cada uno de ellos y los valores que deben recibir.

1.7.3.2.3.-PROCESADO - ENVÍO DE DATOS AL CONTROLADOR:

El envío de datos es una de las partes más importantes del programa, debido a que la correcta ejecución del proceso depende directamente de ello. Para que el controlador sea capaz de reconocer los datos introducidos estos deben ser preparados para enviarlos con el protocolo MODBUS serie RTU (explicado con anterioridad).

Es necesario configurar la comunicación rellenando de manera adecuada los campos de la figura 13, Serial Parameters (Mode = RTU y Slave Address = 1), Bit de paridad = None, Selección de salida de datos = COM1 y Velocidad de transmisión del mensaje = 9600.

Una vez rellenados estos datos al inicializar la comunicación el controlador será reconocido por el PC y esta será exitosa.

Cada dato de temperatura y tiempo ha de ser enviado a un registro diferente dentro del controlador, para que posteriormente cuando se esté ejecutando el proceso, la rampa siga los puntos adecuados.

También es necesario configurar varios registros más del controlador que delimitarán cuál es su modo de empleo.

Esto hace referencia al modo de programador del controlador, ya que puede ser controlado por línea serie (modo elegido), por teclado (el teclado del controlador), por las entradas digitales y por las alarmas habilitadas del controlador.

Todo esto queda mejor definido en el datasheet del GEFTRAN 800P adjuntado en los anexos del proyecto (anexo 4).

Labview aporta una librería con funciones previamente diseñadas para el uso de dicho protocolo dentro de su software.

Las VI's de esta librería han sido modificadas un poco para adecuarlas al código concreto de nuestro programa, debido a que daban algunos problemas en su ejecución. Además no permitían ver el proceso internamente en el modo “depuración” de Labview y por tanto tampoco dónde ocurrían los errores.

A continuación se muestra el frame de esta parte que ocupa el número 2 dentro de la Stacked Sequence Structure del programa principal (figura 49).

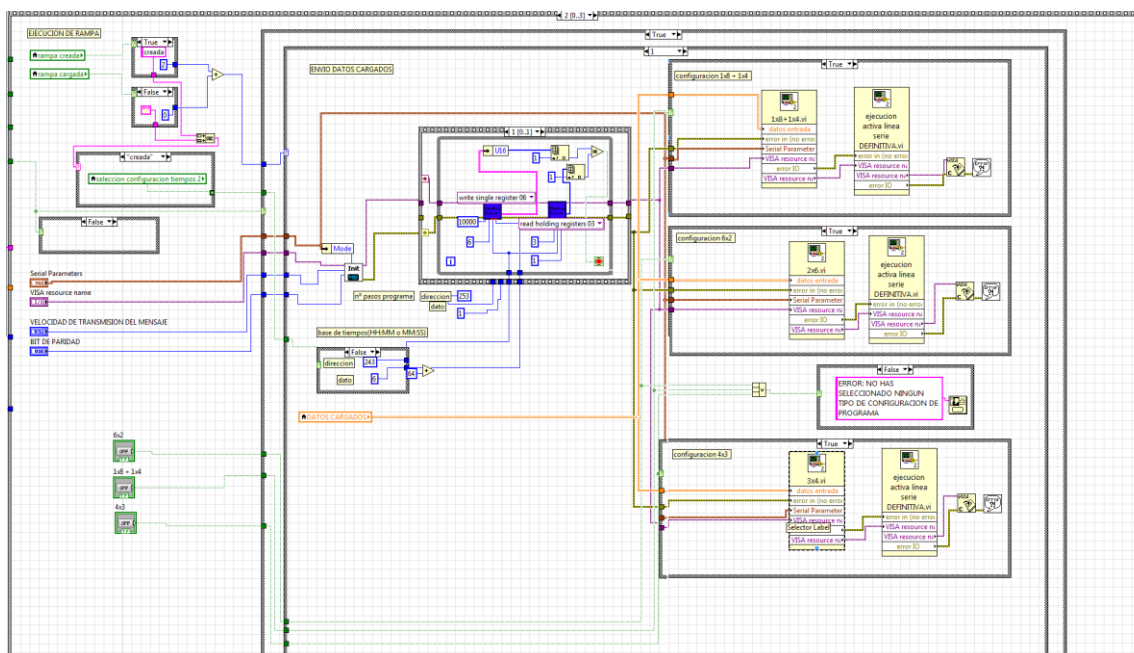


Figura 49. Frame 2 programa principal “Ejecución de rampa”.

Dentro de él hay una estructura Case que realizará uno de sus dos subprogramas (True o False) dependiendo de la procedencia de los datos, si han sido cargados o creados. La diferencia entre ellos es el array de datos que se carga.

Esta selección del subprograma a ejecutar depende de que se hayan creado o cargado los datos, que se haya seleccionado alguna de las configuraciones de programa y que haya sido pulsado el botón de “ejecutar rampa”.

Ambos subprogramas (cargar y crear rampa) son similares en cuanto a la comunicación por el puerto serie del PC. Ambos inician la comunicación con el controlador si hemos rellenado los campos de configuración previamente explicados.

Posteriormente se comienza con el envío de valores a los registros del controlador para configurar el funcionamiento de este.

Los primeros registros enviados hacen referencia al modo de funcionamiento del programador. Estos son:

-SP.Pt (dirección de registro 253) que define la habilitación del modo programador del controlador, el número de pasos posibles (admite hasta 12 pasos o 16 con linearización personalizada) y si existen grupos de regulación. Se le ha asignado un valor de 1 que deshabilita los grupos de regulación y lo configura como programador de 12 pasos.

-SP.Pr (dirección de registro 243) que define el tipo de programador desde donde va a ser controlado (teclado o entradas digitales) y la base de tiempos que se utilizará (HH:MM o MM:SS). El valor de este registro dependerá de la selección del usuario a la hora de introducir los datos de tiempo y de la función Hold Back Band.

-_S.S.t. (dirección de registro 236) que define desde donde se realizará el arranque y paro del programador. Se le ha asignado un valor de 21 que habilita a la línea serie para su control y le asigna un auto-reset del programa en ejecución si es detenido el proceso.

Una vez definido el modo de funcionamiento del modo programador, se procederá a realizar el envío de los valores a los diferentes registros para delimitar la rampa a seguir.

Se ha seleccionado un programador de 12 pasos los cuales son configurables en 3 modos diferentes, 1 programa de 8 pasos y otro de 4 pasos, 2 programas de 6 pasos y 3 programas de 4 pasos. Estas configuraciones se pueden seleccionar desde la pantalla inicial del programa principal, todas ellas funcionan de un modo similar. A continuación se explica una de las 3 VI's que se pueden ver en la imagen:

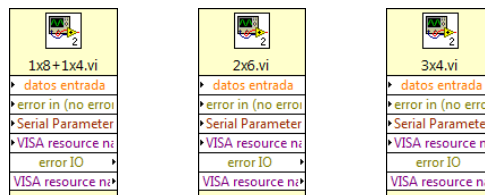


Figura 50. Imágenes de las diferentes subVI's para las diferentes configuraciones.

Se explica un ejemplo de la configuración 1x8+1x4. Esta VI está diseñada para dar valores a los registros de modo que se habiliten dos programas de 8 y 4 pasos.

Para ello se deben mandar a los siguientes registros los valores adecuados:

-LEn (dirección de registro $2731+4*N$, $0 \leq N \leq 3$ N=número de programa) delimita el número de pasos para el programa uno, 8 en este caso.

-P.tY (dirección de registro $2732+4*N$, $0 \leq N \leq 3$ N=número de programa) delimita el modo de reactivación del programa uno, el valor introducido será de 9. Al introducir este valor definimos que será desde condiciones de parada y a la espera de conmutación arranque y paro.

-SP.S (dirección de registro $1530+12*m$, $0 \leq m \leq 11$ m =número de paso) delimita el número de setpoint (paso) y el valor máximo al que llegará, ya que el mínimo será la variable de proceso (caso inicial del programa) o el valor del paso anterior. El valor de este registro dependerá de la rampa que ha sido introducida por teclado o archivo.

-rP.t (dirección de registro $1532+12*m$, $0 \leq m \leq 11$ m =número de paso) define el tiempo de rampa que tendrá el paso hasta alcanzar la temperatura delimitada en el registro anterior, este valor será en segundos o minutos dependiendo de la elección del usuario a la hora de introducir los datos en el archivo o por teclado. El valor de este registro dependerá de la rampa que ha sido introducida por teclado o archivo.

-So.t (dirección de registro $1534+12*m$, $0 \leq m \leq 11$ m =número de paso) define el tiempo de mantenimiento que tendrá el paso con la temperatura delimitada en el registro SP.S, este valor será en segundos o minutos dependiendo de la elección del usuario a la hora de introducir los datos en el archivo o por teclado. El valor de este registro dependerá de la rampa que ha sido introducida por teclado o archivo.

-Hbb (dirección de registro $1535+12*m$, $0 \leq m \leq 11$ m =número de paso) define el valor umbral para la función de mantenimiento Hold Back Band que se explica posteriormente. Será un valor entero introducido dependiendo de la necesidad del usuario en cada paso.

-St.Y (dirección de registro $1536+12*m$, $0 \leq m \leq 11$ m =número de paso) Este registro habilita la función Hold Back Band dependiendo de en qué segmento del paso se desee aplicar o en otro caso deshabilitarla.

Estos son los principales registros que definen un programa. Son implementados dependiendo de la configuración dentro de una Flat Sequence Structure que tendrá tantos frames como subprogramas tenga la configuración. En este caso serán 2 frames diferentes, uno de ellos para el subprograma de 8 pasos y el otro para el de 4 pasos.

Cada uno de estos frames albergará una Stacked Sequence Structure, cuyo número de frames dependerá de los pasos del programa. A continuación se adjunta una imagen del funcionamiento de esta subVI y el modo de adquisición de los datos del array origen y el envío de ellos al controlador a través de las funciones de la librería MODBUS.

Para enviar los datos en cada uno de estos frames existe otra Stacked Sequence Structure que escribirá los valores en los registros. Hay 5 registros que definen un paso (SP.S - rP.t - So.t - Hbb - St.Y) y por ello hay también 4 frames (Los parametros St.Y y Hbb hacen referencia a la misma función y por ello van en el mismo frame). En cada frame de ella se escribe y lee el mismo registro para comprobar que ha sido enviado el dato correcto.

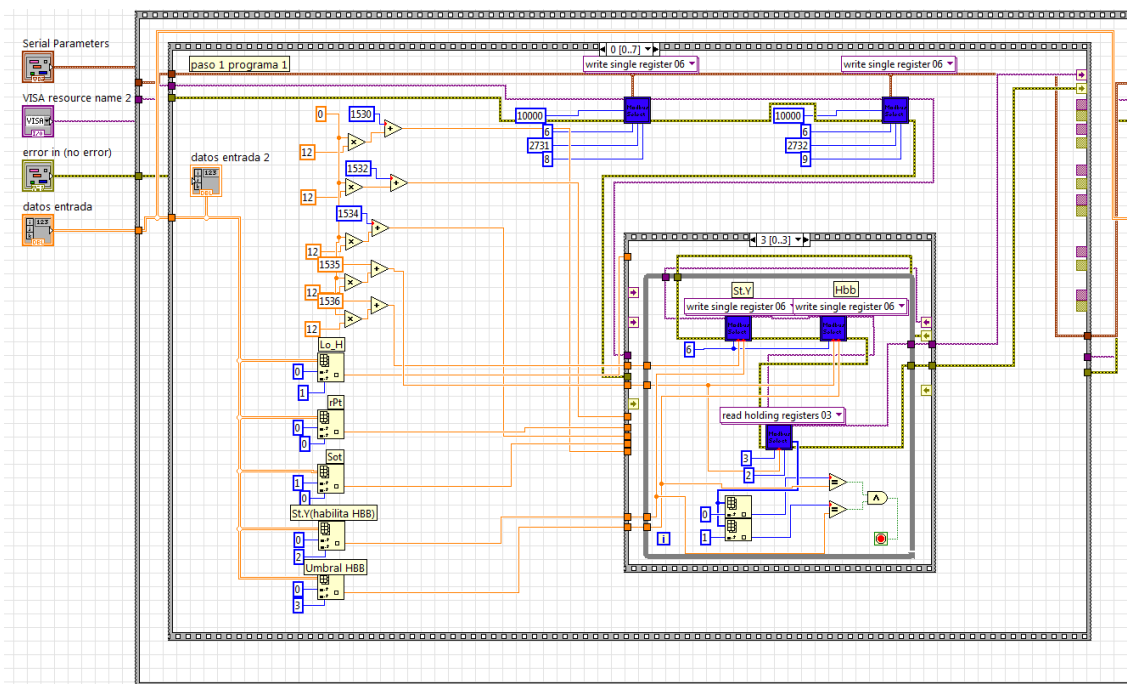


Figura 51. Primer frame SubVI 1x8+1x4.

El segundo frame de la Flat Sequence Structure de esta subVI albergará una Stacked Sequence Structure con 4 frames, es idéntico al mostrado en la figura 51 pero la Stacked Sequence Structure de su interior será de 4 frames en vez de 8. Una vez que se haya ejecutado esta subVI, el programa saldrá de ella con todos los datos de la rampa enviados a sus respectivos registros. Y pasará a la parte de visualización en tiempo real de la ejecución del proceso.

FUNCION HOLD BACK BAND:

El objetivo de esta función es parar el proceso en ejecución hasta que la temperatura real (variable de proceso) alcance el setpoint de paso delimitado en los registros. Es decir si esta función ha sido habilitada en el segmento rampa de un paso, en el momento que el setpoint llegue al valor máximo de rampa si la temperatura real (variable de proceso) no ha llegado a esa misma temperatura menos el umbral de HBB introducido, el proceso se parará hasta que llegue. Después, una vez haya llegado continuará con normalidad.

Esta función es muy útil debido a que en el proceso de calentamiento o enfriamiento la rampa introducida por el usuario a menudo alcanzará el valor de setpoint mucho antes que la rampa real. Es posible habilitar esta función en ambos segmentos de un paso (rampa y mantenimiento), en uno solo o en ninguno.

Para ello se introducirá el valor adecuado (0, 1, 2 ó 3) en la columna de “ENABLE HBB” o en el archivo introducido.

Se recomienda habilitar esta función en ambos segmentos (introducir un valor de 3) para que la rampa real sea lo más similar posible a la ideal.

Sty	HBB habilitado en rampa	HBB habilitado en mantenimiento	HBB referido a PV o a la entrada auxiliar
0			PV
1	x		PV
2		x	PV
3	x	x	PV
4			Entr. aux.
5	x		Entr. aux.
6		x	Entr. aux.
7	x	x	Entr. aux.

Figura 52. Valores para habilitar la función HBB.

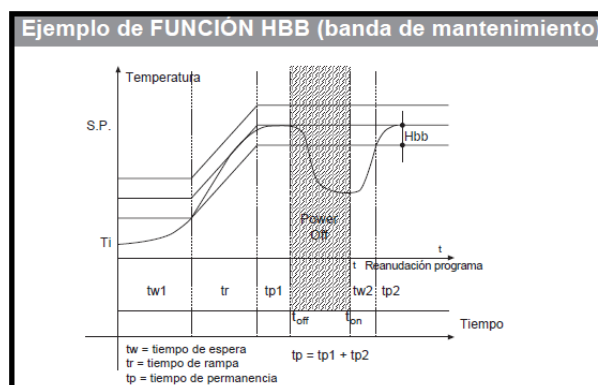


Figura 53. Ejemplo de función HBB.

Los valores de 4 a 7 no son utilizados debido a que para los procesos que se desarrollarán en principio no es necesaria una entrada auxiliar. Lo que no implica que si posteriormente se desea implementar procesos con una entrada auxiliar, esto no sea posible. A continuación se muestra un ejemplo del funcionamiento de HBB (figura 53).

1.7.3.2.4.- VISUALIZACION DE LA EJECUCION:

Esta parte de la ejecución del proceso mostrará una pantalla de control en cuanto se comience a ejecutar la subVI creada para esta función. Posteriormente a la carga de datos en el controlador por las subVI's de las configuraciones de programa se comienza a ejecutar esta función (figura 54)

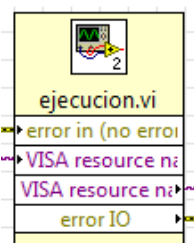


Figura 54. SubVI "ejecución".

Una vez iniciada se mostrará el panel frontal de la misma (figura 55). Esta subVI permite iniciar la ejecución del proceso cargado previamente en los registros del controlador con la configuración seleccionada.

Una vez iniciada esta VI se delimitan las entradas y salidas en la estructura "case" que serán habilitadas en el controlador desde el programa del PC con la primera iteración del bucle "while" al escribir los registros HRD1 y HRD2 (figura 56) en el frame 0.

Ambos reciben un valor de 15 lo que habilita todas las entradas y salidas, por si fuera necesario en un futuro.

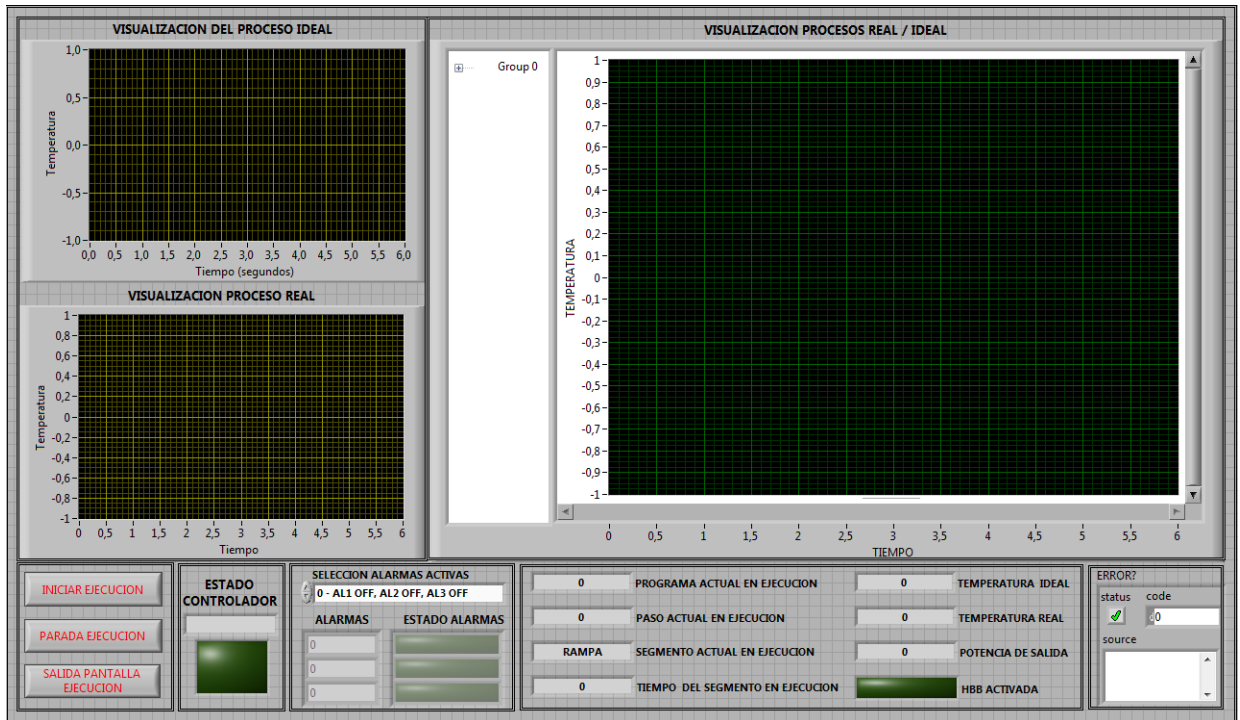


Figura 55. Panel frontal de la subVI “ejecución”.

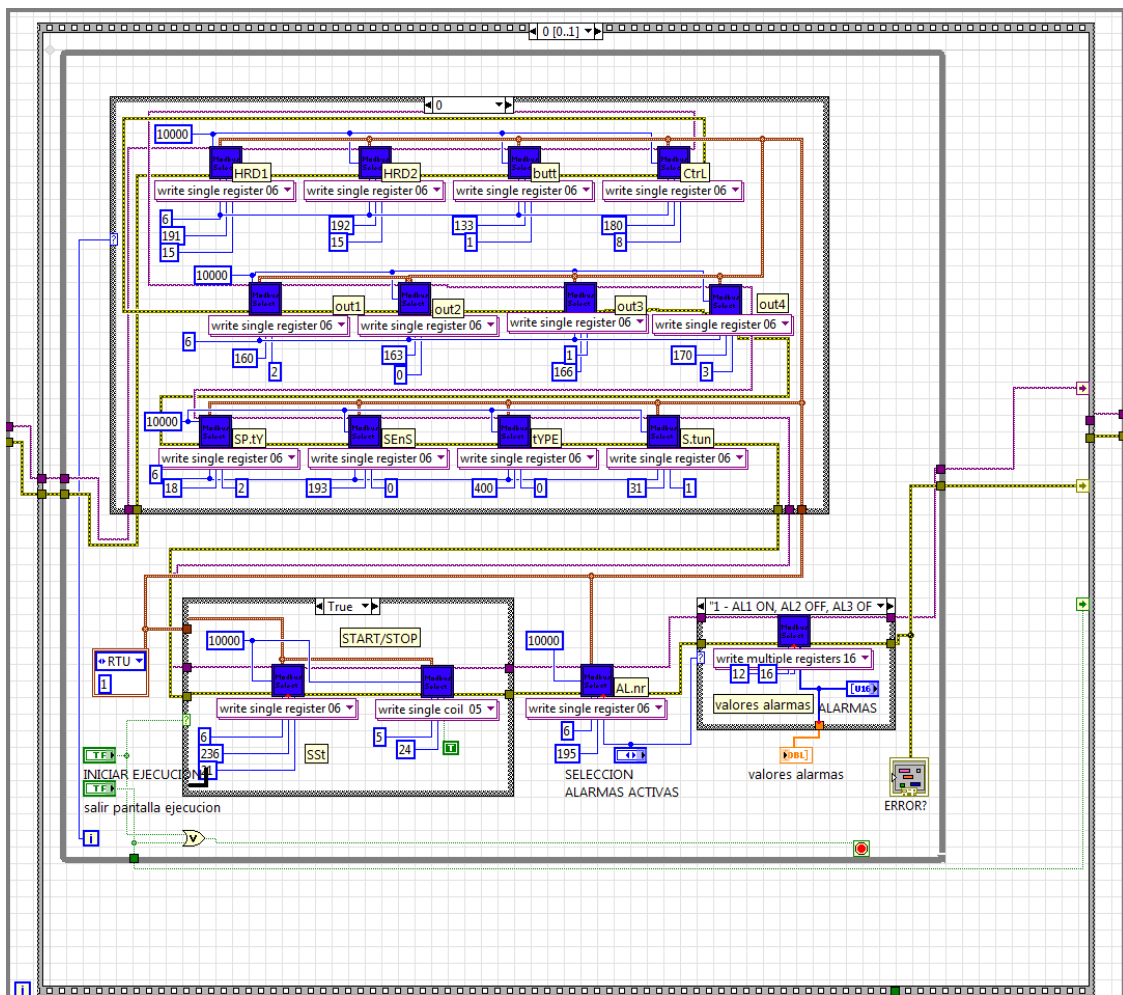


Figura 56. Frame 0 subVI “ejecución”.

La entrada principal se configura a través de los parámetros SP.tY, SenS y tYPE:


- SP.tY, (dirección de registro = 18) este registro permite la selección del tipo de setpoint (local o remoto). Se ha configurado como un set absoluto definido desde la línea serie al asignarle un valor de 2. que especifican el uso de un termopar tipo J (Fe-CuNi).
- SenS, (dirección de registro = 193) define el tipo de sensor utilizado para la entrada principal. En este caso se ha seleccionado un termopar al asignarle un valor de 0.
- tYPE, (dirección de registro = 400) una vez elegido el tipo de sensor, este parámetro delimita el tipo de sonda utilizado, en este caso, un tipo J (Fe-CuNi) al asignarle un valor de 0.


Las salidas se configuran al escribir los registros rL.o.1, rL.o.2, rL.o.3 y rL.o.4 (OUT1, OUT2, OUT3 y OUT4 respectivamente), delimitando cada uno de ellos para una función concreta. Estos registros están representados por los 4 leds del frontal del controlador:

- rL.o.1, (dirección de registro = 160), corresponde a la entrada 1 y al primer led de la pantalla del controlador. Hará referencia a la alarma AL1 estableciéndolo con un valor de 2.
- rL.o.2, (dirección de registro = 163), corresponde a la entrada 1 y al primer led de la pantalla del controlador. Hará referencia a la salida de control de calentamiento estableciéndolo con un valor de 0.
- rL.o.3, (dirección de registro = 166), corresponde a la entrada 1 y al primer led de la pantalla del controlador. Hará referencia a la salida de control de enfriamiento estableciéndolo con un valor de 1.
- rL.o.4, (dirección de registro = 170), corresponde a la entrada 1 y al primer led de la pantalla del controlador. Hará referencia a la alarma AL2 estableciéndolo con un valor de 3.

También se configura el tipo control automático necesaria para que el programa siga el proceso introducido. Para ello se configuran los registros Ctrl y S.tun:

- Ctrl, (dirección de registro = 180) que será llevado a cabo por un PID calor/frío al establecerle un valor de 8.
- S.tun, (dirección de registro = 31), habilita la función de auto-tuning(ver datasheet GEFran 800P anexo 3.5) que consiste en un reajuste de los valores de potencia para evitar oscilaciones y las desviaciones de la variable de proceso dándole un valor de 1.

Se configura el controlador por si se desea utilizarlo en modo manual o automático, por defecto funcionará de manera automática (ya que para que este software funcione así debe ser) pero en el caso que el usuario necesitare el modo manual solo es necesario pulsar el botón  del controlador y pasará a este modo. Esto se configura desde el registro butt.

- butt, (dirección de registro = 133), selecciona la función del botón  como selector de las funciones manual/automático del controlador. Para ello se le ha dado un valor de 1.

Para activar las alarmas existe una variable numérica en el panel frontal de la VI que permite habilitar las alarmas que el usuario desee, e introducir el valor umbral en el que se activarán. Todas ellas hacen referencia al valor de la temperatura real (variable de proceso). La alarma AL3 es una alarma introducida en el programa solamente ya que no hay más salidas en el controlador.

Una vez que se configuran los parámetros citados en el párrafo anterior la VI se mantendrá en el bucle “while” del frame 0 hasta que se pulse el botón “Iniciar ejecución” del panel frontal. En este momento se escribe en el controlador un registro que controla el arranque y paro del controlador para que inicie el proceso introducido. El registro es el S.s.t que se ocupa de la configuración de arranque y paro del modo programador del controlador, está configurado para la activación desde línea serie. Posteriormente se da a la bobina o “coil” número 24 un valor de uno para que inicie el arranque. En este momento el programa pasará al frame 1 (figura 57).

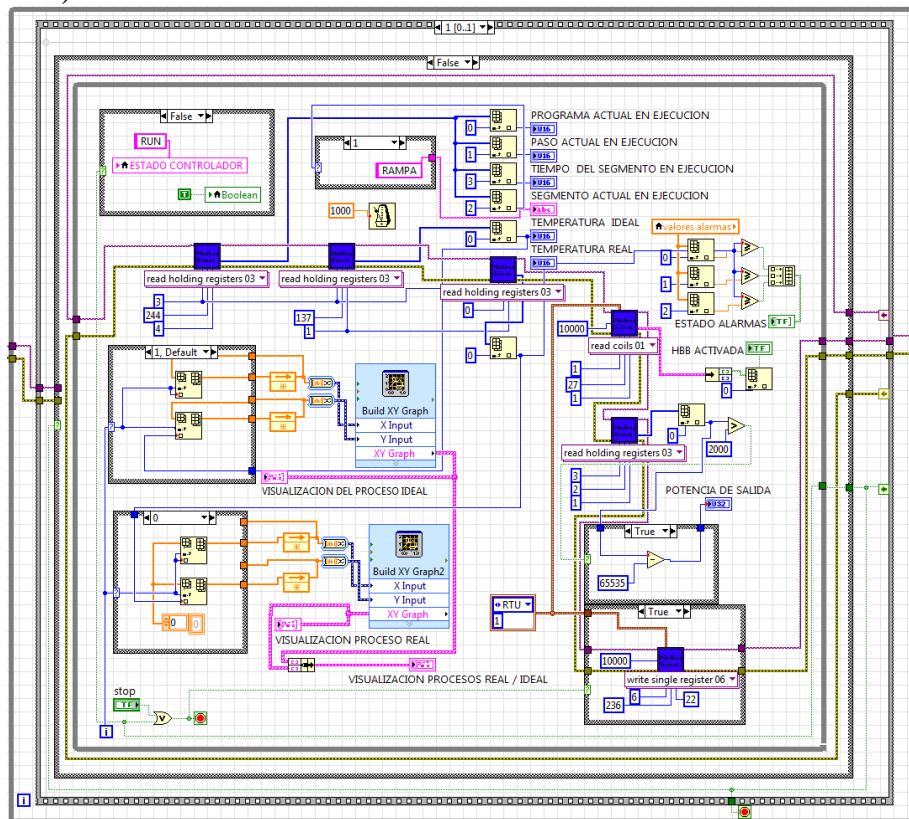


Figura 57. Frame 1 subVI “ejecución”.

El frame 1 de esta VI se encarga de adquirir datos en tiempo real de la ejecución del proceso y mostrarlos en 3 gráficas diferentes.

Dentro de este frame se utilizan 5 funciones MODBUS para la lectura de registros y bobinas:

- La primera se encarga de leer los datos del número de programa activo, el número de paso del programa, el segmento del paso y el valor del timer del segmento que se está ejecutando. Son los registros desde el 244 al 247 respectivamente, todos estos datos se muestran en los diferentes indicadores numéricos del panel frontal.
- La segunda función de lectura se encarga de adquirir los datos ideales de temperatura, que son aquellos que han sido introducidos previamente en los registros determinados. Es el registro 137, todos estos datos se almacenan en un array que se irá visualizando cada segundo. También se muestra en un indicador numérico el valor de la temperatura ideal en cada momento del proceso.
- La tercera función de lectura se encarga de adquirir los datos reales de temperatura. Es el registro 0, todos estos datos se almacenan en un array que se irá visualizando cada segundo. Esta función actúa de manera idéntica a la anterior variando únicamente en los datos adquiridos. También se muestra en un indicador numérico el valor de la temperatura real en cada momento del proceso.
- La cuarta función es la encargada de la lectura de la bobina referenciada a la función Hold Back Band que ha sido explicada previamente, la bobina es la número 27. Se ha implementado un indicador luminoso o led que se activará cuando esta función esté activa.
- La quinta función de adquisición de datos es la encargada de recopilar los datos de la potencia utilizada por el horno para incrementar o decrementar la temperatura. El registro encargado es el número 2, y los valores de salida pueden ser positivos y negativos. Serán negativos cuando la temperatura este en decrecimiento y positivos cuando este en incremento.

Todos estos datos se visualizan de una manera diferente dependiendo del tipo de variable a la que hagan referencia. Pero el más importante es la lectura de la temperatura real, que se visualizará simultáneamente a la temperatura del proceso introducido en una gráfica denominada “Visualización de proceso real/ideal”.

También existen 3 indicadores luminosos que hacen referencia a las 3 alarmas existentes en el programa. Estas se comparan con la temperatura real del proceso y se activan cuando la temperatura real sobrepasa el umbral delimitado para cada una de ellas.

Hay implementado también un apartado para el estado del controlador que indica si está parado o funcionando. Esto se muestra a través de un indicador led y un indicador de texto, este último mostrará un mensaje de “RUN” o “STOP” dependiendo del momento. Además se ha implementado un indicador de errores por si ocurriera algún problema con la comunicación, se pueda ver la razón del fallo.

Para controlar la ejecución de la VI y del controlador existen 3 pulsadores diferentes. Al iniciarse la VI será necesario pulsar el pulsador “INICIAR EJECUCION”, posteriormente para pararlo en el momento que se desee existe un pulsador denominado “PARADA EJECUCION” y una vez parado el proceso para volver al programa principal del proyecto será necesario pulsar “SALIDA PANTALLA EJECUCION”. Estos pulsadores controlan la condición de los bucles “while” y “case” que albergan todo el código del programa.

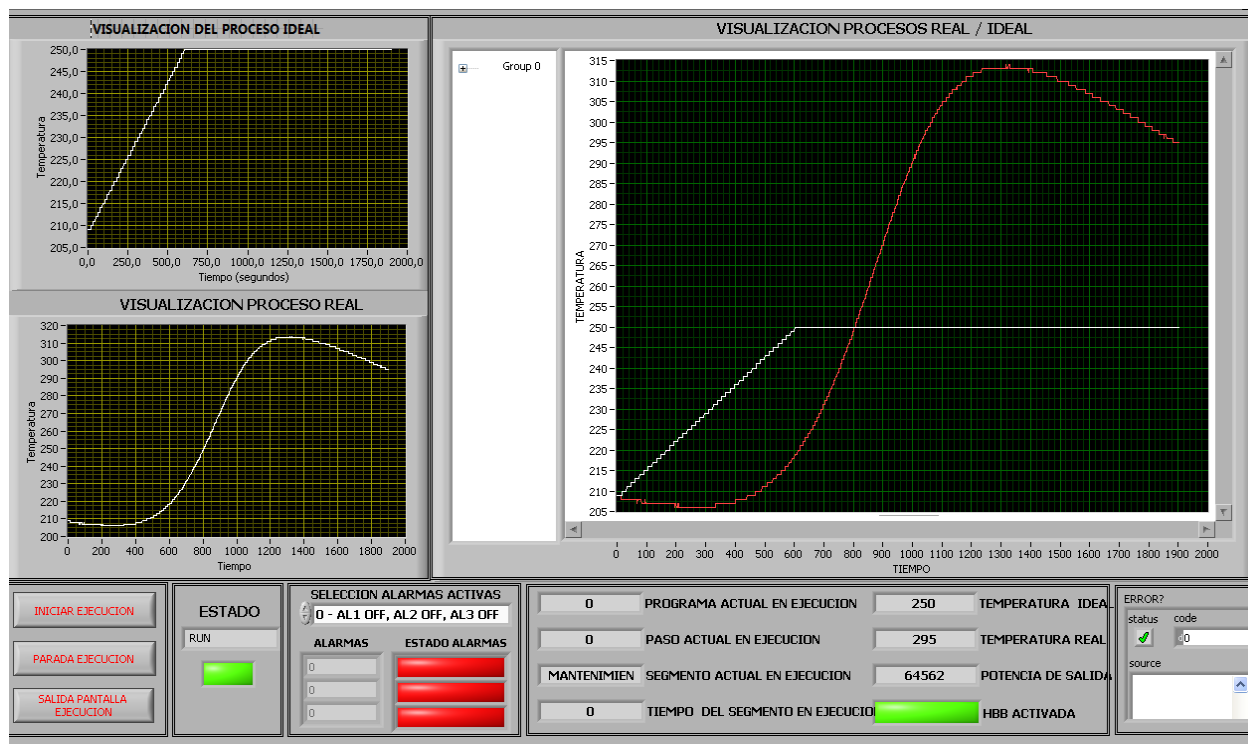


Figura 58. Proceso en ejecución.

1.7.3.3.- GUARDADO Y EXPORTACION DE DATOS:

FRAME 3 – GUARDAR RAMPA

Este subprograma tiene por objeto guardar el proceso ejecutado en los frames anteriores y posteriormente exportarlo a un archivo de texto para poder almacenar dicho proceso.

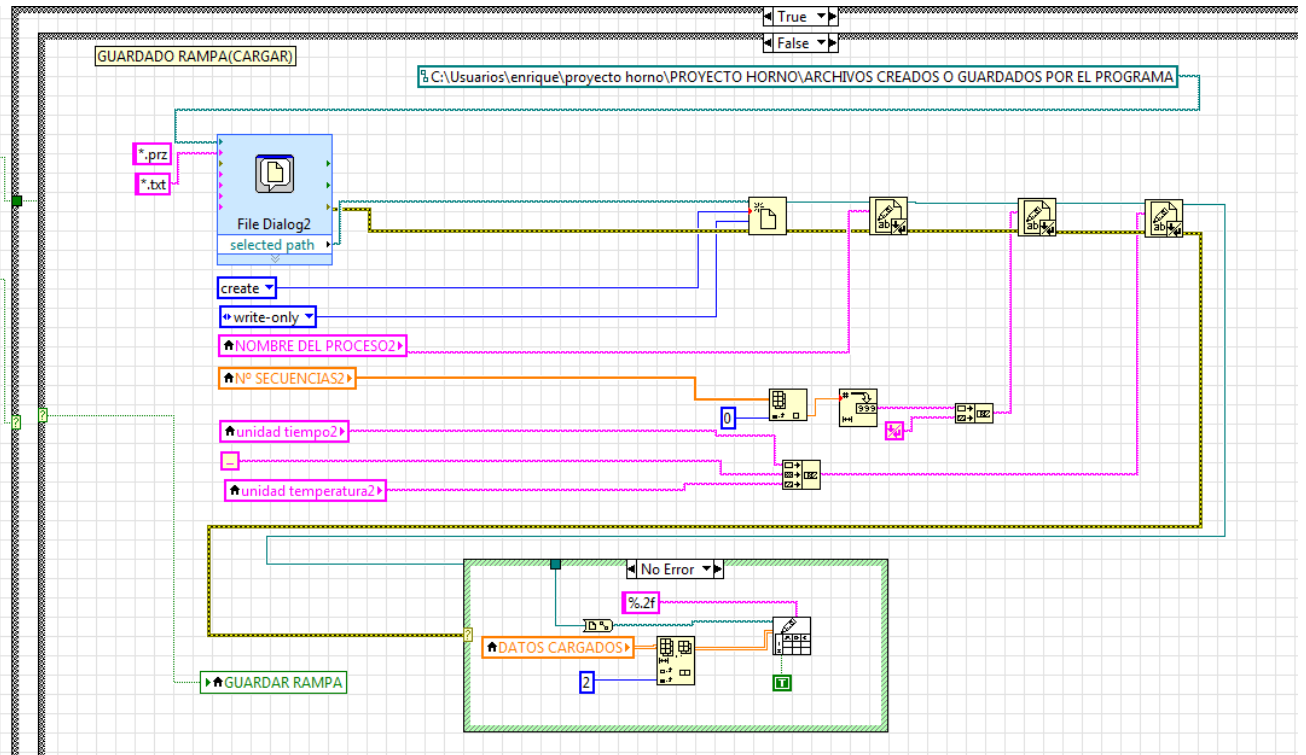


Figura 59. Guardado de rampa.

Dentro de este frame existe una estructura case la cual tiene implementada otra estructura case en su interior (figura 59). La primera de ellas sirve para que el programa detecte si ha sido pulsada la opción de guardar rampa y debe ejecutar la estructura. Para ello está programada la secuencia de puertas lógicas de la figura, que con las variables almacenadas previamente dependiendo del proceso actuará de una manera u otra.

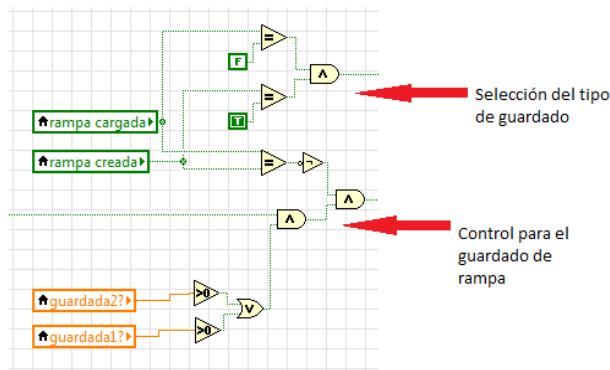


Figura 60. Selección de la función a ejecutar por la estructura.

En la figura anterior se puede observar que existen dos partes en el control de la estructura, la llamada “Control para el guardado de rampa” permite el funcionamiento del primer case. Para ello el checkbox de la pantalla inicial debe estar presionado y una de las variables “guardada 1?” o “guardada 2?” debe dar un valor de 1. Estas variables adquieren ese valor de 1 cuando se ejecuta el frame de introducción de datos en el que están implementadas. La variable “guardada 1?” corresponde a cargar rampa y la “guardada 2?” a crear rampa.

La segunda parte llamada “Selección del tipo de guardado”, está dirigida al control del case interno de la estructura. Dependiendo de si hemos introducido los datos por teclado o a través de un archivo realizará el caso “true” o “false” de la estructura. Ambos casos son idénticos, lo único que los diferencia son las variables de entrada que cada una viene de un tipo de cargado de datos diferente. Una vez explicado el control de la estructura es más sencillo entender el funcionamiento interno de la misma. Después de que se haya seleccionado el tipo de guardado, la estructura comienza abriendo un diálogo con una dirección dentro del disco duro del pc. En él pedirá la selección de un archivo de extensión “.txt” vacío en el cual realizará el guardado de los datos del proceso.

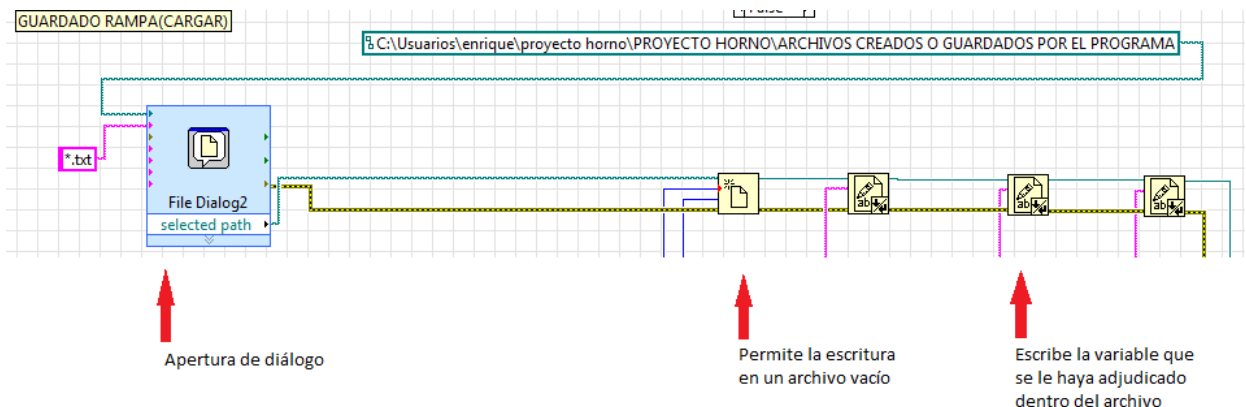


Figura 61.Inicio dialogo de guardado.

A continuación de esta apertura de diálogo (figura 60) , la línea de diálogo va a la función “create file” la cual permite la escritura en el archivo creado para el guardado.

Después la línea de diálogo sigue a una sucesión de funciones “write to text file” las cuales van escribiendo los valores de las variables en el orden concreto.

Para los datos de la rampa ejecutada es necesario una función diferente llamada “write to spreadsheet file” la cual escribe los datos de la rampa en el archivo.Una vez que esto ha sido realizado el guardado del proceso finaliza y también termina el programa principal.

1.8.-CONCLUSIONES:

En este proyecto fin de carrera se ha conseguido comunicar un PC con un controlador Gefran 800P que controla un horno Linn VMK39. Esta parte del proyecto se alargó debido a las complicaciones que se sufrieron a la hora de comunicar el PC con el controlador. Las complicaciones surgidas se han debido a la necesidad de realizar una conversión entre las interfaces de salida del puerto serie del PC (RS232) y la interfaz de entrada al controlador (RS485).

El principal problema de esta conversión está basado en el control de flujo de comunicación entre dispositivos, ya que los primeros circuitos adquiridos para este fin requerían un control manual del flujo de comunicación que complica en sobremanera la comunicación.

Este problema se solucionó al adquirir un circuito con control de flujo automático de la comunicación. Previamente fue necesario realizar diferentes pruebas con diferentes programas de comunicación.

El siguiente logro ha sido controlar un proceso en el horno Linn VMK39 gobernado desde el controlador Gefran 800P, el cual era controlado desde el PC. Esto se ha conseguido desarrollando un programa con el software Labview. Este permite la introducción de los datos de un proceso de calor/frío a selección del usuario, la ejecución de este proceso controlado desde el programa, la visualización en tiempo real del proceso y la exportación de los datos a un archivo.

En la parte de ejecución del proceso introducido, uno de los objetivos era conseguir que la rampa de temperatura real fuera lo más similar posible. Como es posible observar en la figura 62, la rampa real antes de estabilizarse a la temperatura del setpoint del paso sufre un sobreimpulso producido por la potencia de la corriente de regulación. Para solucionar esto se utiliza la función autotuning que realiza un control automático de los parámetros del PID.

Se ha implementado la función Hold Back Band por la necesidad de que la rampa ideal espere a que la variable de proceso alcance el valor de setpoint requerido.

Ha quedado pendiente el desarrollo de un subprograma cuya función sea la introducción de la pendiente de una rampa delimitando con ella los valores de tiempo y temperatura de cada paso. Además de la implementación de un selector del programa a ejecutar de los 2 o 3 posibles que ofrece el controlador.

1.9.- LINEAS FUTURAS DEL PROYECTO:

La automatización del controlador se ha conseguido de manera satisfactoria, pero se me ha pedido en última instancia el desarrollo de una función dirigida al control de las pendientes de las rampas de calentamiento y enfriamiento.

El objetivo es controlar los picos que se crean cuando sube y baja la temperatura en las rampas (Figura 62). No ha sido posible en el tiempo restante la implementación de esta función, pero lo investigado hasta el momento sugiere que la solución para ello está en los grupos de regulación existentes en el controlador.

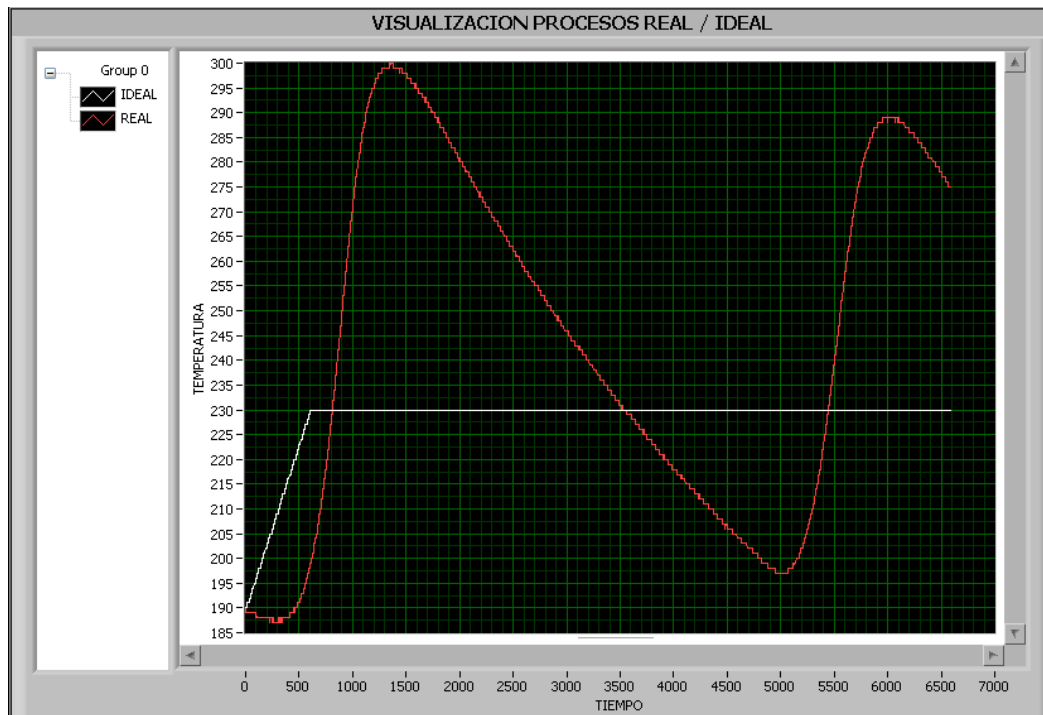


Figura 62. Estabilización de la T^a en una rampa de subida.

Para que sea posible utilizar estos grupos de regulación lo prioritario es configurar el registro SP.Pt (dirección de registro 253) con un valor de 2 lo que habilita el uso de estos grupos. Dentro de estos grupos de regulación cabe la posibilidad de delimitar la potencia utilizada en cada paso, limitar los valores máximos y mínimos de la potencia desarrollada durante el proceso de calentamiento o enfriamiento, así como los valores de tiempo derivativo e integral y las bandas proporcionales de enfriamiento y calentamiento. Estos valores son los que controla el PID en el modo autotuning.

Para los pasos de un programa es posible determinar el uso de uno de estos grupos de regulación a través de el registro GrP.x ($1541+12*m$, $0 \leq m \leq 11$ m=número de paso) la x hace referencia al paso del programa en cuestión.

La configuración de los grupos de regulación se realiza a través de los registros desde el 2930 hasta el 2975 como se indica en el anexo 5 (direcciones de modbus en el controlador). Existen 4 grupos diferentes de regulación aplicables a cuantos pasos se desee. Además en el anexo 4 (datasheet del controlador GEFAN 800p) viene una pequeña explicación del modo de funcionamiento de las regulaciones.



PRESUPUESTO

Departamento de Ingeniería Eléctrica y Electrónica

Alumno: Enrique Ruiz Blanco

Tutor: Ignacio Del Villar Fernández

2.-PRESUPUESTO:

Se ha realizado una estimación del presupuesto del presente Proyecto Fin de Carrera como si se tratase de un trabajo de investigación que se realiza dentro de una empresa. Para ello se ha de tener en cuenta varios apartados que se muestran a continuación.

2.1.- Materiales Laboratorio:

El horno Linn VMK39 y el controlador GEFTRAN 800P no se incluyen en el presupuesto porque fueron adquiridos por la universidad previamente a la propuesta de este proyecto.

	CANTIDAD	PRECIO UNIDAD (€)	COSTES ENVIO (€)	IMPORTE(€)
TARJETA RS232/RS485 amazon.es	1	5,32 €	2,34 €	7,66 €
TARJETA RS232/RS485 todoelectronica.es	1	8,14 €	3,10 €	11,24 €
TARJETA RS232/RS485 B&B ELECTRONICS	1	58,59 €	5 €	63,59 €
COMPONENTES CIRCUITO CONVERSION (FIGURA10)	-	10,56 €	-	10,56 €
TOTAL				93,05 €

2.2.- Software utilizado:

	CANTIDAD	PRECIO UNIDAD (€)	COSTES ENVIO (€)	IMPORTE(€)
Software Labview	1	1.551,00 €	17,83 €	1.568,63 €
Software GF_express	1	GRATUITO	-	-
Software Docklight	1	GRATUITO	-	-
Software Hyperterminal	1	GRATUITO	-	-
TOTAL				1.568,63 €

2.3.- Mano de obra:

En este apartado del presupuesto se tiene en cuenta el coste del personal encargado de realizar el proyecto.

Se considera el sueldo de dos personas, ya que el proyecto ha sido realizado por un ingeniero y una persona que ha supervisado el proyecto en todo momento. Se estima el sueldo de un futuro ingeniero y un responsable que dedica un 20% de su trabajo en tareas de asesoramiento y revisión. Se tendrá en cuenta los cargos sociales.

2.4.- Salario base:

	Meses	Sueldo / Mes (€)	Total (€)
Ingeniero Técnico Industrial, Especialidad Electricidad	6	1500	9000
Responsable Asesor	6	600	3600
Total			12600

2.5.-Cargos sociales:

	Porcentaje
Indemnización despido	3%
Seguros de accidente	5%
Subsidio familiar	3%
Subsidio vejez	5%
Abono días festivos	10%
Días de enfermedad	2%
Plus de cargas familiares	3%
Gratificación extraordinaria	10%
Otros conceptos	8%
TOTAL	49%

2.6.-Salarios efectivos:

Para el cálculo de sueldo final se añade al salario base los cargos sociales correspondientes.

	Salario base (€)	Cargas sociales(€)	Salario total(€)
Ingeniero Técnico Industrial, especialidad Electricidad	9000	4410	13410
Responsable asesor	3600	1764	5364
TOTAL(€)			18.774 €

2.7.-Presupuesto total:

El I.V.A. ha sido añadido en cada apartado, por lo que no es necesario añadirlo aquí. Se ha de tener en cuenta el beneficio industrial en la suma total del presupuesto.

Coste Material Laboratorio	93,05 €
Coste Personal	18.774 €
Coste Software	1.568,63 €
Coste Total Ejecución Material	20.435,68 €
Beneficio Industrial 10% (E.M.)	2.043,57 €
COSTE TOTAL	22.479,25 €

El coste del proyecto asciende a la cantidad de VEINTIDOS MIL CUATROCIENTOS SETENTA Y NUEVE EUROS CON VEINTICINCO CENTIMOS DE EURO.

ANEXOS

3.1.-MANUAL DE USUARIO

En este documento se explica el modo de utilizar el software desarrollado para el control de un horno Linn VMK39 con un controlador GEFRAN 800P.

Lo primero será iniciar el PC desde el que se controlará el horno. En la carpeta “Horno Linn VMK39” se encuentra el archivo del programa desarrollado con Labview, de nombre “proyecto horno”.

Una vez abierto ese archivo se nos mostrará la siguiente pantalla:

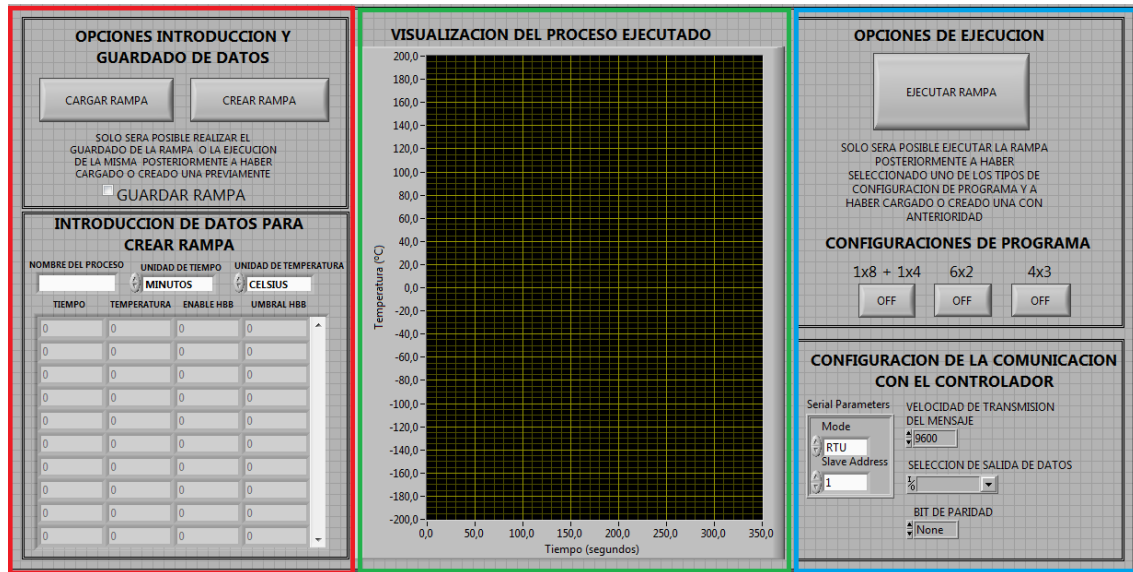


Imagen 1.

Esta es la pantalla principal del proyecto desde la cual se controla el funcionamiento que tendrá el horno. En ella quedan diferenciadas dos partes muy importantes que son el tratamiento de los datos (recuadrada en rojo), la otra que es la parte de comunicación con el horno y ejecución del proceso (recuadrada en azul) y se puede ver una gráfica (recuadrada en verde) que mostrará la rampa después de que haya sido ejecutada.

Se procede a explicar el modo de introducir los datos y posteriormente se explicará un proceso concreto de ejemplo.

INTRODUCCION DE LOS DATOS:

Se ofrecen dos maneras diferentes de cargar los datos, una es cargarlos desde un archivo (CARGAR RAMPA) y la otra es introducir los valores de temperatura y tiempo en los apartados delimitados para ello en el software desarrollado en el proyecto(CREAR RAMPA).

CARGAR RAMPA:

Para cargar un archivo solamente será necesario darle al modo “RUN” de Labview y posteriormente clicar en la opción señalada por la flecha en la imagen. Después de hacer esto se abrirá un dialogo para seleccionar un archivo en el disco duro del PC, seleccionar el archivo elegido y este se cargará en el programa.

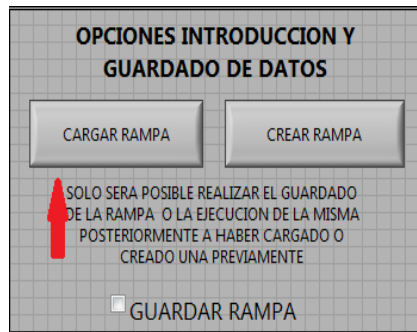


Imagen 2.

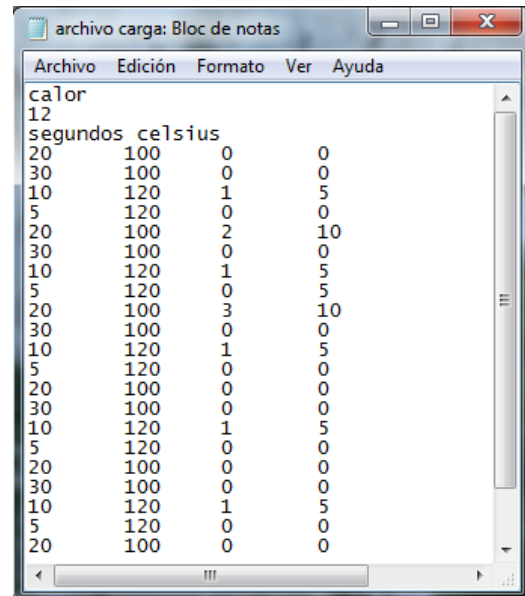


Imagen 3.

El archivo para cargar debe ser de tipo “txt”, creado por el editor de textos “bloc de notas” existente en la mayoría de PC’s. En el archivo se deben colocar los datos de la siguiente manera:

NOMBRE PROCESO			
Nº PASOS			
UD TIEMPO	UD TEMPERATURA	HAB. HBB	UMBRAL HBB
t RAMPA PASO1	Tª RAMPA PASO1	HAB SEGMENTO	UMBRAL DE GRADOS
t MANTENIMIENTO PASO1	Tª MANTENIMIENTO PASO1	-	-
t RAMPA PASO2	Tª RAMPA PASO2	HAB SEGMENTO	UMBRAL DE GRADOS
t MANTENIMIENTO PASO2	Tª MANTENIMIENTO PASO2	-	-
t RAMPA PASO3	Tª RAMPA PASO3	HAB SEGMENTO	UMBRAL DE GRADOS
t MANTENIMIENTO PASO3	Tª MANTENIMIENTO PASO3	-	-

Cada paso está definido por un segmento de rampa, otro de mantenimiento, la habilitación de la HBB y el umbral designado para ella. A continuación se muestra y explica un ejemplo:

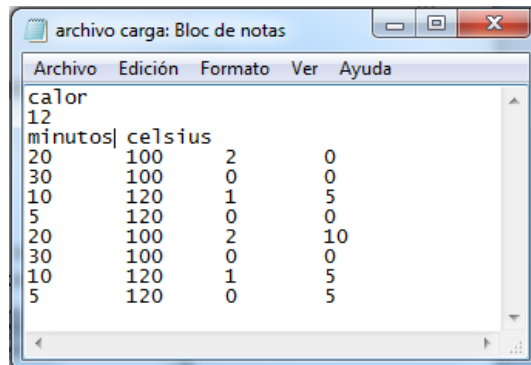


Imagen 4.

Para el primer paso se ha designado un tiempo de rampa de 20 minutos y una temperatura de 100°C, se habilita la HBB en el segmento de mantenimiento con un umbral de 0°C. Y se le asigna un tiempo de mantenimiento de 30 minutos a 100°C. Así sucesivamente con cada paso. Se recomienda introducir un valor de 3 en la habilitación de la función HBB y un umbral de 2 °C. Para ello basta con introducir estos valores en la primera fila del paso en concreto y quedará definida la función en ese paso.

CREAR RAMPA:

Esta función permite crear una rampa desde el propio programa, para ello será preciso introducir los datos en los campos señalados en rojo en la imagen 5.

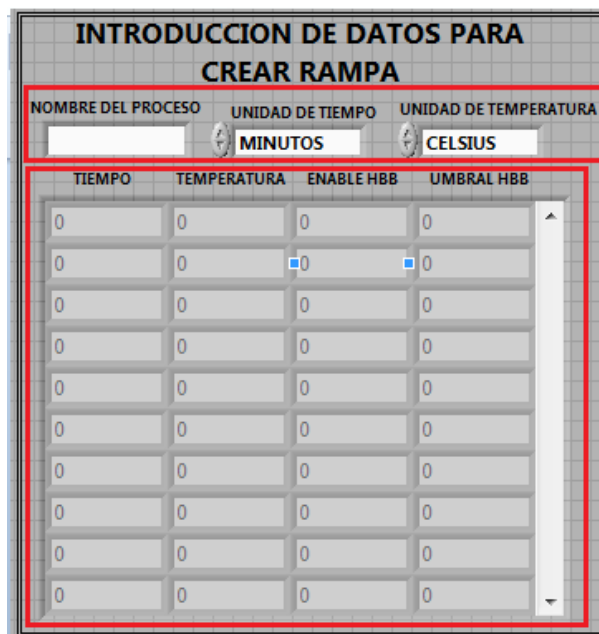


Imagen 5.

La rampa que se desea crear es de temperatura/tiempo, es necesario introducir las unidades en las que serán introducidos los valores de tiempo y temperatura y el nombre que recibirá este proceso.

Para introducir los valores de la rampa se debe tener en cuenta que cada paso deberá tener dos partes, una de rampa y otra de mantenimiento. En la de rampa, para el primer paso el valor partirá desde la variable de proceso e irá hasta el valor delimitado de temperatura en el tiempo que se le indique. Y en la de mantenimiento mantendrá el valor de temperatura delimitado durante el tiempo que este estipulado.

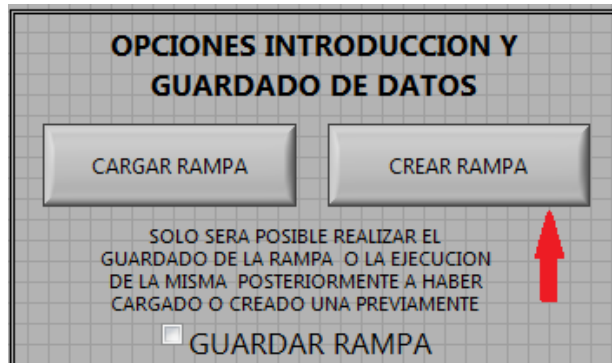


Imagen 5.

Para cargar la rampa se deben introducir los datos previamente en los campos mostrados. Posteriormente pulsar el botón de “run” de labview y a continuación pulsar el botón “crear rampa” mostrado en la imagen 5.

EJEMPLO DE EJECUCIÓN DE UN PROCESO:

1.- Configurar la comunicación con el dispositivo: Es necesario ver si las características de la comunicación son las siguientes para que sea posible el proceso.

- Velocidad = 9600
- Paridad = None
- Puerto de salida de datos = COM1 (si el PC tiene más de un puerto serie podría ser COM2, COM3, etc)
- Mode (parámetros serie) = RTU
- Slave address = 1

2.- Configuración del proceso: es necesario seleccionar una de las configuraciones de programa visibles en la imagen 6 así como el botón de ejecutar señalado.

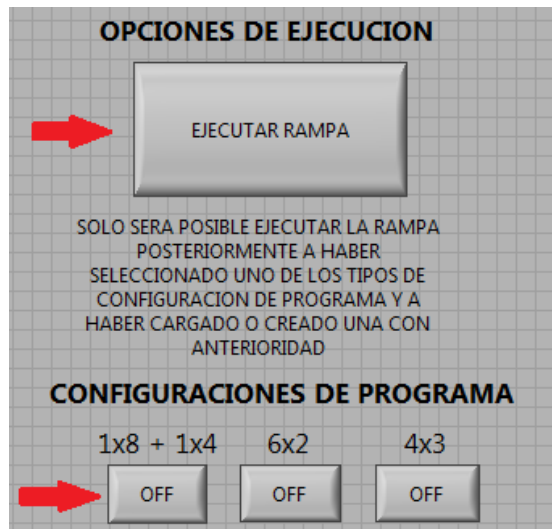


Imagen 6.

3.- Guardado de los datos de la rampa:

Para que estos se guarden es necesario marcar el checkbox existente en la casilla de opciones de introducción de los datos.

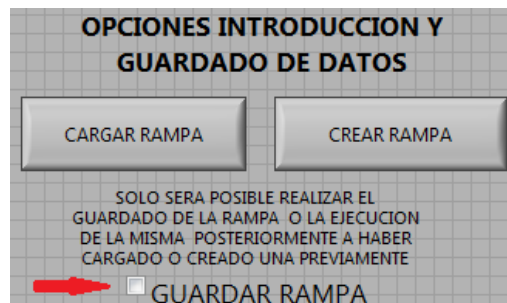


Imagen 7.

Cabe la posibilidad de introducir un proceso y guardarlo sin necesidad de ejecutarlo. Para ello se introducen los datos como se ha indicado, se marca el checkbox de “guardar rampa” y se da a la tecla “run” de Labview. Mostrará un mensaje de advertencia diciendo que no se ha marcado ningún tipo de ejecución y procederá al guardado de los datos.

4.- Introducción de los datos:

- Cargar rampa: en este caso una vez hechos los pasos anteriores y teniendo preparado el archivo que queremos ejecutar, debemos pulsar el botón “run” de labview y luego el pulsador “cargar rampa”. Se mostrará una ventana en la que podremos seleccionar el archivo previamente preparado.

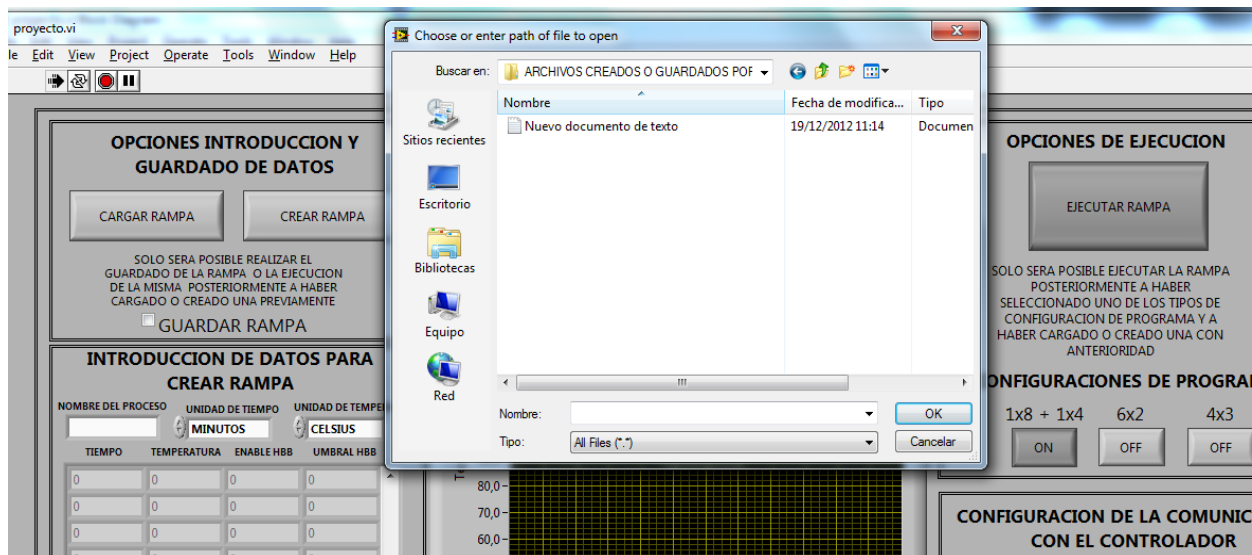


Imagen 8.

Una vez seleccionado este archivo el programa pasará al siguiente paso.

- Crear rampa: En este caso es necesario introducir los datos en los casilleros como se ha explicado antes y posteriormente pulsar el botón “run” de labview y luego el pulsador “crear rampa”.

5.- Ejecución del proceso:

Una vez realizados los pasos anteriores, el programa cargará los datos en los registros del controlador y cuando haya terminado con esta tarea mostrará un nuevo panel frontal de Labview (imagen 9).

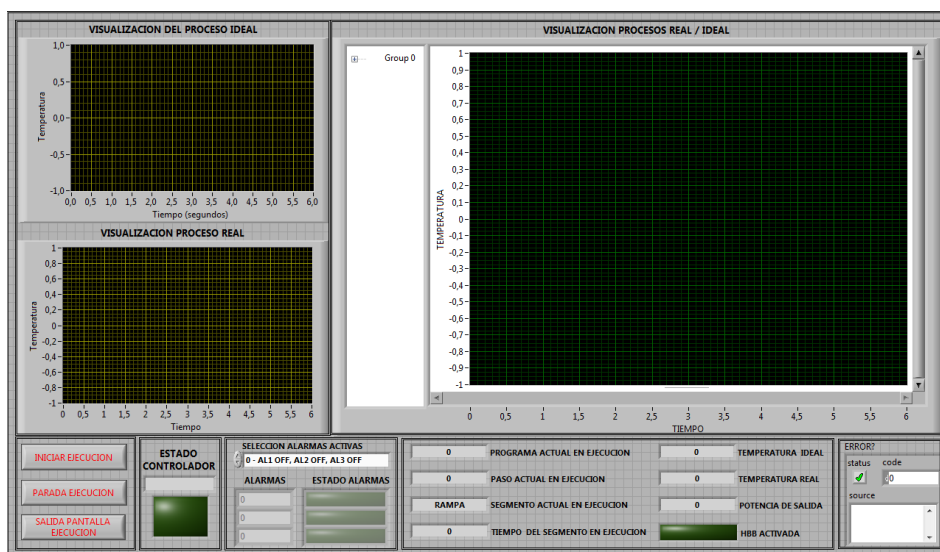


Imagen 9.

Dentro de este panel nos será posible seleccionar el número de alarmas activas que habrá durante el proceso de ejecución y el valor umbral para ellas. Una vez hecho esto lo siguiente es pulsar el botón de “iniciar ejecución” y el proceso que haya sido cargado comenzará a ejecutarse.

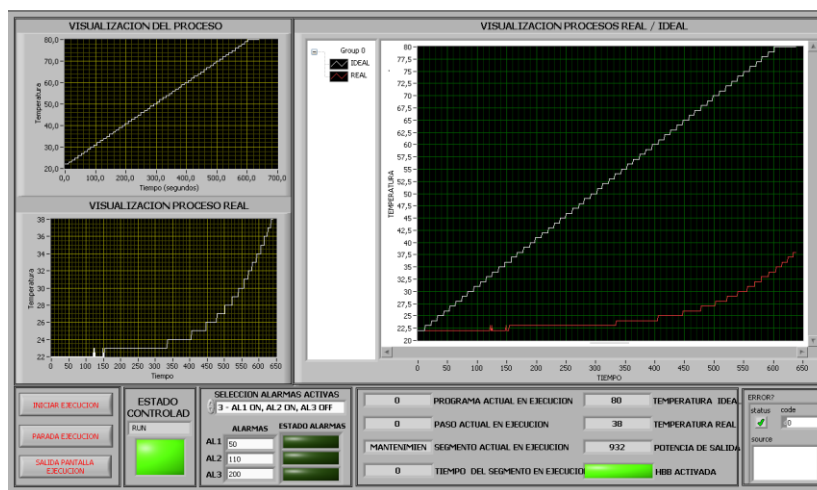


Imagen 10. Proceso recién iniciado.

Durante la ejecución podemos pulsar “parada de la ejecución” en cualquier momento y esta se parará, o esperaremos al final del proceso en ejecución. Para que el programa pueda salir de esta pantalla es necesario pulsar ese botón y posteriormente el de “salida pantalla ejecución” y el programa volverá a la pantalla inicial.

6.- Guardado de los datos: Una vez que se termina el proceso en ejecución y hemos marcado el checkbox de guardar rampa el programa nos mostrará una ventana de diálogo para que seleccionemos el archivo en el que queremos guardar los datos. Después de esto el programa ha terminado su proceso y mostrará en la gráfica de la pantalla principal el proceso ideal y terminará. En el caso que no haya sido marcado el checkbox de guardar rampa el programa mostrará en la gráfica de la pantalla principal el proceso ideal y terminará.

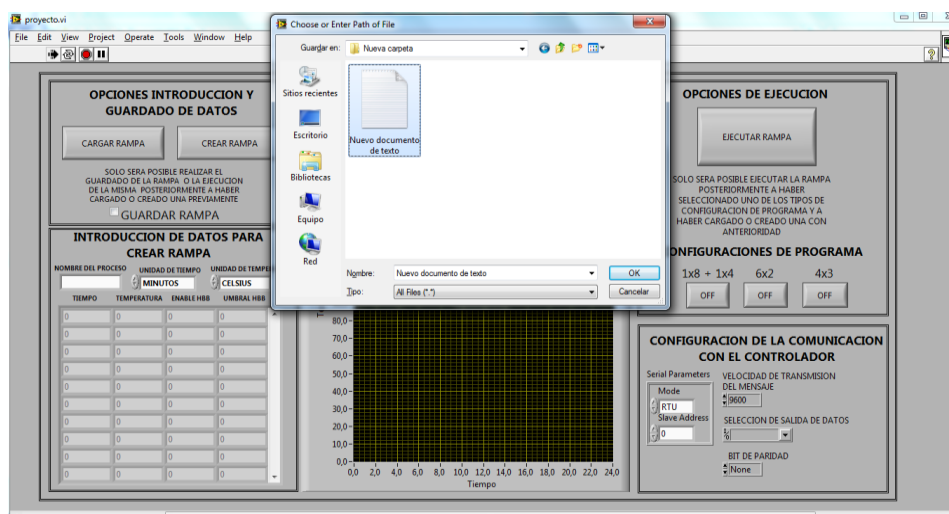


Imagen 11. Diálogo de guardado de archivo.

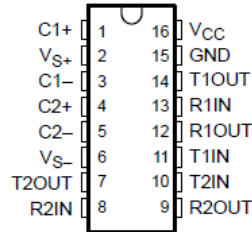
• 3.2.-DATASHEET MAX232

MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLLS0471 – FEBRUARY 1989 – REVISED OCTOBER 2002

- Meet or Exceed TIA/EIA-232-F and ITU Recommendation V.28
- Operate With Single 5-V Power Supply
- Operate Up to 120 kbit/s
- Two Drivers and Two Receivers
- ± 30 -V Input Levels
- Low Supply Current . . . 8 mA Typical
- Designed to be Interchangeable With Maxim MAX232
- ESD Protection Exceeds JESD 22 – 2000-V Human-Body Model (A114-A)
- Applications
 - TIA/EIA-232-F
 - Battery-Powered Systems
 - Terminals
 - Modems
 - Computers

MAX232 . . . D, DW, N, OR NS PACKAGE
 MAX232I . . . D, DW, OR N PACKAGE
 (TOP VIEW)



description/ordering information

The MAX232 is a dual driver/receiver that includes a capacitive voltage generator to supply EIA-232 voltage levels from a single 5-V supply. Each receiver converts EIA-232 inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V and a typical hysteresis of 0.5 V, and can accept ± 30 -V inputs. Each driver converts TTL/CMOS input levels into EIA-232 levels. The driver, receiver, and voltage-generator functions are available as cells in the Texas Instruments LinASIC™ library.

ORDERING INFORMATION

TA	PACKAGE†		ORDERABLE PART NUMBER	TOP-SIDE MARKING
0°C to 70°C	PDIP (N)	Tube	MAX232N	MAX232N
		SOIC (D)	Tube	MAX232D
		Tape and reel	MAX232DR	
	SOIC (DW)	Tube	MAX232DW	MAX232
		Tape and reel	MAX232DWR	
	SOP (NS)	Tape and reel	MAX232NSR	MAX232
-40°C to 85°C	PDIP (N)	Tube	MAX232IN	MAX232IN
		SOIC (D)	Tube	MAX232ID
		Tape and reel	MAX232IDR	
	SOIC (DW)	Tube	MAX232IDW	MAX232I
		Tape and reel	MAX232IDWR	

† Package drawings, standard packing quantities, thermal data, symbolization, and PCB design guidelines are available at www.ti.com/sc/package.



Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

LinASIC is a trademark of Texas Instruments.

PRODUCTION DATA information is current as of publication date. Products conform to specifications per the terms of Texas Instruments standard warranty. Production processing does not necessarily include testing of all parameters.

Copyright © 2002, Texas Instruments Incorporated



POST OFFICE BOX 655303 • DALLAS, TEXAS 75265

1

MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047I – FEBRUARY 1989 – REVISED OCTOBER 2002

Function Tables

EACH DRIVER

INPUT TIN	OUTPUT TOUT
L	H
H	L

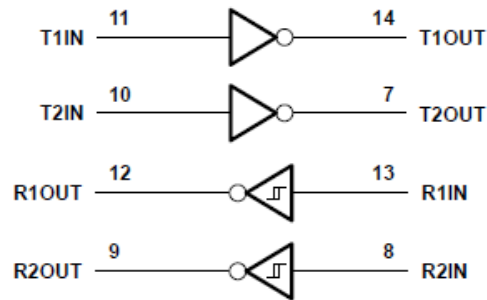
H = high level, L = low level

EACH RECEIVER

INPUT RIN	OUTPUT ROUT
L	H
H	L

H = high level, L = low level

logic diagram (positive logic)



absolute maximum ratings over operating free-air temperature range (unless otherwise noted)†

Input supply voltage range, V_{CC} (see Note 1)	-0.3 V to 6 V
Positive output supply voltage range, V_{S+}	$V_{CC} - 0.3$ V to 15 V
Negative output supply voltage range, V_{S-}	-0.3 V to -15 V
Input voltage range, V_I : Driver	-0.3 V to $V_{CC} + 0.3$ V
Receiver	± 30 V
Output voltage range, V_O : T1OUT, T2OUT	$V_{S-} - 0.3$ V to $V_{S+} + 0.3$ V
R1OUT, R2OUT	-0.3 V to $V_{CC} + 0.3$ V
Short-circuit duration: T1OUT, T2OUT	Unlimited
Package thermal impedance, θ_{JA} (see Note 2): D package	73°C/W
DW package	57°C/W
N package	67°C/W
NS package	64°C/W
Lead temperature 1,6 mm (1/16 inch) from case for 10 seconds	260°C
Storage temperature range, T_{stg}	-65°C to 150°C

† Stresses beyond those listed under "absolute maximum ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under "recommended operating conditions" is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 1: All voltage values are with respect to network ground terminal.

2. The package thermal impedance is calculated in accordance with JESD 51-7.

recommended operating conditions

		MIN	NOM	MAX	UNIT
V_{CC}	Supply voltage	4.5	5	5.5	V
V_{IH}	High-level input voltage (T1IN, T2IN)	2			V
V_{IL}	Low-level input voltage (T1IN, T2IN)			0.8	V
R1IN, R2IN	Receiver input voltage			± 30	V
T_A	Operating free-air temperature	MAX232	0	70	°C
		MAX232I	-40	85	

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted) (see Note 3 and Figure 4)

PARAMETER		TEST CONDITIONS		MIN	TYP‡	MAX	UNIT
I_{CC}	Supply current	$V_{CC} = 5.5$ V, $T_A = 25^\circ\text{C}$	All outputs open,		8	10	mA

‡ All typical values are at $V_{CC} = 5$ V and $T_A = 25^\circ\text{C}$.

NOTE 3: Test conditions are C1–C4 = 1 μF at $V_{CC} = 5$ V ± 0.5 V.

MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLLS047I – FEBRUARY 1989 – REVISED OCTOBER 2002

DRIVER SECTION

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 3)

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V _{OH}	High-level output voltage	T1OUT, T2OUT R _L = 3 kΩ to GND	5	7		V
V _{OL}	Low-level output voltage‡	T1OUT, T2OUT R _L = 3 kΩ to GND		-7	-5	V
r _o	Output resistance	T1OUT, T2OUT V _{S+} = V _{S-} = 0, V _O = ±2 V	300			Ω
I _{OS} §	Short-circuit output current	T1OUT, T2OUT V _{CC} = 5.5 V, V _O = 0		±10		mA
I _{IS}	Short-circuit input current	T1IN, T2IN V _I = 0			200	μA

† All typical values are at V_{CC} = 5 V, T_A = 25°C.

‡ The algebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

§ Not more than one output should be shorted at a time.

NOTE 3: Test conditions are C1–C4 = 1 μF at V_{CC} = 5 V ± 0.5 V.

switching characteristics, V_{CC} = 5 V, T_A = 25°C (see Note 3)

PARAMETER		TEST CONDITIONS	MIN	TYP	MAX	UNIT
SR	Driver slew rate	R _L = 3 kΩ to 7 kΩ, See Figure 2			30	V/μs
SR(t)	Driver transition region slew rate	See Figure 3		3		V/μs
	Data rate	One TOUT switching		120		kbit/s

NOTE 3: Test conditions are C1–C4 = 1 μF at V_{CC} = 5 V ± 0.5 V.

RECEIVER SECTION

electrical characteristics over recommended ranges of supply voltage and operating free-air temperature range (see Note 3)

PARAMETER		TEST CONDITIONS	MIN	TYP†	MAX	UNIT
V _{OH}	High-level output voltage	R1OUT, R2OUT I _{OH} = -1 mA	3.5			V
V _{OL}	Low-level output voltage‡	R1OUT, R2OUT I _{OL} = 3.2 mA			0.4	V
V _{IT+}	Receiver positive-going input threshold voltage	R1IN, R2IN V _{CC} = 5 V, T _A = 25°C		1.7	2.4	V
V _{IT-}	Receiver negative-going input threshold voltage	R1IN, R2IN V _{CC} = 5 V, T _A = 25°C	0.8	1.2		V
V _{hys}	Input hysteresis voltage	R1IN, R2IN V _{CC} = 5 V	0.2	0.5	1	V
r _i	Receiver input resistance	R1IN, R2IN V _{CC} = 5, T _A = 25°C	3	5	7	kΩ

† All typical values are at V_{CC} = 5 V, T_A = 25°C.

‡ The algebraic convention, in which the least positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

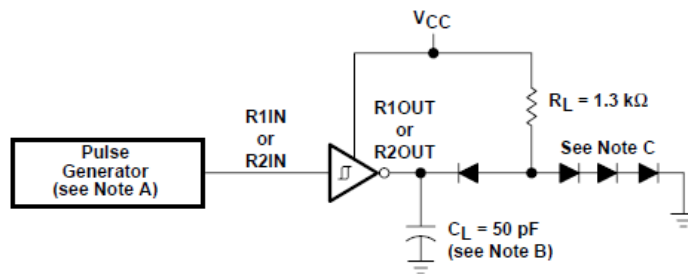
NOTE 3: Test conditions are C1–C4 = 1 μF at V_{CC} = 5 V ± 0.5 V.

switching characteristics, V_{CC} = 5 V, T_A = 25°C (see Note 3 and Figure 1)

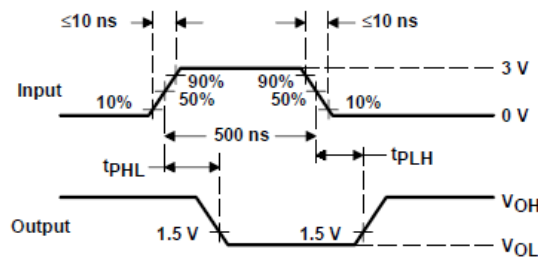
PARAMETER		TYP	UNIT
t _{PLH(R)}	Receiver propagation delay time, low- to high-level output	500	ns
t _{PHL(R)}	Receiver propagation delay time, high- to low-level output	500	ns

NOTE 3: Test conditions are C1–C4 = 1 μF at V_{CC} = 5 V ± 0.5 V.

PARAMETER MEASUREMENT INFORMATION



TEST CIRCUIT



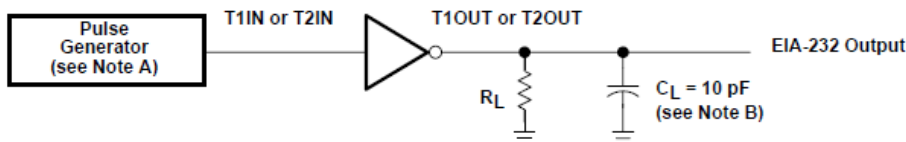
WAVEFORMS

- NOTES: A. The pulse generator has the following characteristics: $Z_O = 50 \Omega$, duty cycle $\leq 50\%$.
 B. C_L includes probe and jig capacitance.
 C. All diodes are 1N3064 or equivalent.

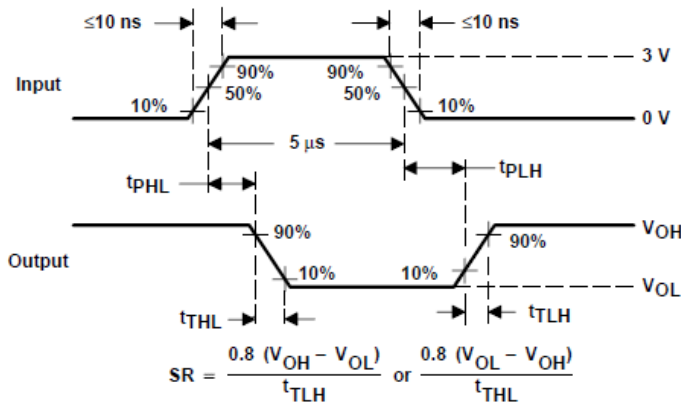
Figure 1. Receiver Test Circuit and Waveforms for t_{PHL} and t_{PLH} Measurements

MAX232, MAX232I
 DUAL EIA-232 DRIVERS/RECEIVERS

PARAMETER MEASUREMENT INFORMATION



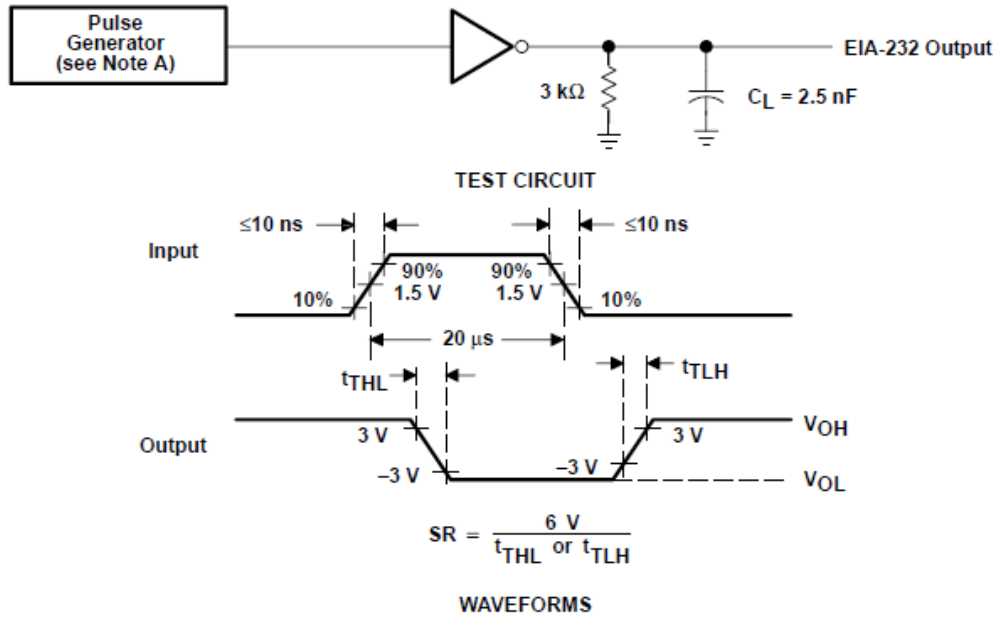
TEST CIRCUIT



WAVEFORMS

- NOTES: A. The pulse generator has the following characteristics: $Z_O = 50 \Omega$, duty cycle $\leq 50\%$.
 B. C_L includes probe and jig capacitance.

Figure 2. Driver Test Circuit and Waveforms for t_{PHL} and t_{PLH} Measurements (5- μ s Input)



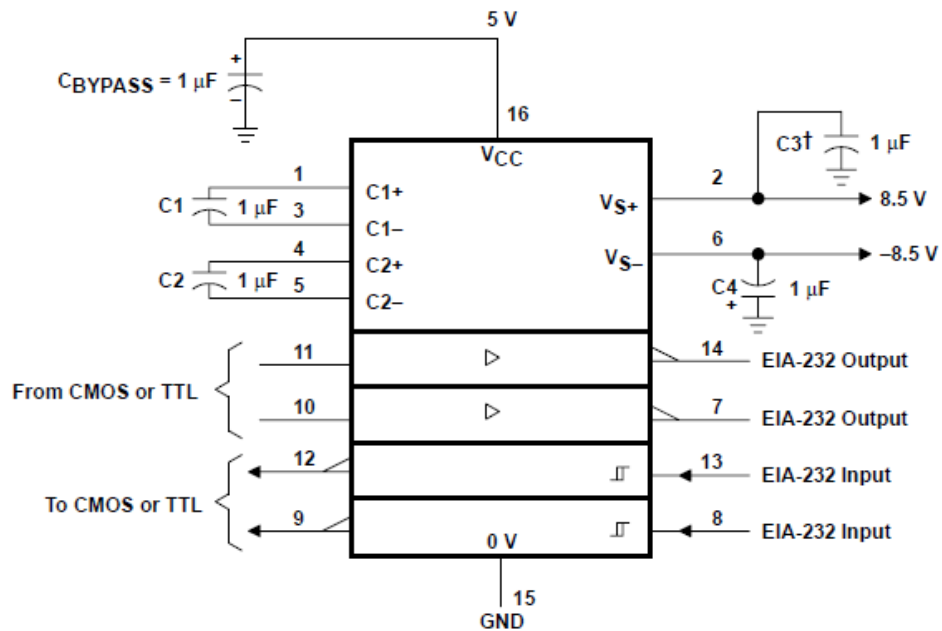
NOTE A: The pulse generator has the following characteristics: $Z_O = 50 \Omega$, duty cycle $\leq 50\%$.

Figure 3. Test Circuit and Waveforms for t_{THL} and t_{TLH} Measurements (20- μ s Input)

MAX232, MAX232I DUAL EIA-232 DRIVERS/RECEIVERS

SLLS0471 – FEBRUARY 1989 – REVISED OCTOBER 2002

APPLICATION INFORMATION



† C3 can be connected to V_{CC} or GND.

Figure 4. Typical Operating Circuit

• 3.3.-DATASHEET MAX485

19-0122; Rev 7; 6/03



Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

General Description

The MAX481, MAX483, MAX485, MAX487–MAX491, and MAX1487 are low-power transceivers for RS-485 and RS-422 communication. Each part contains one driver and one receiver. The MAX483, MAX487, MAX488, and MAX489 feature reduced slew-rate drivers that minimize EMI and reduce reflections caused by improperly terminated cables, thus allowing error-free data transmission up to 250kbps. The driver slew rates of the MAX481, MAX485, MAX490, MAX491, and MAX1487 are not limited, allowing them to transmit up to 2.5Mbps.

These transceivers draw between 120µA and 500µA of supply current when unloaded or fully loaded with disabled drivers. Additionally, the MAX481, MAX483, and MAX487 have a low-current shutdown mode in which they consume only 0.1µA. All parts operate from a single 5V supply.

Drivers are short-circuit current limited and are protected against excessive power dissipation by thermal shutdown circuitry that places the driver outputs into a high-impedance state. The receiver input has a fail-safe feature that guarantees a logic-high output if the input is open circuit.

The MAX487 and MAX1487 feature quarter-unit-load receiver input impedance, allowing up to 128 MAX487/MAX1487 transceivers on the bus. Full-duplex communications are obtained using the MAX488–MAX491, while the MAX481, MAX483, MAX485, MAX487, and MAX1487 are designed for half-duplex applications.

Applications

- Low-Power RS-485 Transceivers
- Low-Power RS-422 Transceivers
- Level Translators
- Transceivers for EMI-Sensitive Applications
- Industrial-Control Local Area Networks

Features

- ◆ In µMAX Package: Smallest 8-Pin SO
- ◆ Slew-Rate Limited for Error-Free Data Transmission (MAX483/487/488/489)
- ◆ 0.1µA Low-Current Shutdown Mode (MAX481/483/487)
- ◆ Low Quiescent Current:
120µA (MAX483/487/488/489)
230µA (MAX1487)
300µA (MAX481/485/490/491)
- ◆ -7V to +12V Common-Mode Input Voltage Range
- ◆ Three-State Outputs
- ◆ 30ns Propagation Delays, 5ns Skew (MAX481/485/490/491/1487)
- ◆ Full-Duplex and Half-Duplex Versions Available
- ◆ Operate from a Single 5V Supply
- ◆ Allows up to 128 Transceivers on the Bus (MAX487/MAX1487)
- ◆ Current-Limiting and Thermal Shutdown for Driver Overload Protection

Ordering Information

PART	TEMP RANGE	PIN-PACKAGE
MAX481CPA	0°C to +70°C	8 Plastic DIP
MAX481CSA	0°C to +70°C	8 SO
MAX481CUA	0°C to +70°C	8 µMAX
MAX481C/D	0°C to +70°C	Dice*

Ordering Information continued at end of data sheet.
 *Contact factory for dice specifications.

Selection Table

PART NUMBER	HALF/FULL DUPLEX	DATA RATE (Mbps)	SLEW-RATE LIMITED	LOW-POWER SHUTDOWN	RECEIVER/ DRIVER ENABLE	QUIESCENT CURRENT (µA)	NUMBER OF TRANSMITTERS ON BUS	PIN COUNT
MAX481	Half	2.5	No	Yes	Yes	300	32	8
MAX483	Half	0.25	Yes	Yes	Yes	120	32	8
MAX485	Half	2.5	No	No	Yes	300	32	8
MAX487	Half	0.25	Yes	Yes	Yes	120	128	8
MAX488	Full	0.25	Yes	No	No	120	32	8
MAX489	Full	0.25	Yes	No	Yes	120	32	14
MAX490	Full	2.5	No	No	No	300	32	8
MAX491	Full	2.5	No	No	Yes	300	32	14
MAX1487	Half	2.5	No	No	Yes	230	128	8



Maxim Integrated Products 1

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

ABSOLUTE MAXIMUM RATINGS

Supply Voltage (V _{CC}).....	12V	14-Pin SO (derate 8.33mW/°C above +70°C).....	667mW
Control Input Voltage (RE, DE).....	-0.5V to (V _{CC} + 0.5V)	8-Pin μMAX (derate 4.1mW/°C above +70°C).....	830mW
Driver Input Voltage (DI).....	-0.5V to (V _{CC} + 0.5V)	8-Pin CERDIP (derate 8.00mW/°C above +70°C).....	640mW
Driver Output Voltage (A, B).....	-8V to +12.5V	14-Pin CERDIP (derate 9.09mW/°C above +70°C).....	727mW
Receiver Input Voltage (A, B).....	-8V to +12.5V	Operating Temperature Ranges	
Receiver Output Voltage (RO).....	-0.5V to (V _{CC} + 0.5V)	MAX4_C_/MAX1487C_A.....	0°C to +70°C
Continuous Power Dissipation (T _A = +70°C)		MAX4_E_/MAX1487E_A.....	-40°C to +85°C
8-Pin Plastic DIP (derate 9.09mW/°C above +70°C)	727mW	MAX4_MJ_/MAX1487MJA.....	-55°C to +125°C
14-Pin Plastic DIP (derate 10.00mW/°C above +70°C) ..	800mW	Storage Temperature Range.....	
8-Pin SO (derate 5.88mW/°C above +70°C).....	471mW	-65°C to +160°C	
		Lead Temperature (soldering, 10sec).....	
		+300°C	

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC ELECTRICAL CHARACTERISTICS

(V_{CC} = 5V ±5%, T_A = T_{MIN} to T_{MAX}, unless otherwise noted.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Differential Driver Output (no load)	V _{OD1}				5	V
Differential Driver Output (with load)	V _{OD2}	R = 50Ω (RS-422)	2			V
		R = 27Ω (RS-485), Figure 4	1.5		5	V
Change in Magnitude of Driver Differential Output Voltage for Complementary Output States	ΔV _{OD}	R = 27Ω or 50Ω, Figure 4			0.2	V
Driver Common-Mode Output Voltage	V _{OC}	R = 27Ω or 50Ω, Figure 4			3	V
Change in Magnitude of Driver Common-Mode Output Voltage for Complementary Output States	ΔV _{OC}	R = 27Ω or 50Ω, Figure 4			0.2	V
Input High Voltage	V _{IH}	DE, DI, RE	2.0			V
Input Low Voltage	V _{IL}	DE, DI, RE			0.8	V
Input Current	I _{IN1}	DE, DI, RE			±2	μA
Input Current (A, B)	I _{IN2}	DE = 0V; V _{CC} = 0V or 5.25V, all devices except MAX487/MAX1487	V _{IN} = 12V		1.0	mA
			V _{IN} = -7V		-0.8	
		MAX487/MAX1487, DE = 0V, V _{CC} = 0V or 5.25V	V _{IN} = 12V		0.25	mA
			V _{IN} = -7V		-0.2	
Receiver Differential Threshold Voltage	V _{TH}	-7V ≤ V _{CM} ≤ 12V	-0.2		0.2	V
Receiver Input Hysteresis	ΔV _{TH}	V _{CM} = 0V		70		mV
Receiver Output High Voltage	V _{OH}	I _O = -4mA, V _{ID} = 200mV	3.5			V
Receiver Output Low Voltage	V _{OL}	I _O = 4mA, V _{ID} = -200mV			0.4	V
Three-State (high impedance) Output Current at Receiver	I _{OZR}	0.4V ≤ V _O ≤ 2.4V			±1	μA
Receiver Input Resistance	R _{IN}	-7V ≤ V _{CM} ≤ 12V, all devices except MAX487/MAX1487	12			kΩ
		-7V ≤ V _{CM} ≤ 12V, MAX487/MAX1487	48			kΩ

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

DC ELECTRICAL CHARACTERISTICS (continued)

($V_{CC} = 5V \pm 5\%$, $T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
No-Load Supply Current (Note 3)	I_{CC}	MAX488/MAX489, DE, DI, RE = 0V or V_{CC}		120	250	μA
		MAX490/MAX491, DE, DI, RE = 0V or V_{CC}		300	500	
		MAX481/MAX485, RE = 0V or V_{CC}	DE = V_{CC}	500	900	
			DE = 0V	300	500	
		MAX1487, RE = 0V or V_{CC}	DE = V_{CC}	300	500	
			DE = 0V	230	400	
MAX483/MAX487, RE = 0V or V_{CC}	DE = 5V	MAX483	350	650		
		MAX487	250	400		
	DE = 0V		120	250		
Supply Current in Shutdown	I_{SHDN}	MAX481/483/487, DE = 0V, RE = V_{CC}		0.1	10	μA
Driver Short-Circuit Current, $V_O = \text{High}$	I_{OSD1}	$-7V \leq V_O \leq 12V$ (Note 4)	35		250	mA
Driver Short-Circuit Current, $V_O = \text{Low}$	I_{OSD2}	$-7V \leq V_O \leq 12V$ (Note 4)	35		250	mA
Receiver Short-Circuit Current	I_{OSR}	$0V \leq V_O \leq V_{CC}$	7		95	mA

SWITCHING CHARACTERISTICS—MAX481/MAX485, MAX490/MAX491, MAX1487

($V_{CC} = 5V \pm 5\%$, $T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS	
Driver Input to Output	t_{PLH}	Figures 6 and 8, $R_{DIFF} = 54\Omega$, $C_{L1} = C_{L2} = 100pF$	10	30	60	ns	
	t_{PHL}		10	30	60		
Driver Output Skew to Output	t_{SKEW}	Figures 6 and 8, $R_{DIFF} = 54\Omega$, $C_{L1} = C_{L2} = 100pF$		5	10	ns	
Driver Rise or Fall Time	t_R, t_F	Figures 6 and 8, $R_{DIFF} = 54\Omega$, $C_{L1} = C_{L2} = 100pF$	MAX481, MAX485, MAX1487	3	15	40	ns
			MAX490C/E, MAX491C/E	5	15	25	
			MAX490M, MAX491M	3	15	40	
Driver Enable to Output High	t_{ZH}	Figures 7 and 9, $C_L = 100pF$, S2 closed		40	70	ns	
Driver Enable to Output Low	t_{ZL}	Figures 7 and 9, $C_L = 100pF$, S1 closed		40	70	ns	
Driver Disable Time from Low	t_{LZ}	Figures 7 and 9, $C_L = 15pF$, S1 closed		40	70	ns	
Driver Disable Time from High	t_{HZ}	Figures 7 and 9, $C_L = 15pF$, S2 closed		40	70	ns	
Receiver Input to Output	t_{PLH}, t_{PHL}	Figures 6 and 10, $R_{DIFF} = 54\Omega$, $C_{L1} = C_{L2} = 100pF$	MAX481, MAX485, MAX1487	20	90	200	ns
			MAX490C/E, MAX491C/E	20	90	150	
			MAX490M, MAX491M	20	90	200	
$ t_{PLH} - t_{PHL} $ Differential Receiver Skew	t_{SKD}	Figures 6 and 10, $R_{DIFF} = 54\Omega$, $C_{L1} = C_{L2} = 100pF$		13		ns	
Receiver Enable to Output Low	t_{ZL}	Figures 5 and 11, $C_{RL} = 15pF$, S1 closed		20	50	ns	
Receiver Enable to Output High	t_{ZH}	Figures 5 and 11, $C_{RL} = 15pF$, S2 closed		20	50	ns	
Receiver Disable Time from Low	t_{LZ}	Figures 5 and 11, $C_{RL} = 15pF$, S1 closed		20	50	ns	
Receiver Disable Time from High	t_{HZ}	Figures 5 and 11, $C_{RL} = 15pF$, S2 closed		20	50	ns	
Maximum Data Rate	f_{MAX}		2.5			Mbps	
Time to Shutdown	t_{SHDN}	MAX481 (Note 5)	50	200	600	ns	

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

SWITCHING CHARACTERISTICS—MAX481/MAX485, MAX490/MAX491, MAX1487 (continued)

($V_{CC} = 5V \pm 5\%$, $T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Driver Enable from Shutdown to Output High (MAX481)	$t_{ZH}(SHDN)$	Figures 7 and 9, $C_L = 100pF$, S2 closed		40	100	ns
Driver Enable from Shutdown to Output Low (MAX481)	$t_{ZL}(SHDN)$	Figures 7 and 9, $C_L = 100pF$, S1 closed		40	100	ns
Receiver Enable from Shutdown to Output High (MAX481)	$t_{ZH}(SHDN)$	Figures 5 and 11, $C_L = 15pF$, S2 closed, A - B = 2V		300	1000	ns
Receiver Enable from Shutdown to Output Low (MAX481)	$t_{ZL}(SHDN)$	Figures 5 and 11, $C_L = 15pF$, S1 closed, B - A = 2V		300	1000	ns

SWITCHING CHARACTERISTICS—MAX483, MAX487/MAX488/MAX489

($V_{CC} = 5V \pm 5\%$, $T_A = T_{MIN}$ to T_{MAX} , unless otherwise noted.) (Notes 1, 2)

PARAMETER	SYMBOL	CONDITIONS	MIN	TYP	MAX	UNITS
Driver Input to Output	t_{PLH}	Figures 6 and 8, $R_{DIFF} = 54\Omega$, $C_{L1} = C_{L2} = 100pF$	250	800	2000	ns
	t_{PHL}		250	800	2000	
Driver Output Skew to Output	t_{SKEW}	Figures 6 and 8, $R_{DIFF} = 54\Omega$, $C_{L1} = C_{L2} = 100pF$		100	800	ns
Driver Rise or Fall Time	t_R, t_F	Figures 6 and 8, $R_{DIFF} = 54\Omega$, $C_{L1} = C_{L2} = 100pF$	250		2000	ns
Driver Enable to Output High	t_{ZH}	Figures 7 and 9, $C_L = 100pF$, S2 closed	250		2000	ns
Driver Enable to Output Low	t_{ZL}	Figures 7 and 9, $C_L = 100pF$, S1 closed	250		2000	ns
Driver Disable Time from Low	t_{LZ}	Figures 7 and 9, $C_L = 15pF$, S1 closed	300		3000	ns
Driver Disable Time from High	t_{HZ}	Figures 7 and 9, $C_L = 15pF$, S2 closed	300		3000	ns
Receiver Input to Output	t_{PLH}	Figures 6 and 10, $R_{DIFF} = 54\Omega$, $C_{L1} = C_{L2} = 100pF$	250		2000	ns
	t_{PHL}		250		2000	
$t_{PLH} - t_{PHL}$ Differential Receiver Skew	t_{SKD}	Figures 6 and 10, $R_{DIFF} = 54\Omega$, $C_{L1} = C_{L2} = 100pF$		100		ns
Receiver Enable to Output Low	t_{ZL}	Figures 5 and 11, $C_{RL} = 15pF$, S1 closed		20	50	ns
Receiver Enable to Output High	t_{ZH}	Figures 5 and 11, $C_{RL} = 15pF$, S2 closed		20	50	ns
Receiver Disable Time from Low	t_{LZ}	Figures 5 and 11, $C_{RL} = 15pF$, S1 closed		20	50	ns
Receiver Disable Time from High	t_{HZ}	Figures 5 and 11, $C_{RL} = 15pF$, S2 closed		20	50	ns
Maximum Data Rate	f_{MAX}	$t_{PLH}, t_{PHL} < 50\%$ of data period	250			kbps
Time to Shutdown	t_{SHDN}	MAX483/MAX487 (Note 5)	50	200	600	ns
Driver Enable from Shutdown to Output High	$t_{ZH}(SHDN)$	MAX483/MAX487, Figures 7 and 9, $C_L = 100pF$, S2 closed			2000	ns
Driver Enable from Shutdown to Output Low	$t_{ZL}(SHDN)$	MAX483/MAX487, Figures 7 and 9, $C_L = 100pF$, S1 closed			2000	ns
Receiver Enable from Shutdown to Output High	$t_{ZH}(SHDN)$	MAX483/MAX487, Figures 5 and 11, $C_L = 15pF$, S2 closed			2500	ns
Receiver Enable from Shutdown to Output Low	$t_{ZL}(SHDN)$	MAX483/MAX487, Figures 5 and 11, $C_L = 15pF$, S1 closed			2500	ns

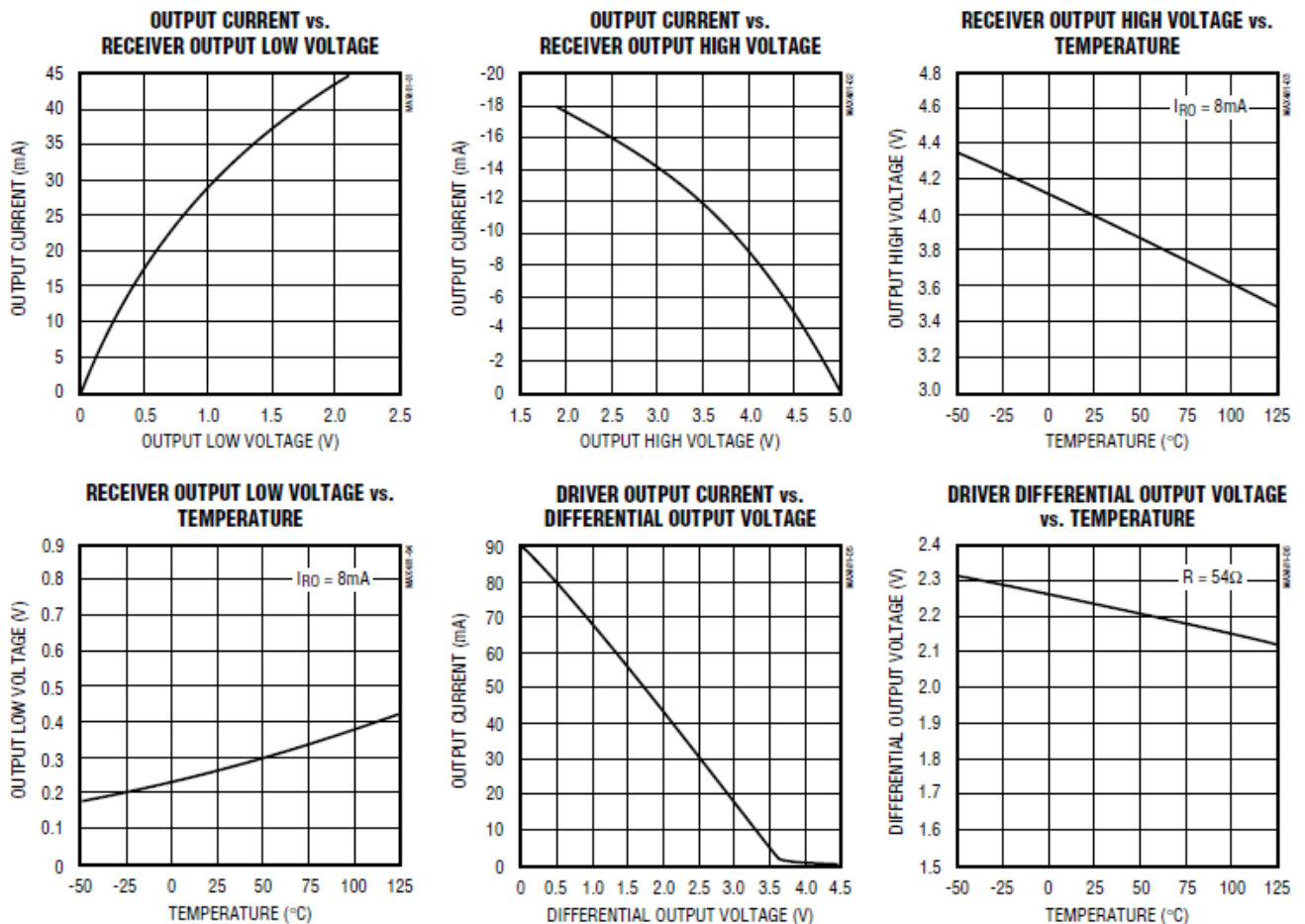
Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

NOTES FOR ELECTRICAL/SWITCHING CHARACTERISTICS

- Note 1:** All currents into device pins are positive; all currents out of device pins are negative. All voltages are referenced to device ground unless otherwise specified.
- Note 2:** All typical specifications are given for $V_{CC} = 5V$ and $T_A = +25^\circ C$.
- Note 3:** Supply current specification is valid for loaded transmitters when $DE = 0V$.
- Note 4:** Applies to peak current. See *Typical Operating Characteristics*.
- Note 5:** The MAX481/MAX483/MAX487 are put into shutdown by bringing \overline{RE} high and DE low. If the inputs are in this state for less than 50ns, the parts are guaranteed not to enter shutdown. If the inputs are in this state for at least 600ns, the parts are guaranteed to have entered shutdown. See *Low-Power Shutdown Mode* section.

Typical Operating Characteristics

($V_{CC} = 5V$, $T_A = +25^\circ C$, unless otherwise noted.)



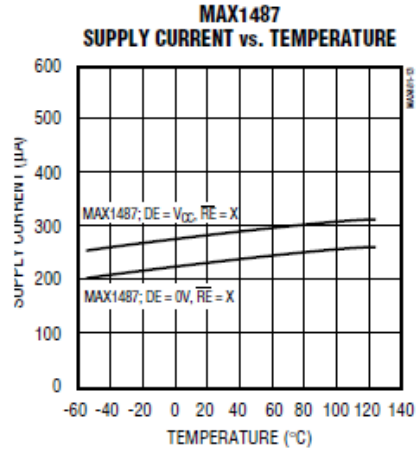
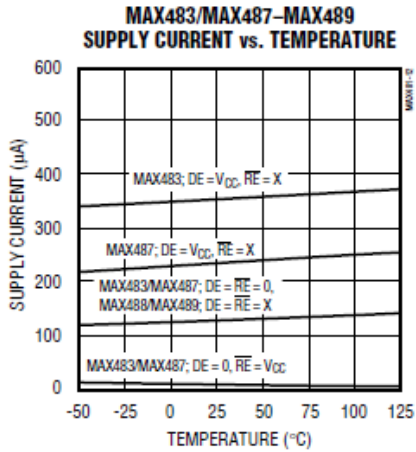
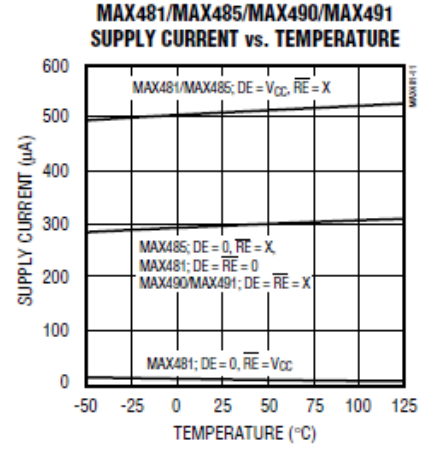
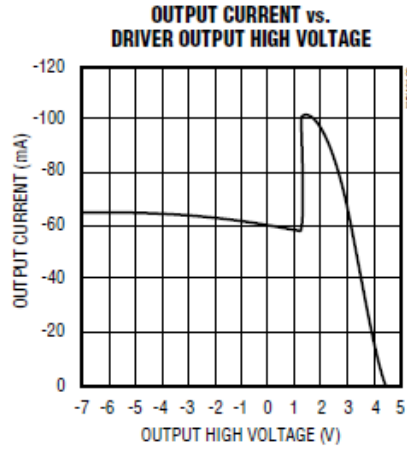
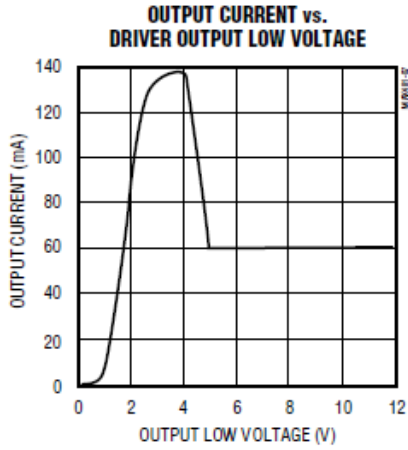
MAXIM

5

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

Typical Operating Characteristics (continued)

($V_{CC} = 5V$, $T_A = +25^\circ C$, unless otherwise noted.)



Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

Pin Description

PIN					NAME	FUNCTION
MAX481/MAX483/ MAX485/MAX487/ MAX1487		MAX488/ MAX490		MAX489/ MAX491		
DIP/SO	μMAX	DIP/SO	μMAX	DIP/SO		
1	3	2	4	2	RO	Receiver Output: If A > B by 200mV, RO will be high; If A < B by 200mV, RO will be low.
2	4	—	—	3	\overline{RE}	Receiver Output Enable. RO is enabled when \overline{RE} is low; RO is high impedance when \overline{RE} is high.
3	5	—	—	4	DE	Driver Output Enable. The driver outputs, Y and Z, are enabled by bringing DE high. They are high impedance when DE is low. If the driver outputs are enabled, the parts function as line drivers. While they are high impedance, they function as line receivers if \overline{RE} is low.
4	6	3	5	5	DI	Driver Input. A low on DI forces output Y low and output Z high. Similarly, a high on DI forces output Y high and output Z low.
5	7	4	6	6, 7	GND	Ground
—	—	5	7	9	Y	Noninverting Driver Output
—	—	6	8	10	Z	Inverting Driver Output
6	8	—	—	—	A	Noninverting Receiver Input and Noninverting Driver Output
—	—	8	2	12	A	Noninverting Receiver Input
7	1	—	—	—	B	Inverting Receiver Input and Inverting Driver Output
—	—	7	1	11	B	Inverting Receiver Input
8	2	1	3	14	V _{CC}	Positive Supply: 4.75V ≤ V _{CC} ≤ 5.25V
—	—	—	—	1, 8, 13	N.C.	No Connect—not internally connected

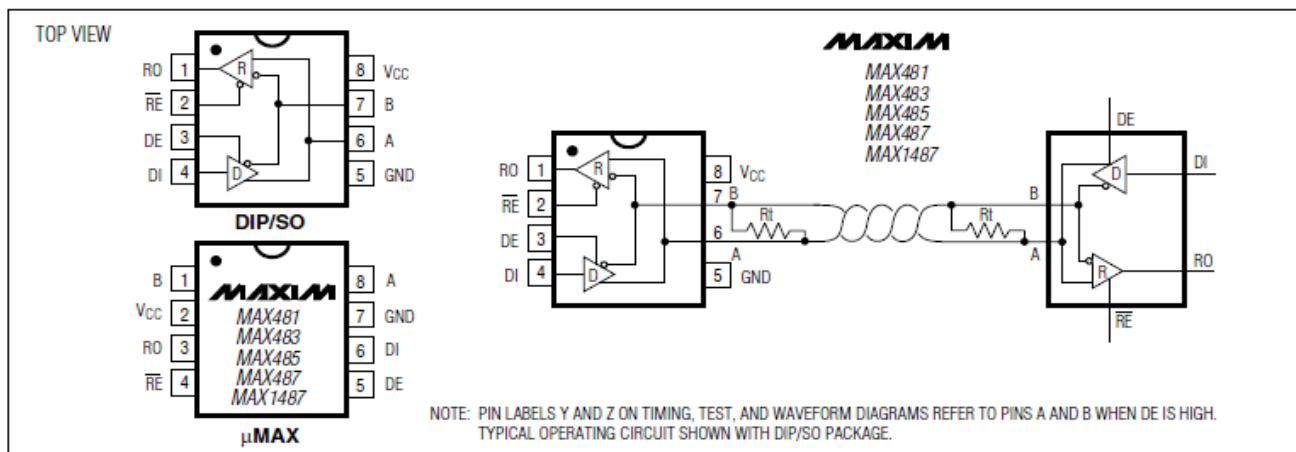


Figure 1. MAX481/MAX483/MAX485/MAX487/MAX1487 Pin Configuration and Typical Operating Circuit

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

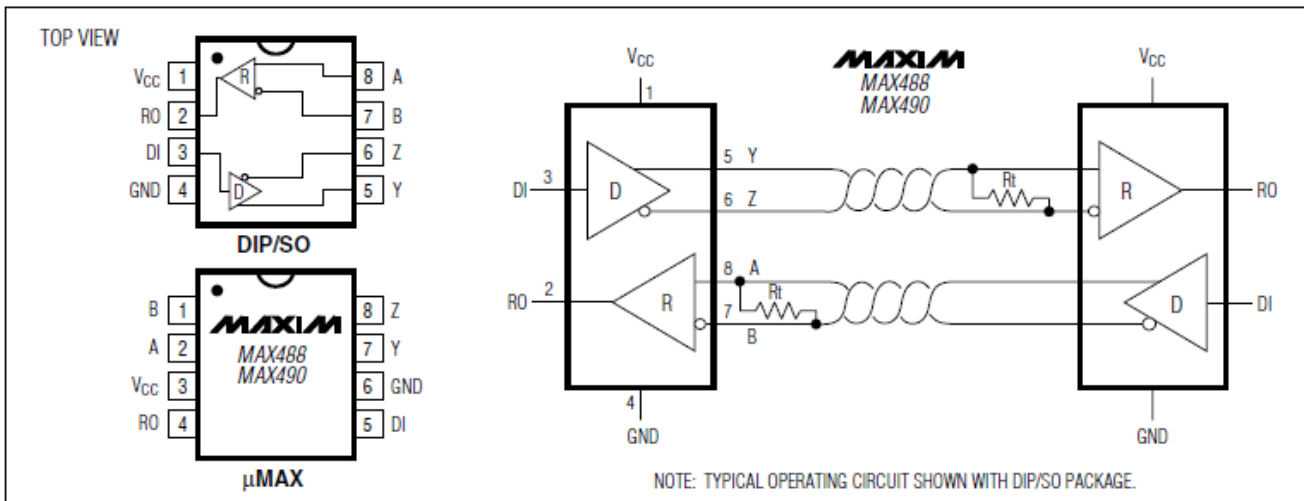


Figure 2. MAX488/MAX490 Pin Configuration and Typical Operating Circuit

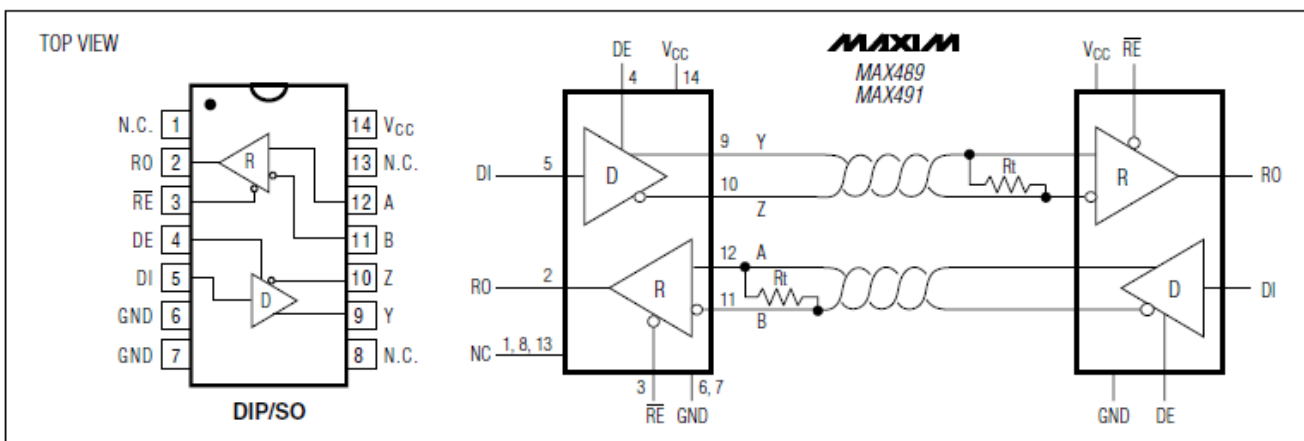


Figure 3. MAX489/MAX491 Pin Configuration and Typical Operating Circuit

Applications Information

The MAX481/MAX483/MAX485/MAX487–MAX491 and MAX1487 are low-power transceivers for RS-485 and RS-422 communications. The MAX481, MAX485, MAX490, MAX491, and MAX1487 can transmit and receive at data rates up to 2.5Mbps, while the MAX483, MAX487, MAX488, and MAX489 are specified for data rates up to 250kbps. The MAX488–MAX491 are full-duplex transceivers while the MAX481, MAX483, MAX485, MAX487, and MAX1487 are half-duplex. In addition, Driver Enable (DE) and Receiver Enable (RE) pins are included on the MAX481, MAX483, MAX485, MAX487, MAX489, MAX491, and MAX1487. When disabled, the driver and receiver outputs are high impedance.

MAX487/MAX1487: 128 Transceivers on the Bus

The 48kΩ, 1/4-unit-load receiver input impedance of the MAX487 and MAX1487 allows up to 128 transceivers on a bus, compared to the 1-unit load (12kΩ input impedance) of standard RS-485 drivers (32 transceivers maximum). Any combination of MAX487/MAX1487 and other RS-485 transceivers with a total of 32 unit loads or less can be put on the bus. The MAX481/MAX483/MAX485 and MAX488–MAX491 have standard 12kΩ Receiver Input impedance.

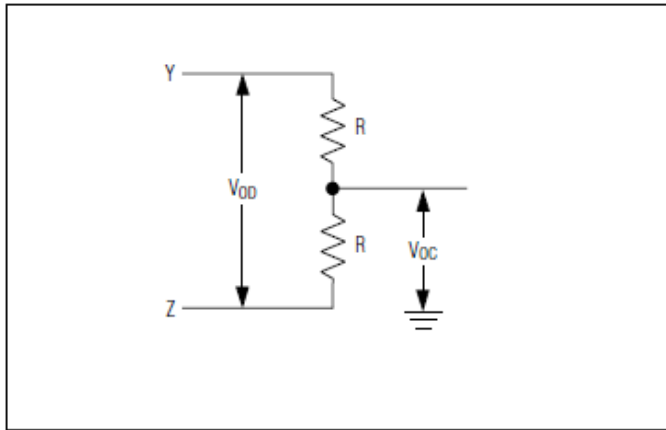


Figure 4. Driver DC Test Load

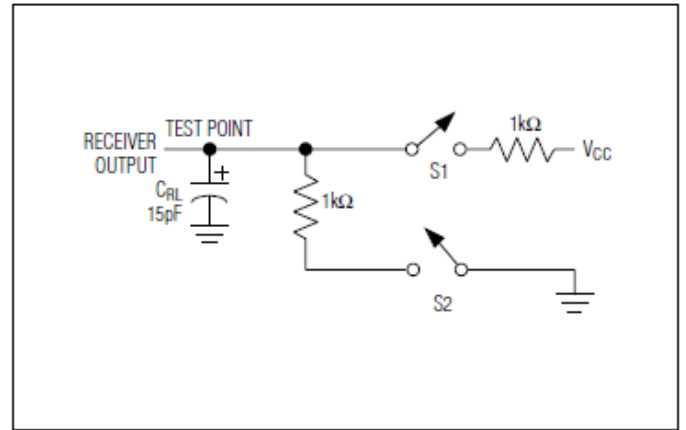


Figure 5. Receiver Timing Test Load

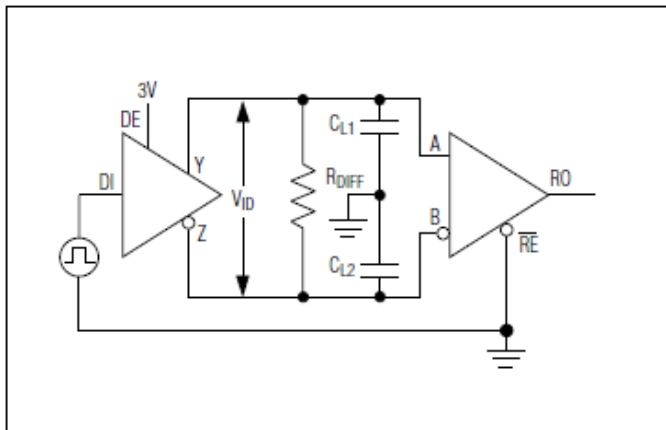


Figure 6. Driver/Receiver Timing Test Circuit

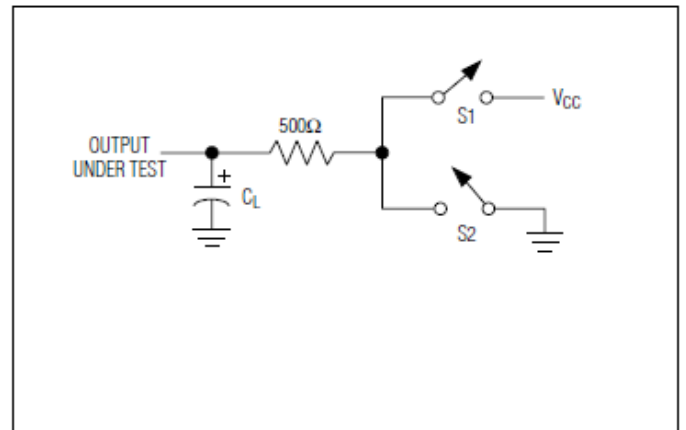


Figure 7. Driver Timing Test Load

**MAX483/MAX487/MAX488/MAX489:
 Reduced EMI and Reflections**

The MAX483 and MAX487–MAX489 are slow-rate limited, minimizing EMI and reducing reflections caused by improperly terminated cables. Figure 12 shows the driver output waveform and its Fourier analysis of a 150kHz signal transmitted by a MAX481, MAX485, MAX490, MAX491, or MAX1487. High-frequency har-

monics with large amplitudes are evident. Figure 13 shows the same information displayed for a MAX483, MAX487, MAX488, or MAX489 transmitting under the same conditions. Figure 13's high-frequency harmonics have much lower amplitudes, and the potential for EMI is significantly reduced.

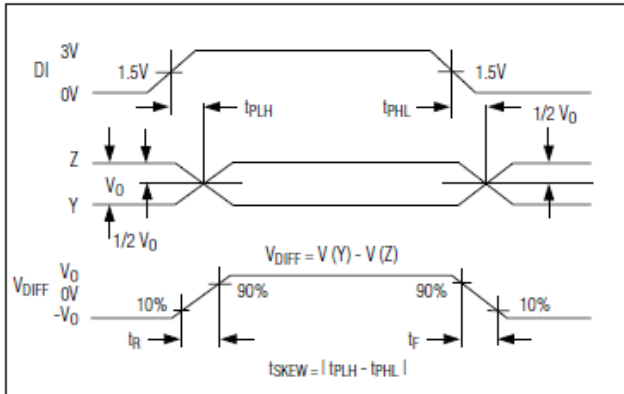


Figure 8. Driver Propagation Delays

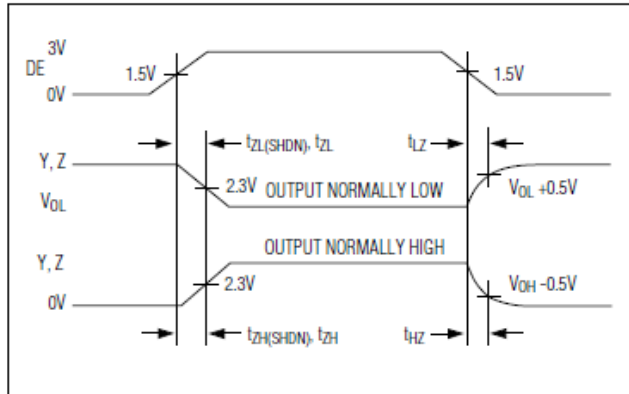


Figure 9. Driver Enable and Disable Times (except MAX488 and MAX490)

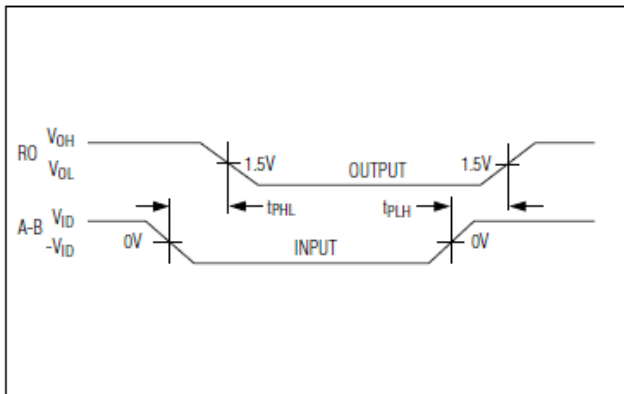


Figure 10. Receiver Propagation Delays

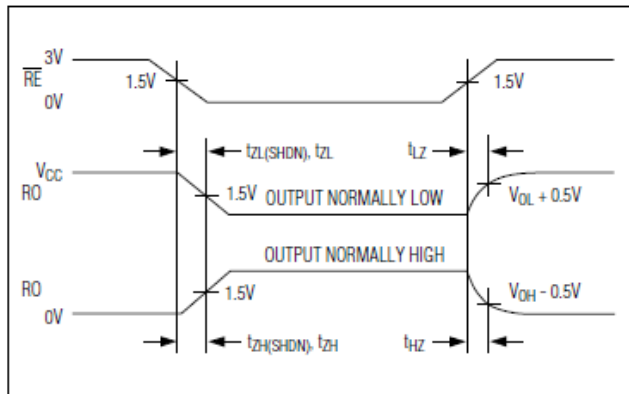


Figure 11. Receiver Enable and Disable Times (except MAX488 and MAX490)

Function Tables (MAX481/MAX483/MAX485/MAX487/MAX1487)

Table 1. Transmitting

INPUTS			OUTPUTS	
RE	DE	DI	Z	Y
X	1	1	0	1
X	1	0	1	0
0	0	X	High-Z	High-Z
1	0	X	High-Z*	High-Z*

X = Don't care
 High-Z = High impedance
 * Shutdown mode for MAX481/MAX483/MAX487

Table 2. Receiving

INPUTS			OUTPUT
RE	DE	A-B	RO
0	0	$\geq +0.2V$	1
0	0	$\leq -0.2V$	0
0	0	Inputs open	1
1	0	X	High-Z*

X = Don't care
 High-Z = High impedance
 * Shutdown mode for MAX481/MAX483/MAX487

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

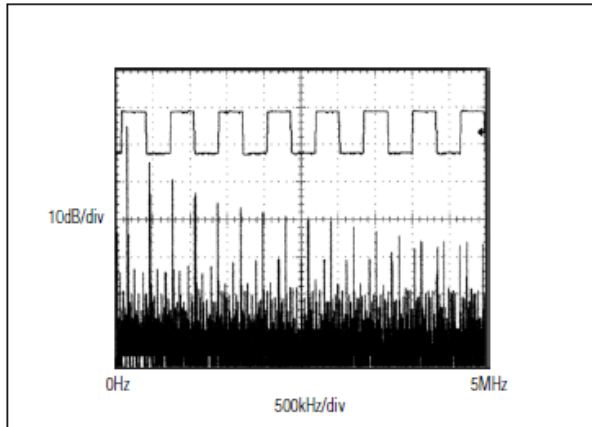


Figure 12. Driver Output Waveform and FFT Plot of MAX481/MAX485/MAX490/MAX491/MAX1487 Transmitting a 150kHz Signal

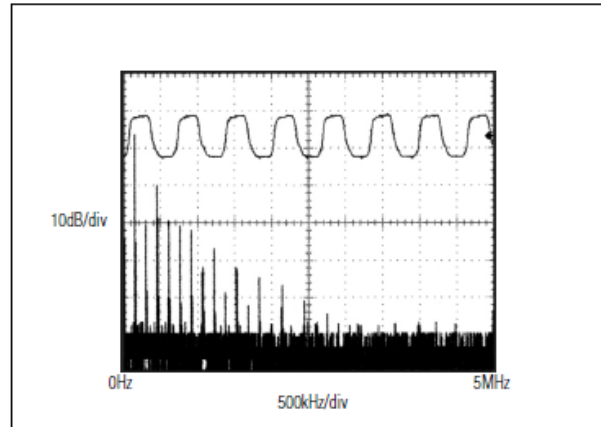


Figure 13. Driver Output Waveform and FFT Plot of MAX483/MAX487-MAX489 Transmitting a 150kHz Signal

Low-Power Shutdown Mode (MAX481/MAX483/MAX487)

A low-power shutdown mode is initiated by bringing both \overline{RE} high and DE low. The devices will not shut down unless both the driver and receiver are disabled. In shutdown, the devices typically draw only 0.1 μ A of supply current.

\overline{RE} and DE may be driven simultaneously; the parts are guaranteed not to enter shutdown if \overline{RE} is high and DE is low for less than 50ns. If the inputs are in this state for at least 600ns, the parts are guaranteed to enter shutdown.

For the MAX481, MAX483, and MAX487, the t_{ZH} and t_{ZL} enable times assume the part was not in the low-power shutdown state (the MAX485/MAX488-MAX491 and MAX1487 can not be shut down). The $t_{ZH}(SHDN)$ and $t_{ZL}(SHDN)$ enable times assume the parts were shut down (see *Electrical Characteristics*).

It takes the drivers and receivers longer to become enabled from the low-power shutdown state ($t_{ZH}(SHDN)$, $t_{ZL}(SHDN)$) than from the operating mode (t_{ZH} , t_{ZL}). (The parts are in operating mode if the \overline{RE} , DE inputs equal a logical 0, 1 or 1, 1 or 0, 0.)

Driver Output Protection

Excessive output current and power dissipation caused by faults or by bus contention are prevented by two mechanisms. A foldback current limit on the output stage provides immediate protection against short circuits over the whole common-mode voltage range (see *Typical Operating Characteristics*). In addition, a thermal shutdown circuit forces the driver outputs into a high-impedance state if the die temperature rises excessively.

Propagation Delay

Many digital encoding schemes depend on the difference between the driver and receiver propagation delay times. Typical propagation delays are shown in Figures 15–18 using Figure 14's test circuit.

The difference in receiver delay times, $|t_{PLH} - t_{PHL}|$, is typically under 13ns for the MAX481, MAX485, MAX490, MAX491, and MAX1487 and is typically less than 100ns for the MAX483 and MAX487-MAX489.

The driver skew times are typically 5ns (10ns max) for the MAX481, MAX485, MAX490, MAX491, and MAX1487, and are typically 100ns (800ns max) for the MAX483 and MAX487-MAX489.

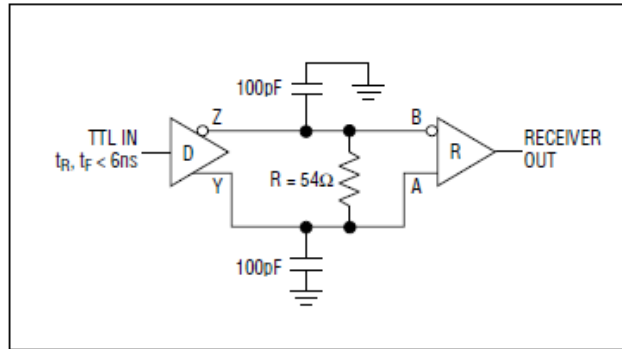


Figure 14. Receiver Propagation Delay Test Circuit

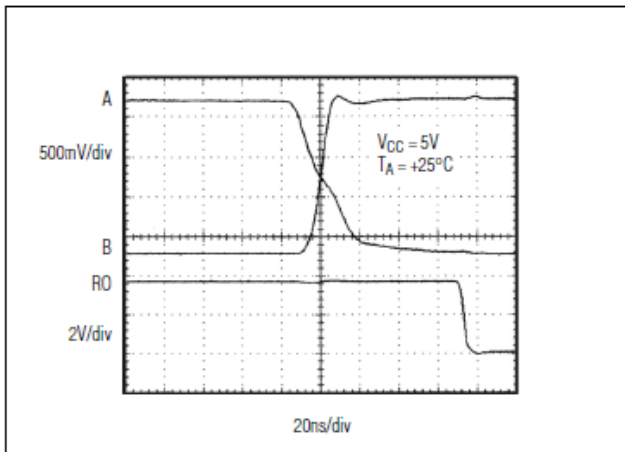


Figure 15. MAX481/MAX485/MAX490/MAX491/MAX1487 Receiver tPHL

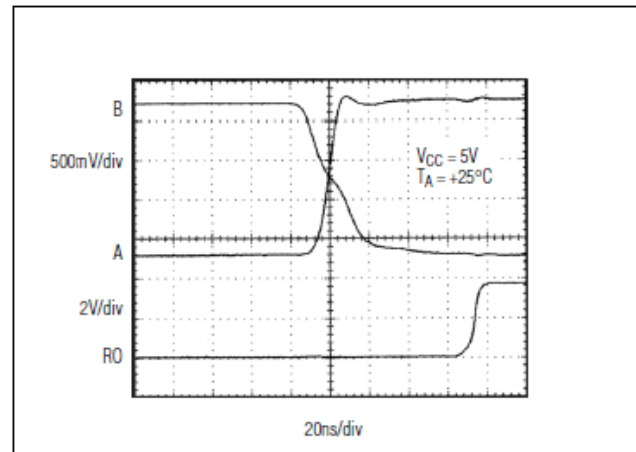


Figure 16. MAX481/MAX485/MAX490/MAX491/MAX1487 Receiver tPLH

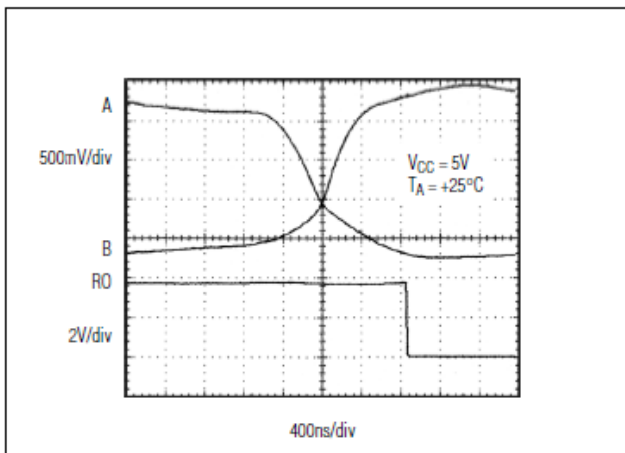


Figure 17. MAX483, MAX487–MAX489 Receiver tPHL

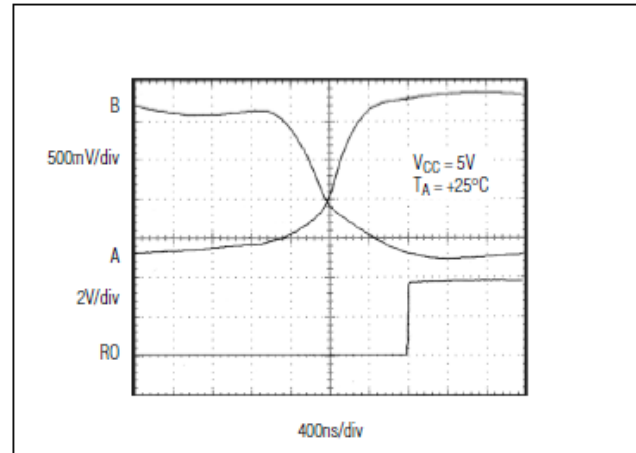


Figure 18. MAX483, MAX487–MAX489 Receiver tPLH

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

Line Length vs. Data Rate

The RS-485/RS-422 standard covers line lengths up to 4000 feet. For line lengths greater than 4000 feet, see Figure 23.

Figures 19 and 20 show the system differential voltage for the parts driving 4000 feet of 26AWG twisted-pair wire at 110kHz into 120Ω loads.

Typical Applications

The MAX481, MAX483, MAX485, MAX487–MAX491, and MAX1487 transceivers are designed for bidirectional data communications on multipoint bus transmission lines.

Figures 21 and 22 show typical network applications circuits. These parts can also be used as line repeaters, with cable lengths longer than 4000 feet, as shown in Figure 23.

To minimize reflections, the line should be terminated at both ends in its characteristic impedance, and stub lengths off the main line should be kept as short as possible. The slew-rate-limited MAX483 and MAX487–MAX489 are more tolerant of imperfect termination.

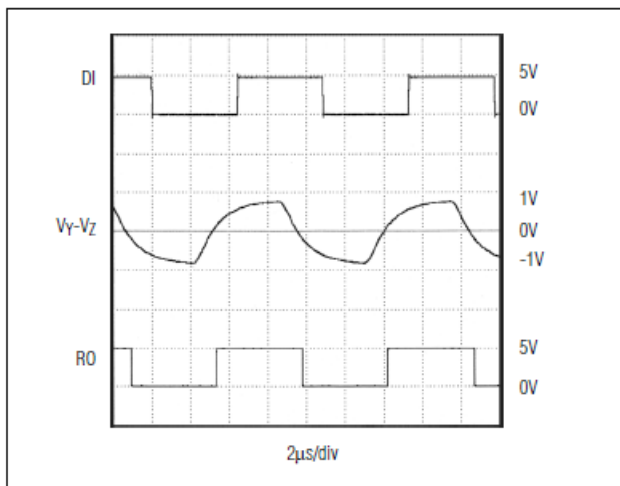


Figure 19. MAX481/MAX485/MAX490/MAX491/MAX1487 System Differential Voltage at 110kHz Driving 4000ft of Cable

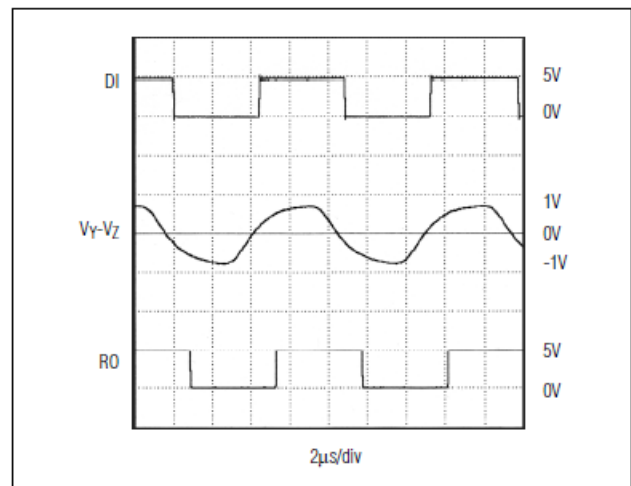


Figure 20. MAX483, MAX487–MAX489 System Differential Voltage at 110kHz Driving 4000ft of Cable

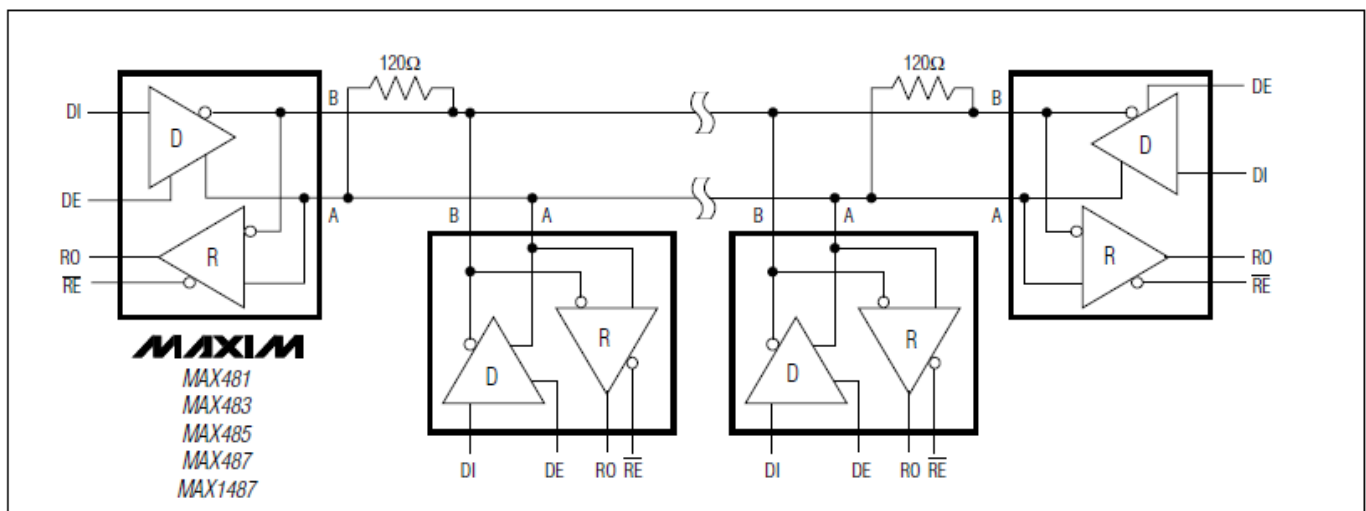


Figure 21. MAX481/MAX483/MAX485/MAX487/MAX1487 Typical Half-Duplex RS-485 Network

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

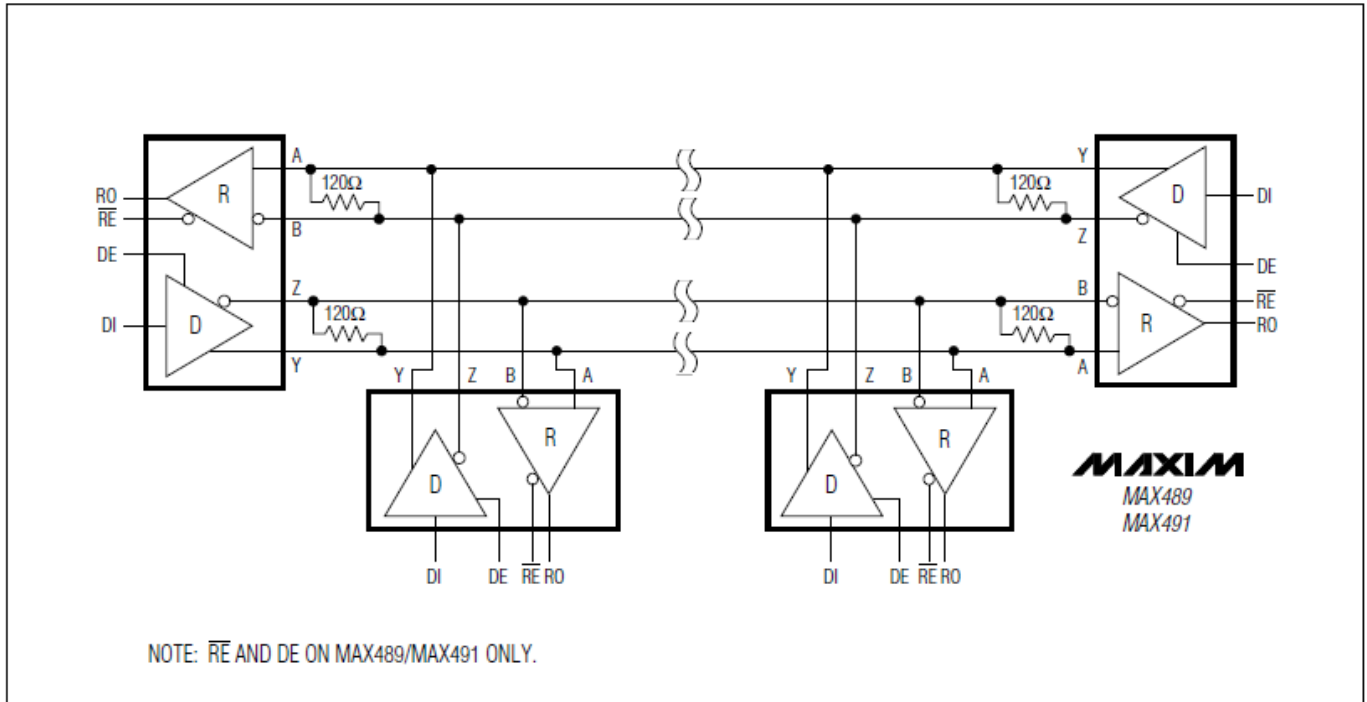


Figure 22. MAX488-MAX491 Full-Duplex RS-485 Network

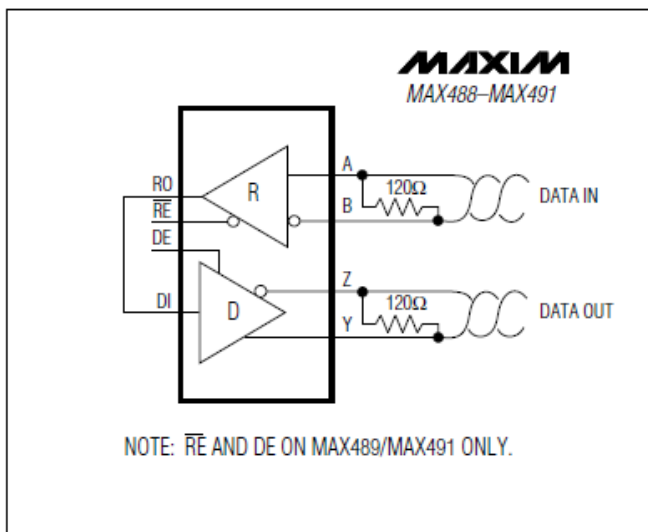


Figure 23. Line Repeater for MAX488-MAX491

Isolated RS-485

For isolated RS-485 applications, see the MAX253 and MAX1480 data sheets.

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

Ordering Information (continued)

PART	TEMP. RANGE	PIN-PACKAGE
MAX481EPA	-40°C to +85°C	8 Plastic DIP
MAX481ESA	-40°C to +85°C	8 SO
MAX481MJA	-55°C to +125°C	8 CERDIP
MAX483 CPA	0°C to +70°C	8 Plastic DIP
MAX483CSA	0°C to +70°C	8 SO
MAX483CUA	0°C to +70°C	8 μMAX
MAX483C/D	0°C to +70°C	Dice*
MAX483EPA	-40°C to +85°C	8 Plastic DIP
MAX483ESA	-40°C to +85°C	8 SO
MAX483MJA	-55°C to +125°C	8 CERDIP
MAX485 CPA	0°C to +70°C	8 Plastic DIP
MAX485CSA	0°C to +70°C	8 SO
MAX485CUA	0°C to +70°C	8 μMAX
MAX485C/D	0°C to +70°C	Dice*
MAX485EPA	-40°C to +85°C	8 Plastic DIP
MAX485ESA	-40°C to +85°C	8 SO
MAX485MJA	-55°C to +125°C	8 CERDIP
MAX487 CPA	0°C to +70°C	8 Plastic DIP
MAX487CSA	0°C to +70°C	8 SO
MAX487CUA	0°C to +70°C	8 μMAX
MAX487C/D	0°C to +70°C	Dice*
MAX487EPA	-40°C to +85°C	8 Plastic DIP
MAX487ESA	-40°C to +85°C	8 SO
MAX487MJA	-55°C to +125°C	8 CERDIP
MAX488 CPA	0°C to +70°C	8 Plastic DIP
MAX488CSA	0°C to +70°C	8 SO
MAX488CUA	0°C to +70°C	8 μMAX
MAX488C/D	0°C to +70°C	Dice*
MAX488EPA	-40°C to +85°C	8 Plastic DIP
MAX488ESA	-40°C to +85°C	8 SO
MAX488MJA	-55°C to +125°C	8 CERDIP
MAX489 CPD	0°C to +70°C	14 Plastic DIP
MAX489CSD	0°C to +70°C	14 SO
MAX489C/D	0°C to +70°C	Dice*
MAX489EPD	-40°C to +85°C	14 Plastic DIP
MAX489ESD	-40°C to +85°C	14 SO
MAX489MJD	-55°C to +125°C	14 CERDIP

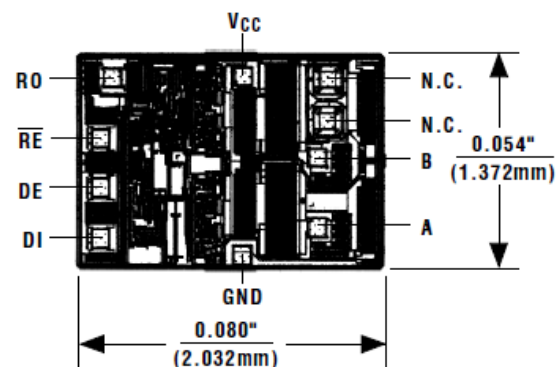
Ordering Information (continued)

PART	TEMP. RANGE	PIN-PACKAGE
MAX490 CPA	0°C to +70°C	8 Plastic DIP
MAX490CSA	0°C to +70°C	8 SO
MAX490CUA	0°C to +70°C	8 μMAX
MAX490C/D	0°C to +70°C	Dice*
MAX490EPA	-40°C to +85°C	8 Plastic DIP
MAX490ESA	-40°C to +85°C	8 SO
MAX490MJA	-55°C to +125°C	8 CERDIP
MAX491 CPD	0°C to +70°C	14 Plastic DIP
MAX491CSD	0°C to +70°C	14 SO
MAX491C/D	0°C to +70°C	Dice*
MAX491EPD	-40°C to +85°C	14 Plastic DIP
MAX491ESD	-40°C to +85°C	14 SO
MAX491MJD	-55°C to +125°C	14 CERDIP
MAX1487 CPA	0°C to +70°C	8 Plastic DIP
MAX1487CSA	0°C to +70°C	8 SO
MAX1487CUA	0°C to +70°C	8 μMAX
MAX1487C/D	0°C to +70°C	Dice*
MAX1487EPA	-40°C to +85°C	8 Plastic DIP
MAX1487ESA	-40°C to +85°C	8 SO
MAX1487MJA	-55°C to +125°C	8 CERDIP

* Contact factory for dice specifications.

Chip Topographies

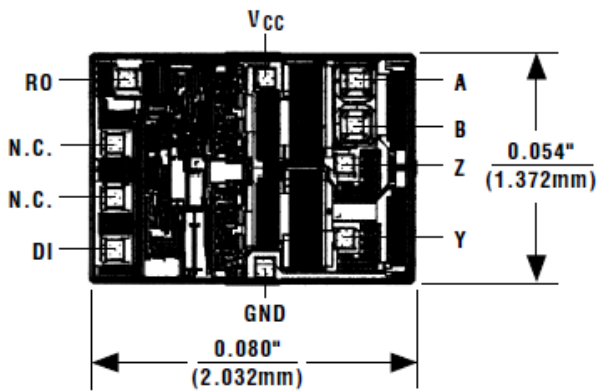
MAX481/MAX483/MAX485/MAX487/MAX1487



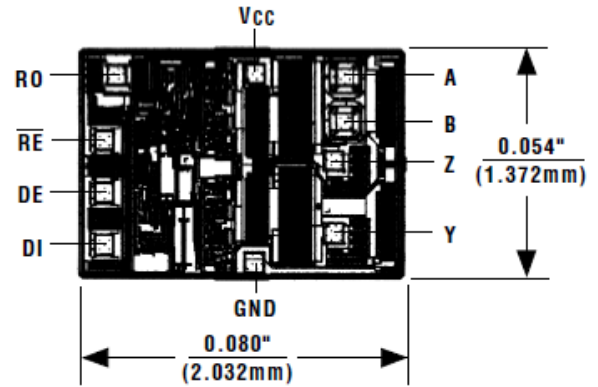
Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

Chip Topographies (continued)

MAX488/MAX490



MAX489/MAX491



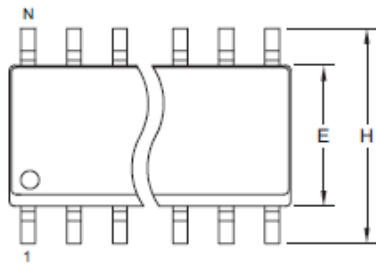
TRANSISTOR COUNT: 248
 SUBSTRATE CONNECTED TO GND

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

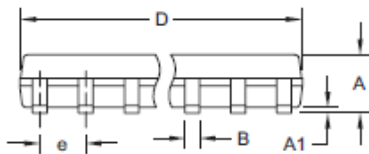
Package Information

(The package drawing(s) in this data sheet may not reflect the most current specifications. For the latest package outline information go to www.maxim-ic.com/packages.)

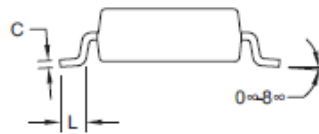
SOICN .15°S



TOP VIEW



FRONT VIEW



SIDE VIEW

NOTES:

1. D&E DO NOT INCLUDE MOLD FLASH.
2. MOLD FLASH OR PROTRUSIONS NOT TO EXCEED 0.15mm (.006").
3. LEADS TO BE COPLANAR WITHIN 0.10mm (.004").
4. CONTROLLING DIMENSION: MILLIMETERS.
5. MEETS JEDEC MS012.
6. N = NUMBER OF PINS.

DIM	INCHES		MILLIMETERS	
	MIN	MAX	MIN	MAX
A	0.053	0.069	1.35	1.75
A1	0.004	0.010	0.10	0.25
B	0.014	0.019	0.35	0.49
C	0.007	0.010	0.19	0.25
e	0.050 BSC		1.27 BSC	
E	0.150	0.157	3.80	4.00
H	0.228	0.244	5.80	6.20
L	0.016	0.050	0.40	1.27

VARIATIONS:

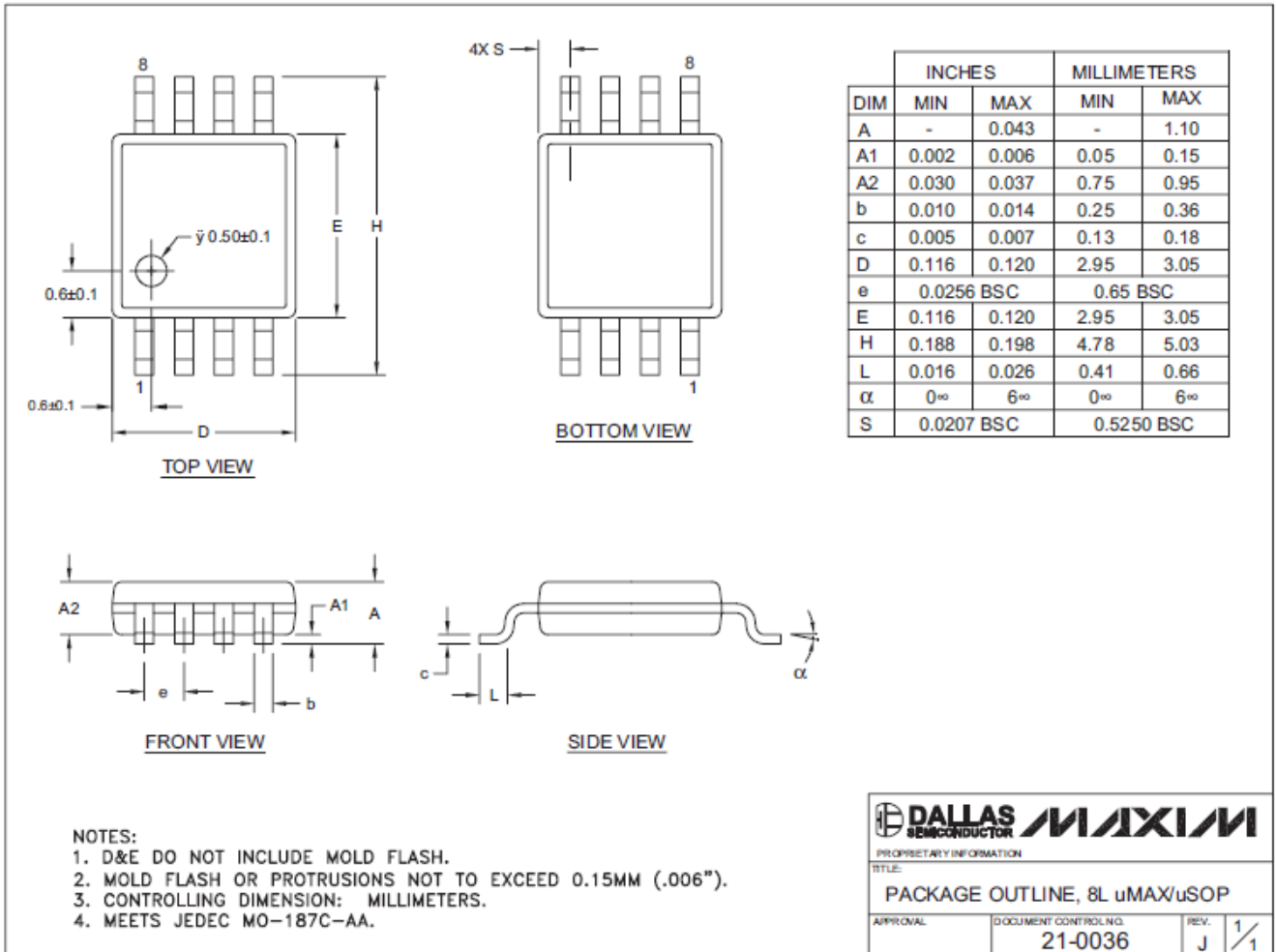
DIM	INCHES		MILLIMETERS		N	MS012
	MIN	MAX	MIN	MAX		
D	0.189	0.197	4.80	5.00	8	AA
D	0.337	0.344	8.55	8.75	14	AB
D	0.386	0.394	9.80	10.00	16	AC

PROPRIETARY INFORMATION			
TITLE: PACKAGE OUTLINE, .150° SOIC			
APPROVAL	DOCUMENT CONTROL NO. 21-0041	REV. B	1/1

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

Package Information (continued)

(The package drawing(s) in this data sheet may not reflect the most current specifications. For the latest package outline information go to www.maxim-ic.com/packages.)

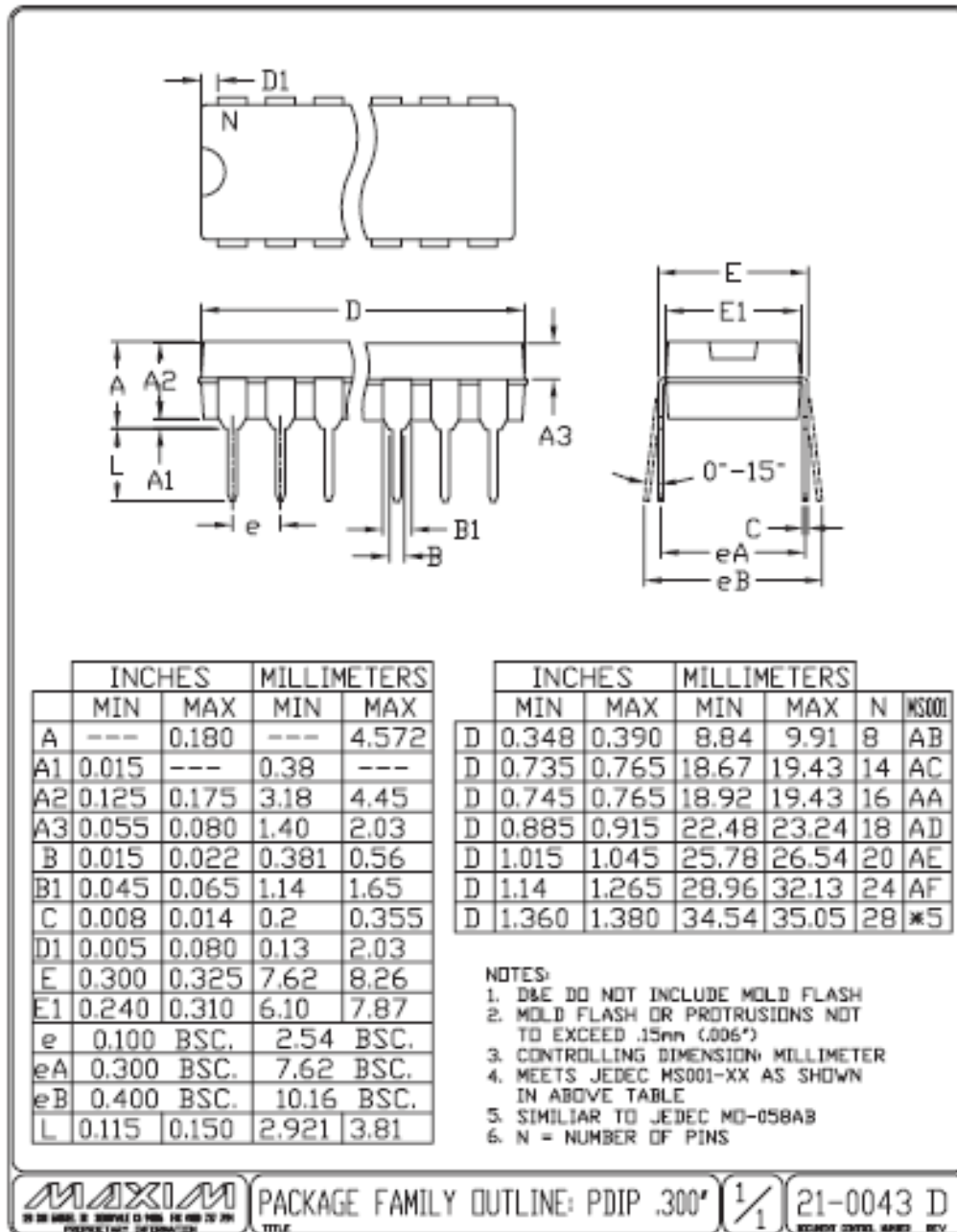


8LUMAXD.EPS

Low-Power, Slew-Rate-Limited RS-485/RS-422 Transceivers

Package Information (continued)

(The package drawing(s) in this data sheet may not reflect the most current specifications. For the latest package outline information go to www.maxim-lc.com/packages.)



Maxim cannot assume responsibility for use of any circuitry other than circuitry entirely embodied in a Maxim product. No circuit patent licenses are implied. Maxim reserves the right to change the circuitry and specifications without notice at any time.

Maxim Integrated Products, 120 San Gabriel Drive, Sunnyvale, CA 94086 408-737-7600 19



© 2003 Maxim Integrated Products Printed USA **MAXIM** is a registered trademark of Maxim Integrated Products.

• 3.4.-DATASHEET CONVERTOR RS232/RS485 B&B

Electronics

Although the 485SD9TB uses handshake lines to power the converter, no handshaking is required to control the RS-485 driver. The RS-485 driver is automatically enabled during each spacing state on the RS-232 side. During the marking or idle state, the RS-485 driver is disabled and the data lines are held in the marking state by the 4.7K ohm pull-up and pull-down resistors. The value of these resistors may need to be changed to a different value when termination is used in order to maintain the proper DC bias during the idle state. See B&B Electronics' RS-422/RS-485 Application Note for more information on termination and DC biasing of an RS-485 network.

The 485SD9TB has an internal connection to prevent data transmitted from the RS-232 port from being echoed back to the RS-232 port. The 485SD9TB is used as a two wire (half duplex) RS-485 converter.

DECLARATION OF CONFORMITY	
Manufacturer's Name:	B&B Electronics Manufacturing Company
Manufacturer's Address:	P.O. Box 1040 707 Dayton Road Ottawa, IL 61350 USA
Model Number:	485SD9TB
Description:	Port-Powered RS-485 Converter
Type:	Light industrial ITE equipment
Application of Council Directive:	89/336/EEC
Standards:	EN 50082-1 (IEC 801-2, IEC 801-3, IEC 801-4) EN 50081-1 (EN 55022, IEC 1000-4-2) EN 61000 (-4-2, -4-3, -4-4, -4-5, -4-6, -4-8, -4-11) ENV 50204 EN 55024
 Michael J. Fabron, Director of Engineering	
	

© B&B Electronics -- Revised November 2000

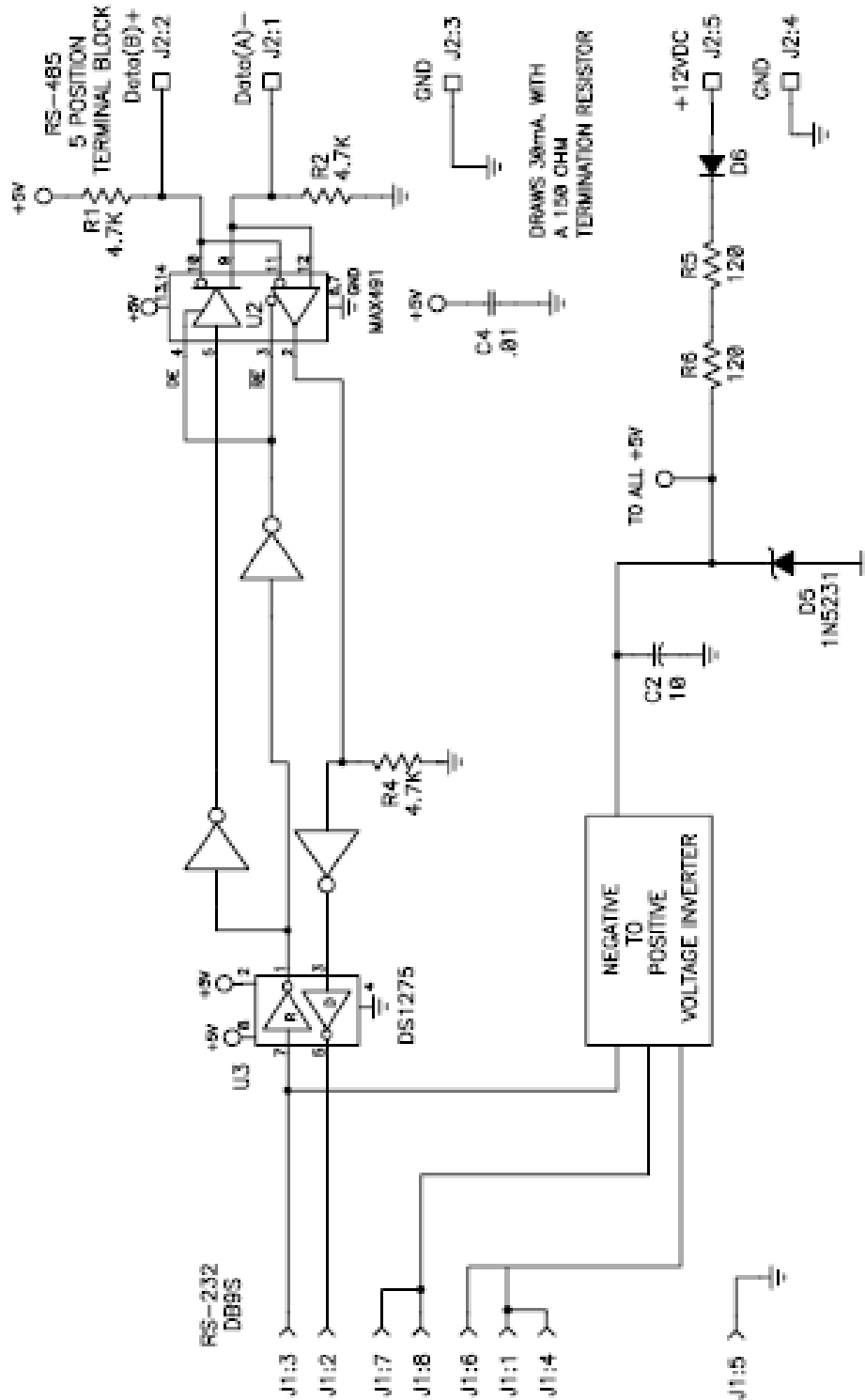
This product designed and manufactured in USA of domestic and imported parts by

B & B Electronics Mfg. Co.

707 Dayton Road - P.O. Box 1040 - Ottawa, IL 61350 USA
 Phone: (815) 433-5100 - Fax: (815) 433-5105
 Home Page: www.bb-elec.com
 Sales e-mail: orders@bb-elec.com - Fax: (815) 433-5109
 Technical Support e-mail: support@bb-elec.com - Fax: (815) 433-5104

B & B Electronics Ltd

Westlink Commercial Park - Oranmore, Co. Galway - Ireland
 Phone: +353 91 792444 - Fax: +353 91 792445
 Home Page: www.bb-europe.com
 Sales e-mail: orders@bb-europe.com
 Technical Support e-mail: support@bb-europe.com



© B&B Electronics - Revised November 2000

This product designed and manufactured in USA of domestic and imported parts by

B&B Electronics Mfg. Co.

707 Dayton Road - P.O. Box 1048 - Ottawa, IL 61320 USA
 Phone: (815) 433-5100 - Fax: (815) 433-5165
 Home Page: www.bb-elec.com
 Sales e-mail: orders@bb-elec.com - Fax: (815) 433-5100
 Technical Support e-mail: support@bb-elec.com - Fax: (815) 433-5104

B&B Electronics Ltd

Westlink Commercial Park - Draxmore, Co. Galway - Ireland
 Phone: +353 91 732444 - Fax: +353 91 732445
 Home Page: www.bb-europe.com
 Sales e-mail: orders@bb-europe.com
 Technical Support e-mail: support@bb-europe.com

• 3.5.-DATASHEET REGULADOR GEFRAN 800P:

GEFRAN

800P

PROGRAMADOR/REGULADOR



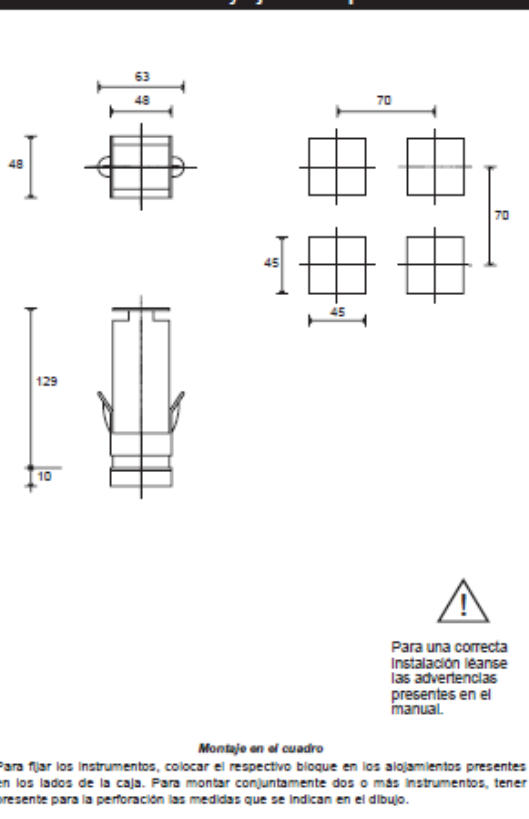
MANUAL DE USO

VERSIÓN SOFTWARE 3.2x
 Código 80215E / Edición 13 - 06/09



1 • INSTALACION

- Dimensiones exteriores y de perforación; colocación y fijación al panel



MARCA CE. El instrumento reúne los requisitos de las Directivas de la Unión Europea 2004/109/CE y 2006/95/CE con referencia a las normas EN 61000-6-2 (Inmunidad en ambientes industriales) EN 61000-6-3 (emisión en ambientes residenciales) EN 61010-1 (seguridad).

MANTENIMIENTO. Las reparaciones deben ser efectuadas sólo por personal especializado o debidamente capacitado. Interrumpir la alimentación al instrumento antes de intervenir en sus partes internas.

No limpiar la caja con disolventes derivados de hidrocarburos (trielina, bencina, etc.). El uso de dichos disolventes afectará a la fiabilidad mecánica del instrumento. Para limpiar las partes externas de plástico, utilizar un paño limpio humedecido con alcohol etílico o con agua.

ASISTENCIA TÉCNICA. El departamento de asistencia técnica GEFRAN se encuentra a disposición del cliente. Quedan excluidos de la garantía los desperfectos derivados de un uso no conforme con las instrucciones de empleo.

2 • CARACTERÍSTICAS TÉCNICAS

Visualizador	2 x 4 dígitos color verde, altura cifras 10 y 7 mm
Teclas	4 de tipo mecánico (Man/Aut, INC, DEC, F)
Precisión	0,2% p.e. a temperatura ambiente de 25 °C
Entrada principal	TC, RTD (PT100 - JPT100), PTC, 50mV RI ≥ 1MΩ; 10V RI ≥ 10KΩ; 20mA, RI = 50Ω
Termopares	IEC 584-1 (J, K, R, S, T, B, E, N, NI-NI18Mo, L, NI-Cr-CuNi)
Error comp. unión fría	0,1° / °C
Tipo RTD (escala configurable en el rango indicado, con o sin coma decimal)	DIN 43760 (PT100, JPT100)
Tipo PTC (bajo pedido)	990Ω, 25°C
Máx. resistencia de línea para RTD	20Ω
Seguridad	detección cortocircuito o apertura de las sondas, alarma LBA, alarma HB
Selección grados C / F	configurable desde teclado
Rango escalas lineales	-1999 ... 9999 con coma decimal configurable
Acciones de control	PID, Auto-tune, on-off
pb / dt / dl	0.0 ... 999.9% / 0.00 ... 99.99min / 0.00 ... 99.99min
Acciones	calor / frío
Salidas de control	on / off, pwm
Tiempo del ciclo	0.1 ... 200 sec
Tipo de salida principal	relé, lógica, continua (opción)
Softstart	0.0 ... 500.0 min
Limitación máx. potencia calor / frío	0.0 ... 100.0 %
Ajuste potencia de fallo	-100.0 ... 100.0 %
Función de apagado	mantiene la visualiz. de PV, posibilidad de exclusión
Alarmas configurables	tres alarmas configurables de tipo: máxima, mínima, simétricas, absolutas/relativas, LBA, HB
Enmascaramiento alarmas	- exclusión del encendido - reset memoria desde teclado y/o contacto externo
Tipo de contacto relé	NO (NC), 5A, 250V, cosφ = 1
Salida lógica para relés estáticos	11Vdc, Rout = 220Ω (5V/20mA)
(Opción) Setpoint remoto o entrada amperimétrica	0 ... 10V, 2 ... 10V, RI ≥ 1MΩ 0 ... 20mA, 4 ... 20mA, RI = 5Ω Potenciómetro > 500Ω, TA 50mAac, 50/60Hz, RI = 1,5Ω, aislamiento 1500V
Rango escala TA	Configurable 0, ... , 100.0A
(Opción) Alimentación para transmisor	10 / 24Vcc 10/24 Vcc filtrada, máx. 30 mA protección cortocircuito, aislamiento 1500 V
(Opción) Retransmisión analógica	10V / 20mA, aislamiento 1500V
(Opción) Entradas lógicas	24V NPN, 4.5mA; 24V PNP, 3.5mA aislamiento 1500V
(Opción) interfaz serie	CL; RS422/485; RS232; aislamiento 1500V
Velocidad transmisión	1200 ... 19200 baudios
Protocolo	GEFRAN / MODBUS
Alimentación (conmutada)	(estándar) 100...240V ac/dc ±10%; 50/60Hz, 12VA max (opcional) 20...27V ac/dc ±10%; 50/60 Hz, 12VA max
Protección frontal	IP65
Temperatura de trabajo/almacenamiento	0...50°C / -20...70°C
Humedad relativa	20 ... 85% H.R. sin condensaciones
Condiciones ambientales del uso	para el uso interno, altitud hasta los 2000m
Instalación	en panel de extracción frontal
Peso	210 g en versión completa

La conformidad de EMC ha sido verificada con las siguientes conexiones

FUNCION	TIPO DE CABLE	LONGITUD UTILIZADA
Cable de alimentación	1 mm ²	1 mt
Hilos salida relé	1 mm ²	3,5 mt
Hilos de conexión serie	0,35 mm ²	3,5 mt
Hilos de conexión T.A.	1,5 mm ²	3,5 mt
Sonda entrada termopar	0,8 mm ² compensado	5 mt
Sonda entrada termoresistencia "PT100"	1 mm ²	3 mt

80215E_MHW_800P_0609_ESP

1

3 • DESCRIPCIÓN PARTE FRONTAL INSTRUMENTO

Indicadores de función:
 Señalan el tipo de funcionamiento del instrumento
 MAN = OFF (regulación automática)
 MAN = ON (regulación manual)
 AUX = ON (programa en reset)
 PRG = ON (programa en ejecución)

Selección regulación Automática/Manual:
 Activo sólo en visualización nivel 1

Teclas "Incrementa" y "Decrementa"

Permiten realizar una operación de aumento (reducción) de cualquier parámetro numérico -- La velocidad de aumento (reducción) es proporcional a la duración de la presión sobre la tecla -- La operación no es cíclica, por lo que una vez alcanzado el máx. (mín.) de un campo de aplicación, incluso manteniendo presionada la tecla, la función de aumento (reducción) queda bloqueada.

Indicación estado de las salidas
 OUT 1 (Main); OUT 2 (AL 1);
 OUT 3 (AL 2); OUT 4 (HB)

Visualizador PV: Indicación de la variable del proceso
 Visualización de errores: LO, HI, Sbr, Err
 LO = el valor de la variable del proceso es < LO, S
 HI = el valor de la variable del proceso es > HI, S
 Sbr = sonda interrumpida o valores de la entrada superando límites máximos
 Err = tensor hilo PT100 interrumpido, PTC o valores de la entrada inferiores a los límites mínimos (por el TC con conexión errónea)

Visualizador SV: Indicación Setpoint de regulación

Tecle función:
 Permite el acceso a las diferentes fases de configuración --
 Confirma la modificación de los parámetros asignados, con paso al parámetro siguiente o al precedente si la tecla Auto/Man está presionada.

4 • CONEXIONES

• Línea serie

Línea serie aislada 1500 V configurable. Current Loop pasiva (máx. 1200 baudios)
 RS422/485 o RS232 bajo pedido

• Salidas

Salidas de uso genérico configurables por el usuario

Out2 (AL1)
 + (20) - (19)

Out1 (Main)
 + (22) - (21)

• Alimentación

PWR
 ~ (23) - (24) ~

Estándar: 100...240Vac/Vdc
Opcional: 20...27Vac/Vdc
 50/60Hz

• Entrada auxiliar

Entrada auxiliar aislada 1500V
 transf. amperimétrico 50mAac, 1,5Ω, 50/60Hz

setpoint remoto
 0...20, 4...20mA, 5Ω
 0...1V, 0...10V, > 1MΩ

~ (14) - (13) ~

• Alimentación transmisor

GND (11) Alimentación transmisor aislada 1500 V
 + Vt (12) 10/24 Vcc, máx. 30 mA protección cortocircuito

• Salidas

Salida de uso genérico configurable por el usuario

Out4 (AL3/HB) (W1)
 - (7) + (8)

Salida de uso genérico configurable por el usuario

Out3 (AL2)
 - (6) + (5)

- relé 5A/250Vac, cosφ=1
 - lógica 11 Vcc, Rout = 220Ω (6V/20mA)
 - analógica aislada 1500V (0...10V, 0...20/4...20mA)

• Entradas

• PH100 2 hilos o PTC
 Utilizar hilos de sección adecuada (mín. 1mm²) PT100, JPT100, PTC

• TC (Termopar)

Termopares disponibles: J, K, R, S, T, B, E, N, Ni-Ni 18Mo, L NiCr-CuNi
 - Respetar las polaridades
 - Para extensiones, usar cable compensado adecuado para el tipo de termopar utilizado

• Lineal (V)

Entrada lineal en tensión continua
 0...50mV, 10...50mV, 0...10V, 2...10V

• Lineare (I)

Entrada lineal en corriente continua
 0...20mA, 4...20mA

• Entradas digitales / Out 5

Out 5 analógica (W2) (alternativa a la entrada digital IN2)
 + (9) - (7) COM (*)

(*) borne 11 en el caso de Out 4 de tipo relé o lógica

Entradas digitales aisladas 1500 V
 - NPN 24V, 4,5mA
 - PNP 24V, 3,6mA (12V, 3,6mA)
 (IN2 en alternativa Out 5)

(9) IN2
 (10) IN1
 (11) COM

5 • CABLEADO SUGERIDO

Línea alimentación y salidas

Canal [C] [D] [B] [A]

A Entradas
 B Serie
 C Salidas Relé
 D Alimentación
 E Salidas Lógicas/Analógicas

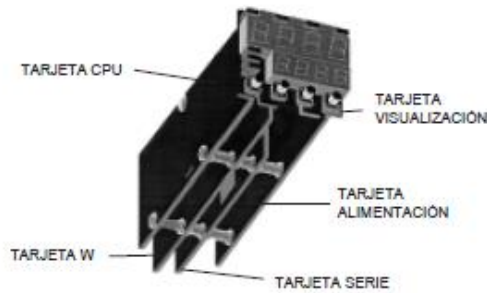
Línea de alimentación

Canal [D] [B] [A]

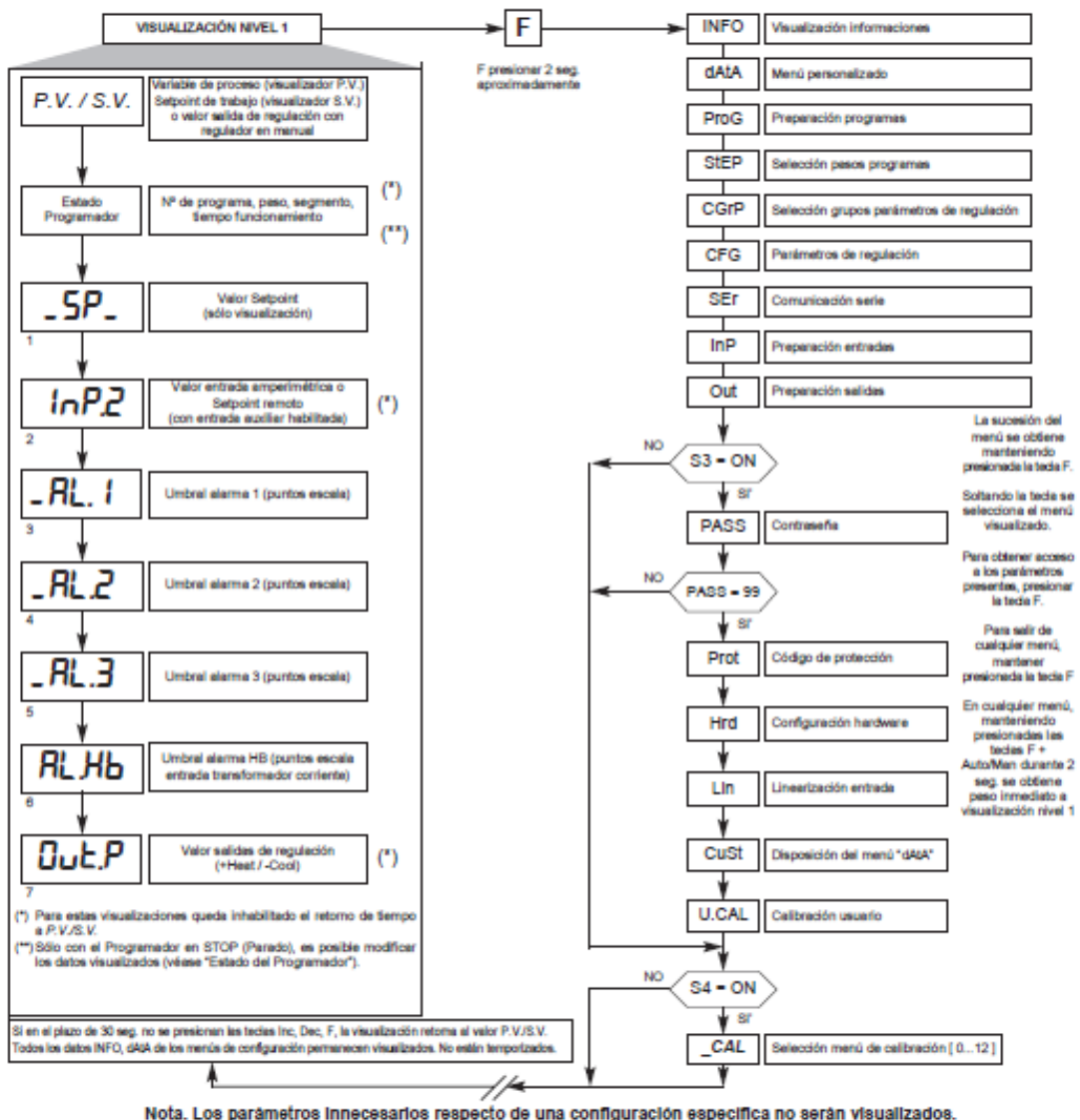
Línea de señal baja

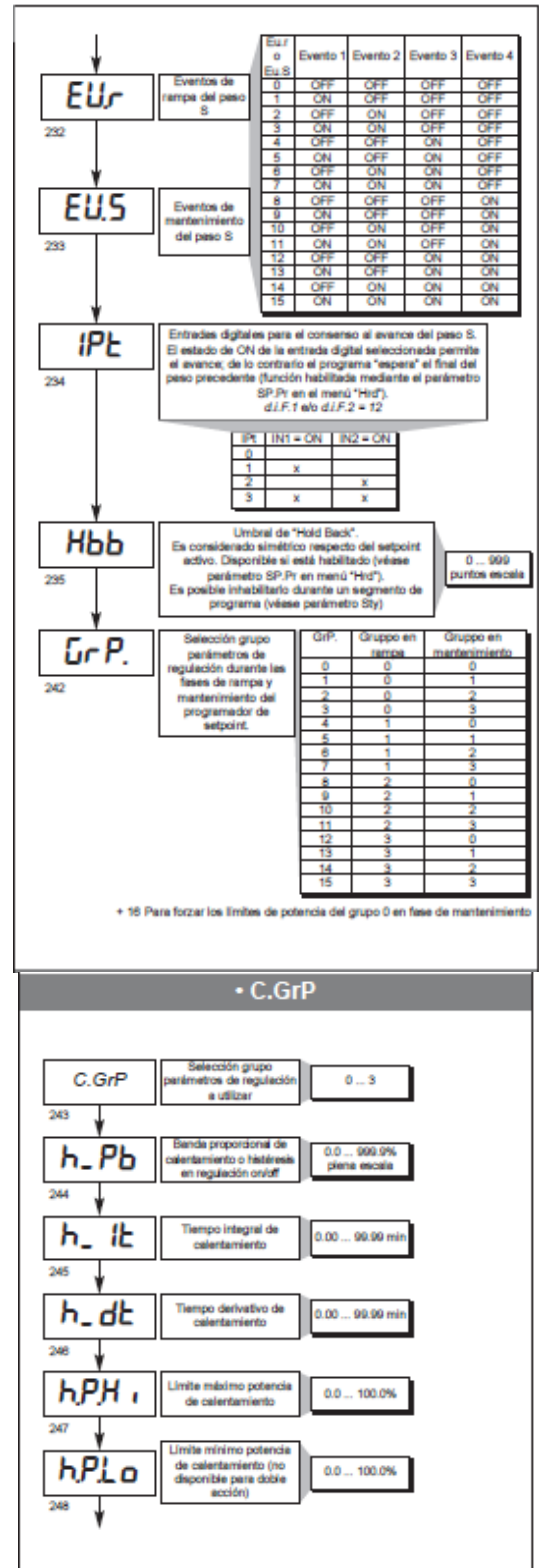
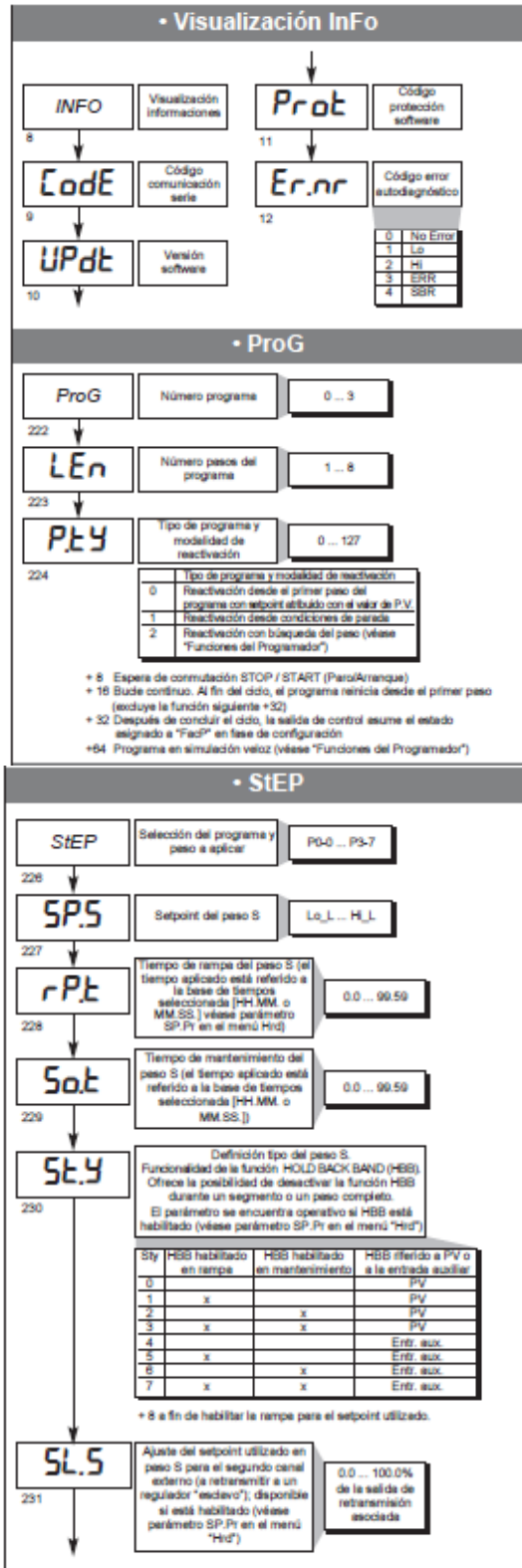
Canal [B] [A]

Estructura del instrumento: identificación tarjetas

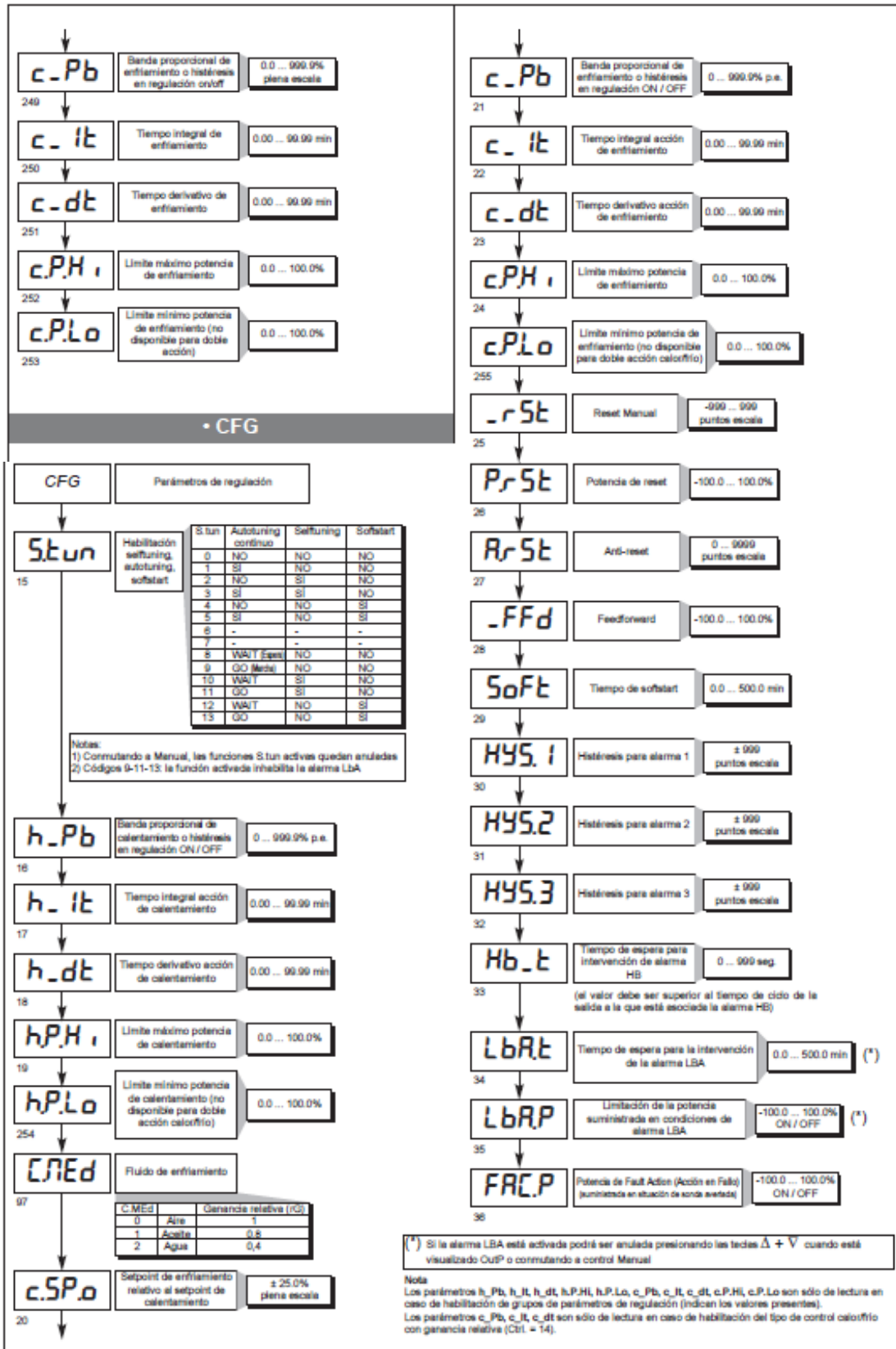


5 • PROGRAMACIÓN Y CONFIGURACIÓN



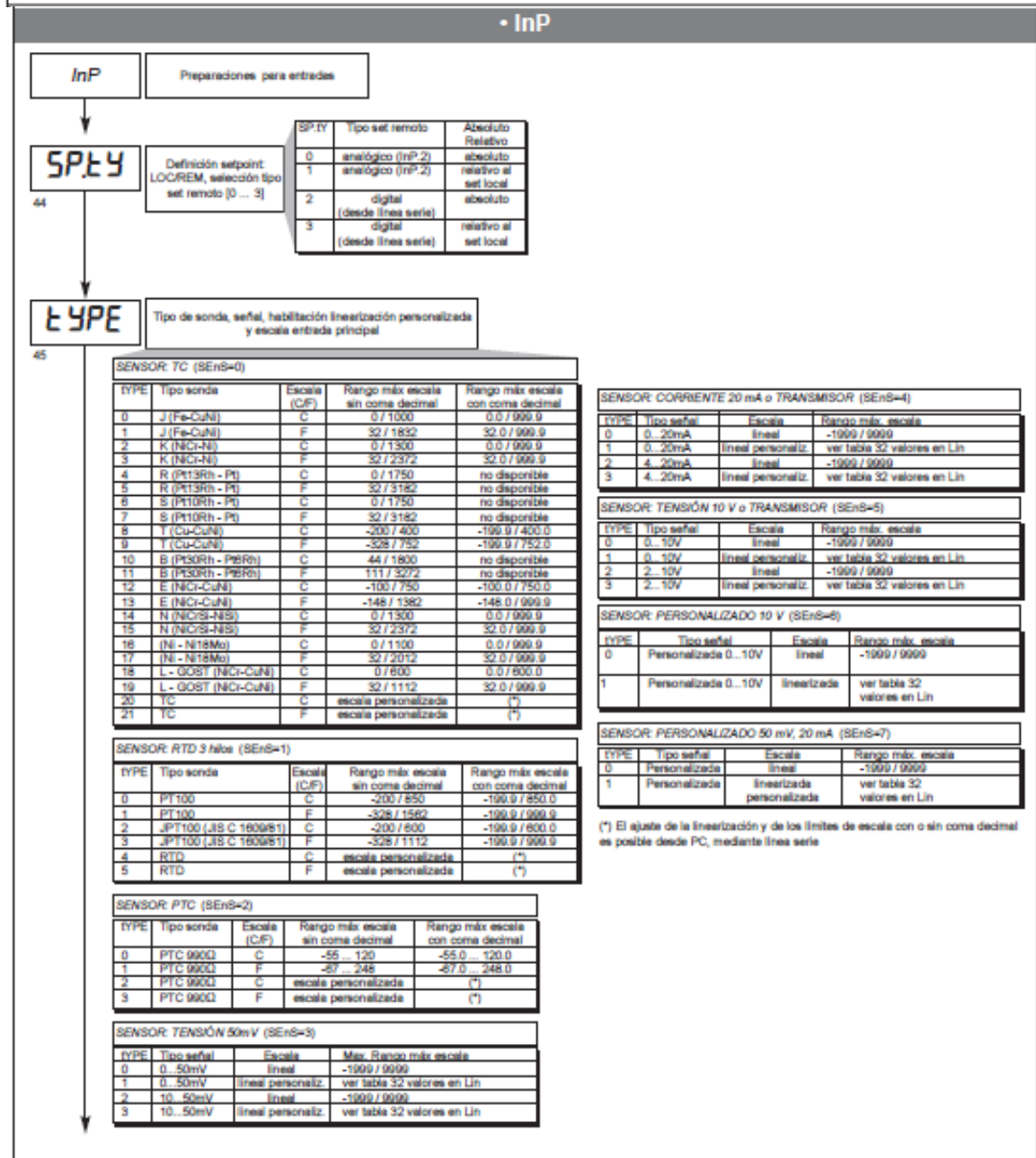
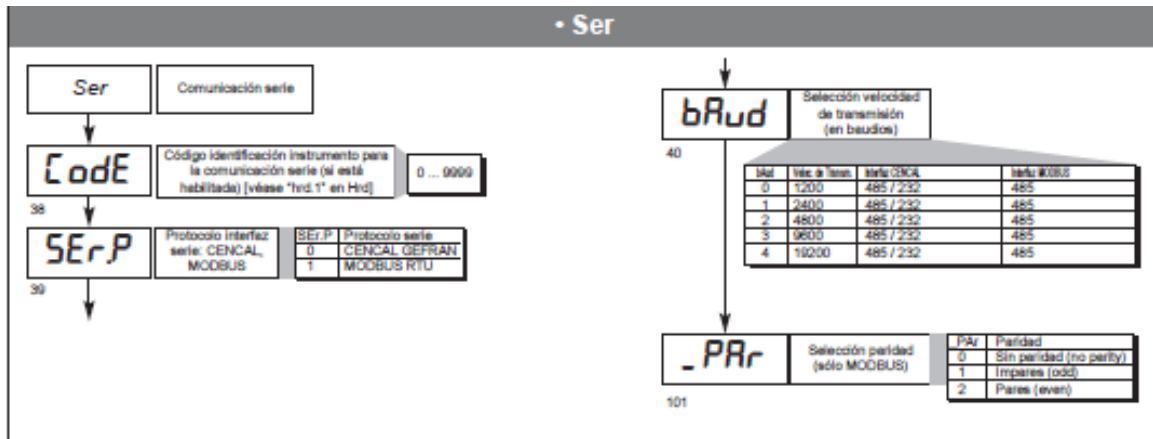


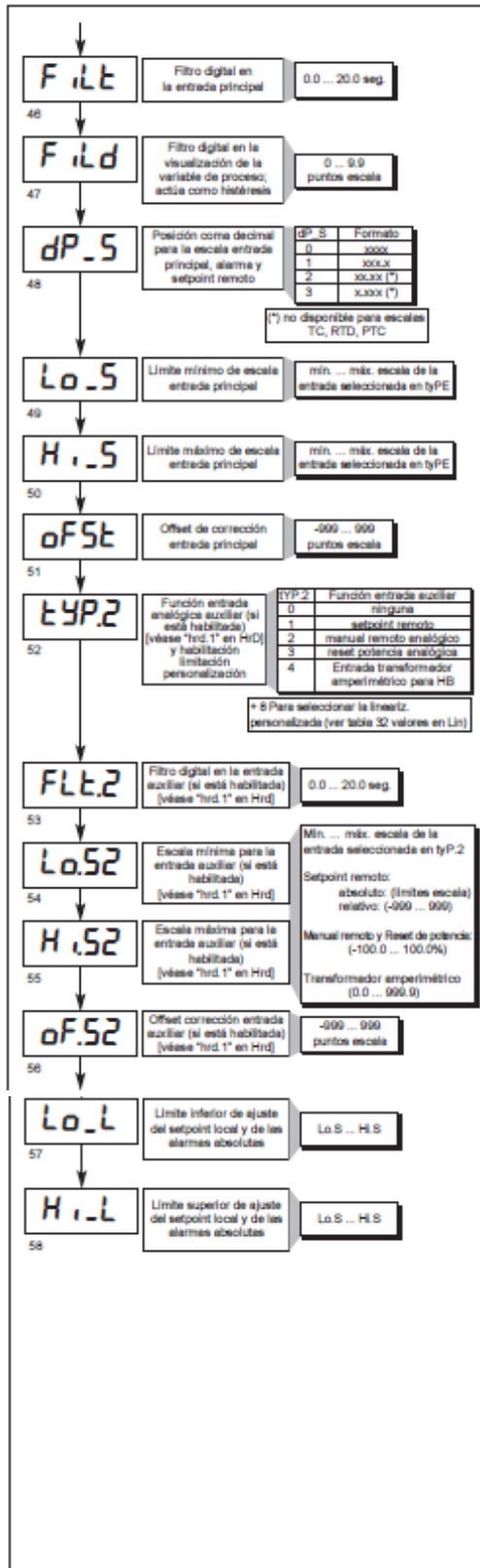
80215E_MHW_800P_0609_ESP



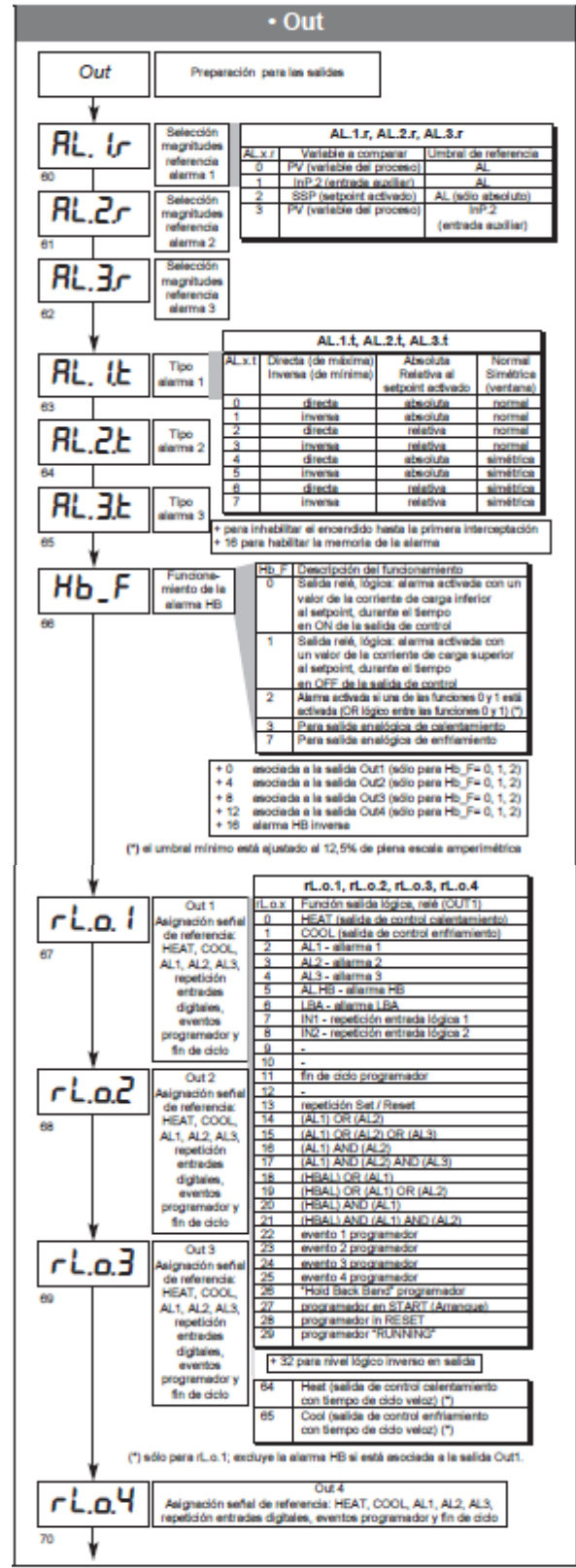
80215E_MHW_800P_0609_ESP

5

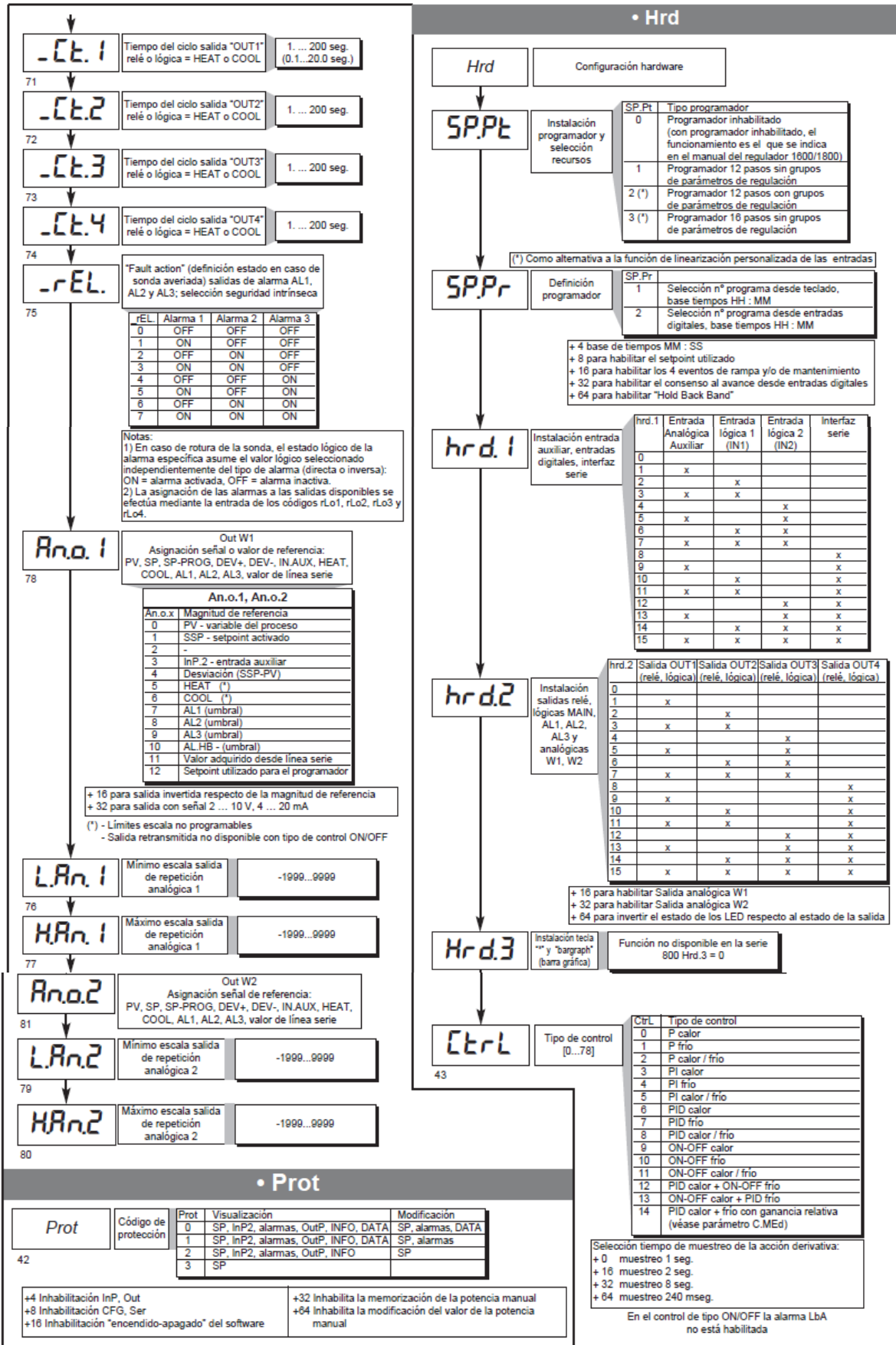


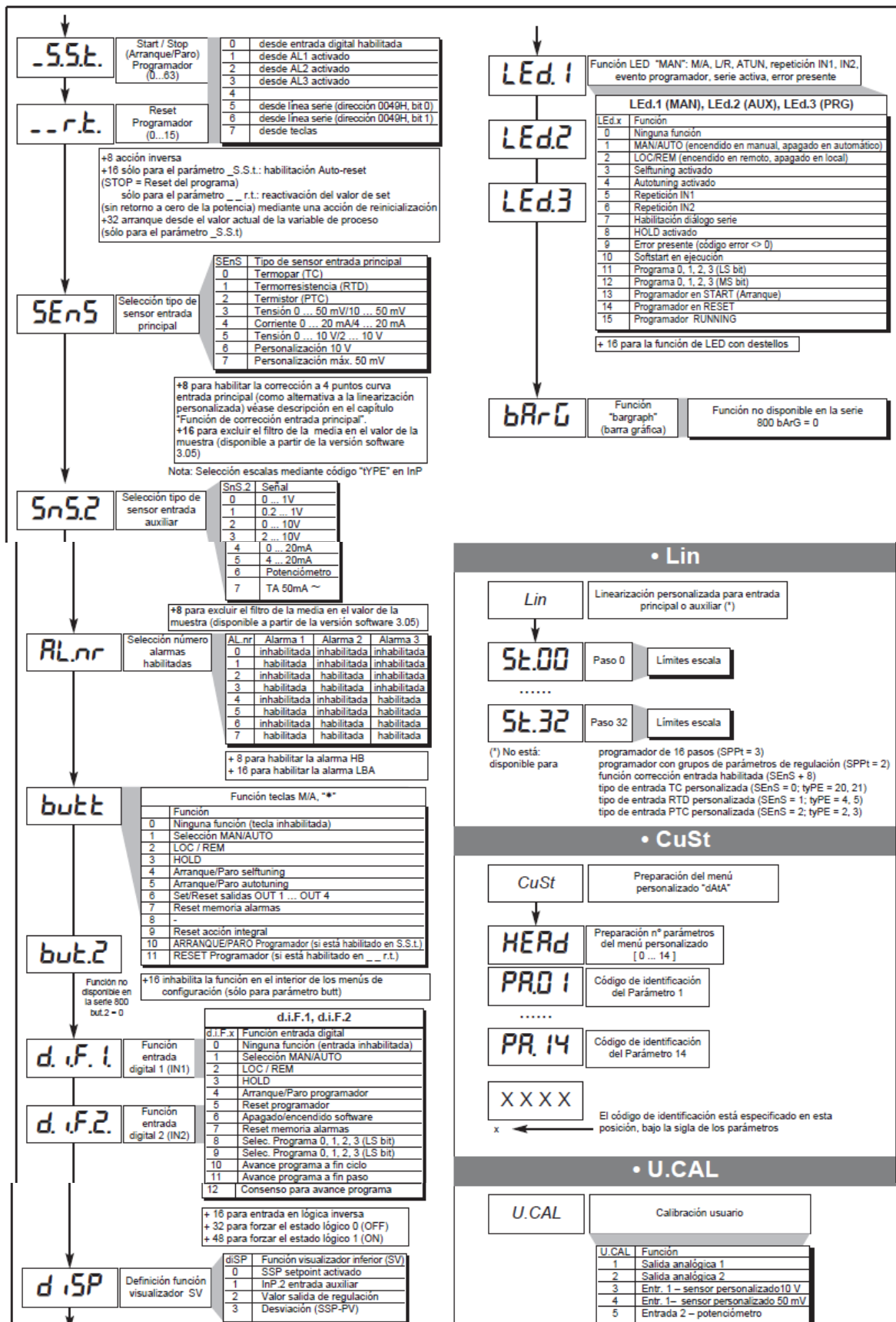


80215E_MHW_800P_0609_ESP



7





80215E_MHW_800P_0609_ESP

9

6 • EL PROGRAMADOR

El instrumento desempeña dos funciones; como regulador y como programador de bucle simple.

La función de programador permite aplicar un programa como a un conjunto de pasos, cada uno de los cuales está formado por dos segmentos:

- √ una rampa
- √ una permanencia.

Cada paso está caracterizado por un conjunto de datos:

- SPs: un valor de setpoint
- rPt: tiempo de rampa entre 0,0 y 99 h 59' (base de tiempos h. m.) o bien 99' 59" (base de tiempos m. s.); programar un tiempo que admita una variación más o menos rápida, en función del valor inicial y del setpoint a alcanzar.
- Sot: tiempo de permanencia entre 0,0 y 99 h 59' (base de tiempos h. m.) o bien 99' 59" (base de tiempos m. s.).
- Hbb: banda de tolerancia simétrica relativa al setpoint y referida a la entrada principal o a la entrada auxiliar.
- Eur: salidas 1 ... 4; código combinación de las salidas (0-15) programables en la fase de rampa.
- EuS: salidas 1 ... 4; código combinación de las salidas (0-15) programables en la fase de mantenimiento.
- iPT: entradas activadas (ON) como consenso para la ejecución.
- SLS: setpoint utilizado para gestionar un regulador "esclavo" con la misma base de tiempos.
- GrP: grupos de parámetros de regulación y límites de potencia (hasta 4) seleccionables al nivel de cada segmento.

Se encuentran disponibles en total 12 (16*) pasos de programa, que pueden constituir un máximo de 4 programas;

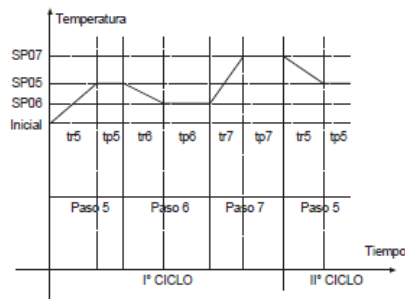
Ejemplos de organización:

2 programas de 8 y 4 pasos; 4 programas de 3 pasos; 2 programas de 6 pasos; etc.

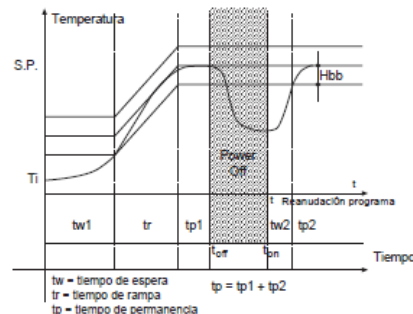
Es importante recordar que el parámetro Sty establece la habilitación de Hbb (en la rampa, en la permanencia o en ambas) y la magnitud de referencia (PV o entrada auxiliar).

(* Como alternativa a la linearización personalizada de las entradas (véase parámetro SP.Pr, menú Hrd)

Ejemplo de PROGRAMA



Ejemplo de FUNCIÓN HBB (banda de mantenimiento)



7 • CARACTERÍSTICAS DEL PROGRAMADOR

- Dispone de un máximo de 12 o 16 (*) pasos, organizables en 4 programas. Un paso de programa comprende la rampa y la permanencia.
- Los tiempos de rampa y de permanencia pueden ser programados con una base seleccionable de 99 horas, 59 min. o bien de 99 min. 59 seg.
- Precisión de la base de tiempos superior a 4 seg. cada 10 horas.
- Selección del programa desde el teclado, entrada digital o línea serie.
- Control del programa desde el teclado, entradas digitales (START/STOP, RESET, fin de programa), desde la línea serie o desde los eventos (AL1, AL2, AL3).
- Modalidad de parada y reactivación del programador: desde entrada digital; desde la tecla "Incrementa" (Arranca), "Decrementa" (PARA) y "M/A" (RESET) en ausencia de otras habilitaciones; desde el estado de las alarmas (ON = ARRANCA); diferentes modalidades de reactivación después de un apagado (power down): desde el setpoint precedente al apagado (power down); desde el valor de la variable del proceso en el momento del encendido; con búsqueda optimizada del setpoint hacia adelante/atrás en el tiempo; con espera del arranque
- En estado de parada es posible modificar: el setpoint existente; el tiempo existente del paso; el n° del programa; el n° del paso; la fase o el segmento (rampa o permanencia)
- Entradas de consenso y salidas de evento asociadas al paso específico. Al inicio de cada paso se analizan las condiciones programadas de entrada. Si estas condiciones se cumplen, la ejecución procede con la actualización de las salidas asociadas y la reactivación de la base de tiempos.
- Señal de fin de programa con o sin forzamiento de las salidas de control.
- Programación de una banda de tolerancia relativa al setpoint, en caso de que la variable supere esta tolerancia, la base de tiempos se detiene (alarma HBB hold back band).
- Setpoint secundario con la misma base de tiempos para gestionar un regulador "esclavo" mediante salida de repetición W1 o W2.
- Modularidad total de las funciones; fácil exclusión de las no deseadas.
- Hasta 4 grupos de parámetros de regulación y límites de potencia seleccionables a nivel de segmento (rampa y/o mantenimiento).(*)

(* Como alternativa a la linearización personalizada de las entradas (véase el parámetro SP.Pr, menú Hrd).

Funcionamiento del Programador

- La modificación del setpoint local, que se verifica durante una fase de parada del programa, provoca la reactivación del paso que se encuentra en fase de ejecución, con conservación del tiempo de rampa predispuesto.
- En caso de apagado y reencendido del instrumento, la ejecución del programa puede continuar o recomenzar desde el primer paso, o buscar el paso con el setpoint más próximo a la variable del proceso (PV) (véase el parámetro Pty en configuración ProG para establecer las condiciones de reactivación).
- La conmutación STOP/START (Paro/Arranque) efectuada al concluirse el programa provoca la reinicialización y reactivación del mismo programa.
- Simulación veloz del programa: un programa seleccionado puede ser controlado fácilmente activándolo en modalidad *simulación veloz*. La habilitación se obtiene operando en el menú ProG en código Pty + 64. El programa opera con tiempos de rampa y permanencia limitados respectivamente a 20 y 10 segundos. Los valores menores que se predispongan, serán respetados.

De esta forma, la duración máxima de un paso es de 30 segundos. Durante el funcionamiento en simulación veloz, la banda de "hold back" (Hbb) queda inhabilitada, mientras que la salida de regulación asume el valor FAC.P.

- Todas las restantes funciones habilitadas, esto es: tipos de reactivación, parada/arranque, reinicialización, manual/automático, fin de ciclo o ciclo continuo, salidas de eventos, consenso desde entradas digitales, setpoint segundo canal, etc. permanecen activadas.
- La función de Auto-reset implica que en fase de STOP (Paro), permanece activado el reset del programador, con la consiguiente adquisición del valor de la variable como setpoint existente y retorno a cero de la base de tiempos.
- Con el regulador en manual o con setpoint remoto absoluto, la base de tiempos del programador permanece detenida.
- Durante el paso desde set remoto a local, el setpoint asume el valor del set remoto del instante de la conmutación
- **Control del programa desde teclado:**

En ausencia de habilitaciones para entradas digitales, alarmas, tecla M/A (butt = 10, 11) el control del programa se efectúa cuando se visualiza el estado del programador mediante las teclas Incrementa, Decrementa y M/A:

Incrementa en STOP (paro) = START (Arranca); Decrementa en START (arranca) = STOP (paro); M/A presionada durante 2 segundos = RESET (manteniendo presionada esta tecla se mantiene la misma situación); Decrementa durante 2 segundos en STOP (paro) = habilitación modificación estado del programador. Mientras no se visualiza el estado del programador, la tecla M/A mantiene la función que ha sido seleccionada con "butt". Incorporación opción para reinicialización del programa con el mantenimiento del setpoint corriente (precedente a la reinicialización), sin puesta en cero de la potencia.

Esta opción se incorpora agregando 16 al valor del parámetro "rt", modalidad de reinicialización programador (menú Hrd).

Modalidad de reinicialización del programador:

El funcionamiento estándar incluye que, con mando activado, el setpoint asuma el valor de la variable del proceso y que la potencia quede forzosamente en valor nulo. Con programación +16 como valor del parámetro "_ _ r.t.", con mando de reset activado se mantienen el setpoint corriente (precedente al reset) y el control de la potencia. Este funcionamiento es válido en caso de reset desde entradas digitales o teclas habilitadas y también en caso de reset sucesivo a un cambio de programa (posible sólo en STOP) o de conmutación STOP/START al concluirse el programa.

Reactivación con búsqueda del paso

En el ejemplo se expone un perfil típico de setpoint realizable con la utilización de un programa de cinco pasos.

Al arranque, si el parámetro Pty = 2 (en ProG), se activa la búsqueda del setpoint con valor igual al de la variable PV.

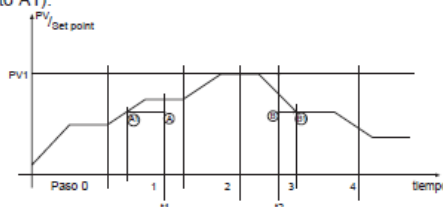
La búsqueda se efectúa desplazando el tiempo aplicado hacia adelante o hacia atrás, saltando fases o pasos.

Si la variable se encuentra en valores inferiores a los requeridos durante una fase de incremento del setpoint (punto A, t1), la reactivación se realiza reduciendo la base de tiempos existente, interceptando el perfil de setpoint (punto A1).

Si la variable se encuentra en valores inferiores a los requeridos durante una fase de reducción del setpoint (punto B, t2), la reactivación se realiza aumentando la base de tiempos establecida, interceptando el perfil del setpoint (punto B1).

Si la interceptación no es posible, como en el caso de la variable del valor PV1, la reactivación del programa se realiza desde el setpoint y del tiempo existente.

En caso de que el control Hbb esté activado, la base de tiempos del programador permanecerá bloqueada mientras que la variable permanecerá fuera de la banda de tolerancia predispuesta, simétrica respecto del valor del setpoint.



8 • ESTADO DEL PROGRAMADOR

EJEMPLO de Visualización del Estado del Programador:

Programa = 2; Paso = 5; Segmento = Mantenimiento; Tiempo transcurrido = 20:42 (MM : SS)



Sólo con el programa en STOP (Paro) es posible modificar el setpoint directamente desde teclado.

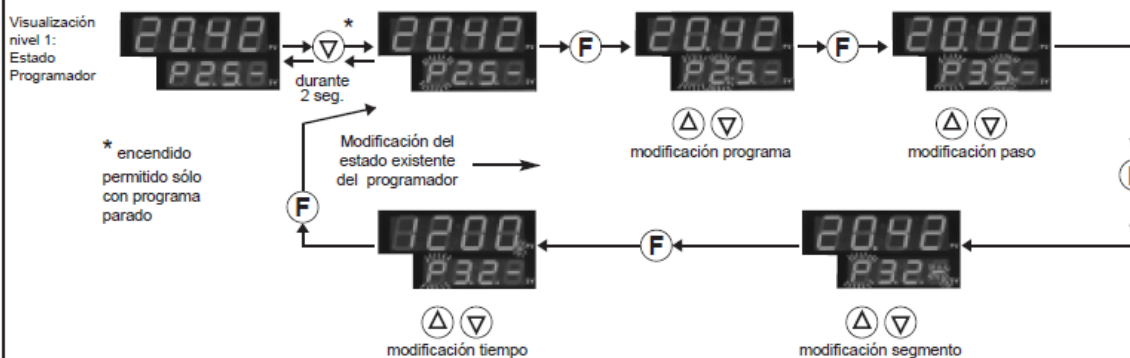
Para modificar el estado del programador: presionando la tecla Decrementa durante 2 segundos, la letra "P" se enciende con intermitencias rápidas. Mediante la tecla "F" se obtiene en secuencia: programa; paso; segmento; tiempo.

La intermitencia de la coma decimal de cada elemento indica la habilitación para la modificación del respectivo valor. Durante esta habilitación, "P" permanece encendida con intermitencias lentas. Mediante las teclas Incrementa y Decrementa se ajustan los valores requeridos. Presionando 2 segundos la tecla Decrementa durante la fase de intermitencias rápidas de "P", o pasando a START (Arranque), se inhabilita la modificación del estado del programador.

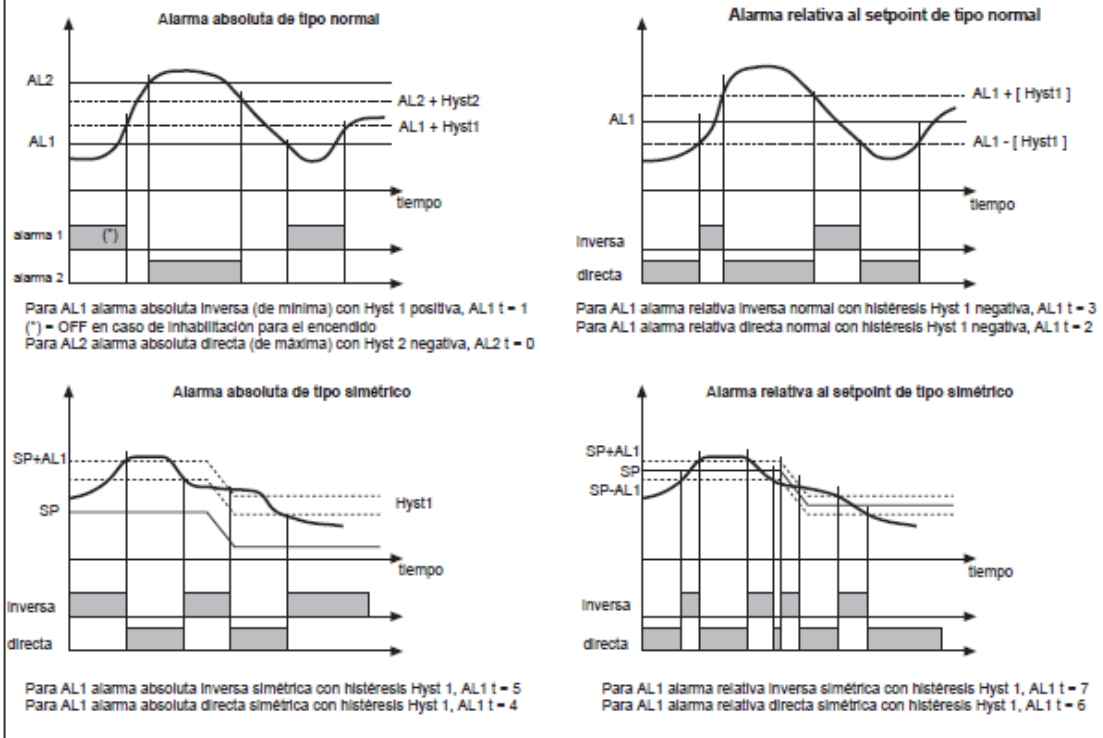
El cambio de programa genera automáticamente un reset.

El estado de reset se obtiene también pasando el paso existente a 0 (cero) y el segmento existente a "Off" (dígito inferior derecho apagado).

Visualización/Modificación del Estado del Programador



9 • ALARMAS



ALARMA HB

Este tipo de alarma está condicionada por el uso de la entrada desde transformador amperimétrico (T.A.). Puede indicar variaciones de absorción en la carga, discriminando el valor de la corriente en entrada amperimétrica en el campo (Lo.S2 ... HI.S2). Es habilitada mediante el código de configuración (Hrd, AL.nr); en este caso, el valor de interceptación de la alarma está expresado en puntos de la escala HB. Mediante el código Hb_F (fase "Out") se selecciona el tipo de funcionamiento y la salida de control asociada. La programación del umbral de alarma es AL.Hb.

La alarma HB directa interviene en caso de que el valor de la entrada amperimétrica descienda por debajo del valor del setpoint entrado para Hb_t segundos del tiempo total en "ON" de la salida seleccionada.

La alarma HB puede activarse sólo con tiempos de ON superiores a 0,4 segundos.

El funcionamiento de la alarma HB incluye el control de la corriente de carga también en el intervalo de OFF del tiempo del ciclo de la salida seleccionada: si durante Hb_t segundos en total del estado en OFF de la salida, la corriente medida supera el 12% de la plena escala amperimétrica, se activa la alarma HB.

El reset de la alarma se efectúa de modo automático al eliminarse la causa de su activación.

Si el setpoint de AL.Hb = 0, inhabilita ambos tipos de alarma HB, con desexcitación del relé asociado.

La indicación de la corriente de carga se visualiza seleccionando la opción InP2 (nivel 1).

NOTA. Los tiempos de ON/OFF se refieren al tiempo de ciclo predispuesto en la salida seleccionada.

La alarma Hb_F = 3 (7), para salida analógica, está activada para un valor de la corriente de carga inferior al setpoint de alarma; ésta queda inhabilitada si el valor de la salida de calentamiento (enfriamiento) es menor del 2%.

ALARMA LBA

Esta alarma indica la interrupción del anillo de regulación debido a un posible cortocircuito en la sonda, sonda invertida o rotura de la carga. Si (AL.nr) está habilitada, determina una alarma en caso de que el valor de la variable no aumente en calentamiento (no se reduzca en enfriamiento) en situación de máxima potencia suministrada durante un tiempo programable (LbA.t).

El valor de la variable queda habilitado sólo fuera de la banda proporcional; para alarma activada, la potencia queda habilitada al valor (LbA.P).

La situación de alarma se reinicializa en caso de aumento de la temperatura en calentamiento (reducción en enfriamiento) o desde teclado, presionando simultáneamente las teclas "V" y "Δ", en visualización nivel 1, opción OutP.

Si el parámetro LbA.t = 0 la función LbA queda inhabilitada.

10 • SOFT-START

Esta función, cuando está habilitada, parcializa porcentualmente la potencia I según el tiempo transcurrido desde que ha sido encendido el instrumento y con referencia al tiempo predispuesto 0,0 ... 500,0 min. (parámetro "SoFt" fase CFG). El softstart es una alternativa al selftuning y se activa después de cada encendido del instrumento. La acción de softstart se reinicializa al pasar a Manual.

11 • ACCIONES DE CONTROL

Acción Proporcional:

acción según la cual la aportación en la salida es proporcional a la desviación en la entrada (la desviación es la diferencia entre variable regulada y valor requerido).

Acción Derivativa:

acción según la cual la aportación en la salida es proporcional a la velocidad de variación de la desviación en la entrada.

Acción Integral:

acción según la cual la aportación en la salida es proporcional a la integral en el tiempo de la desviación de entrada.

Influencia de las acciones Proporcional, Derivativa e Integral en la respuesta del proceso que se está controlando

- * El aumento de la Banda Proporcional reduce las oscilaciones pero aumenta la desviación.
 - * La disminución de la Banda Proporcional reduce la desviación pero provoca oscilaciones de la variable regulada (valores demasiado bajos de la Banda Proporcional confieren inestabilidad al sistema).
 - * El aumento de la Acción Derivativa, correspondiente a un aumento del Tiempo Derivativo, reduce la desviación y evita oscilaciones hasta alcanzarse un valor crítico del Tiempo Derivativo, más allá del cual aumenta la desviación y se verifican oscilaciones prolongadas.
 - * El aumento de la Acción Integral, correspondiente a una reducción del Tiempo Integral, tiende a anular la desviación a régimen entre la variable regulada y el valor requerido (setpoint).
- Si el valor del Tiempo Integral es demasiado largo (Acción Integral débil), es posible que persista la desviación entre la variable regulada y el valor requerido.

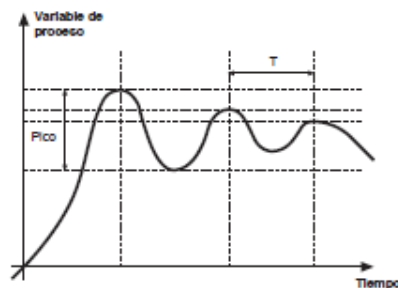
Para mayor información sobre las acciones de control, sírvase contactar con GEFRAN.

12 • TÉCNICA DE SINTONIA MANUAL

A) Ajustar el setpoint a su valor de trabajo.

B) Ajustar la banda proporcional a 0,1% (con regulación de tipo On/Off).

C) Conmutar a Automático y observar la evolución de la variable; se obtendrá un comportamiento similar al de la siguiente figura:



D) Cálculo de los parámetros PID: Valor de la banda proporcional (P.B.).

$$P.B. = \frac{\text{Pico}}{V \text{ máximo} - V \text{ mínimo}} \times 100$$

(V máximo - V mínimo) es el rango de escala.

Valor del tiempo integral $I_t = 1,5 \times T$

Valor del tiempo derivativo $d_t = I_t/4$

E) Conmutar el regulador a Manual, entrar los valores calculados, (rehabilitar la regulación PID ajustando a un tiempo posible del ciclo para salida relé) y volver a conmutar a Automático.

F) De ser posible, para evaluar la optimización de los parámetros, cambiar el valor de setpoint y controlar el comportamiento transitorio; si persiste una oscilación, aumentar el valor de banda proporcional; en cambio, si la respuesta es demasiado lenta, se deberá reducir este valor.

13 • ENCENDIDO/APAGADO DEL SOFTWARE

Cómo apagar: mediante la combinación de teclas "F" e "Incrementa" presionadas conjuntamente durante 5 segundos, es posible desactivar el instrumento, que queda en estado de "OFF", asumiendo un comportamiento similar al del instrumento apagado, sin interrumpir la alimentación de red; mantiene activada la visualización de la variable del proceso, con el visualizador SV apagado.

Todas las salidas (regulación y alarmas) quedan en estado de OFF (nivel lógico 0, relés desexcitados) y todas las funciones del instrumento quedan inhibidas, con excepción de la función de "ENCENDIDO" y el diálogo serie.

Cómo encender: presionando la tecla "F" durante 5 segundos, el instrumento pasa del estado de "OFF" al de "ON". Si durante el estado de "OFF" se interrumpe la alimentación de red, en el siguiente encendido (power-up) el instrumento se predispone en el mismo estado de "OFF"; (el estado de "ON/OFF" está memorizado). La función queda normalmente habilitada; para inhabilitarla se debe entrar el parámetro Prot = Prot + 16. Esta función puede ser asociada a una entrada digital (d.i.F.1 o bien d.i.F.2) y excluye la desactivación desde teclado.

14 • SELF-TUNING

Esta función es válida para sistemas de tipo de acción simple (calor o frío).

La activación del selftuning tiene como objeto el cálculo de los parámetros óptimos de regulación en la fase de inicio del proceso. La variable (por ejemplo, la temperatura) debe ser aquella considerada como a potencia nula (temperatura ambiente).

El regulador suministra el máximo de potencia de salida hasta alcanzarse un valor intermedio entre el valor de inicio y el setpoint, después de lo cual vuelve a cero la potencia. De la evaluación del sobreimpulso y del tiempo necesario para alcanzar el valor de pico se calculan los parámetros PID.

La función completada de este modo se desactiva automáticamente y la regulación continúa aproximándose al setpoint.

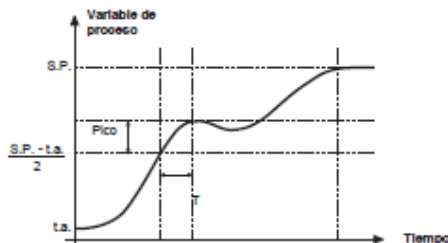
Cómo activar el selftuning:

A. Activación con el encendido

1. Poner el programa en STOP (Paro)
2. Entrar el setpoint al valor requerido
3. Habilitar el selftuning ajustando el parámetro *Stun* al valor 2 (menú CFG)
4. Apagar el instrumento
5. Comprobar que la temperatura sea próxima a la temperatura ambiente
6. Pasar el instrumento a ON (marcha)

B. Activación desde el teclado

1. Comprobar que la tecla M/A esté habilitada para la función START/STOP (Arranque/Paro) del selftuning (código *butt* = 4 menú Hrd)
2. Disponer el programa en STOP (Paro)
3. Ajustar la temperatura a un valor próximo al de la temperatura ambiente
4. Ajustar el setpoint al valor elegido
5. Presionar la tecla M/A para activar el selftuning (Atención: al presionar nuevamente la tecla, el selftuning se interrumpe).



El procedimiento opera de modo automático hasta su finalización. Al final son memorizados los nuevos parámetros PID: banda proporcional, tiempos integral y derivativo calculados para la acción activada (calor y frío). En el caso de doble acción (calor y frío), los parámetros de la acción opuesta son calculados manteniendo la relación inicial entre los respectivos parámetros (ejemplo: $Cpb = Hpb \cdot K$; donde $K = Cpb / Hpb$ en el momento del arranque del selftuning). Al finalizar, el código *Stun* queda anulado automáticamente.

Notas:

- El procedimiento se interrumpe al superarse el setpoint durante su ejecución. En tal caso el código *Stun* no es anulado.
- Se aconseja habilitar uno de los LED configurables para la indicación del estado de selftuning. Predisponiendo en el menú Hrd uno de los parámetros *Led1*, *Led2*, *Led3* = 3 ó 19, el respectivo LED se encenderá con luz fija o intermitente durante la fase de selftuning activado.
- Para el modelo programador, en caso de activación del selftuning a la puesta en marcha del instrumento, el programa queda en STOP (Paro)

15 • AUTO-TUNING

La habilitación de la función autotuning bloquea la entrada manual de los parámetros PID.

Puede ser de dos tipos: permanente y de un sólo impulso.

El primero continúa evaluando las oscilaciones de un sistema buscando lo antes posible los valores de los parámetros PID que reducen la oscilación presente. No interviene si las oscilaciones se reducen a valores inferiores al 1,0 % de la banda proporcional.

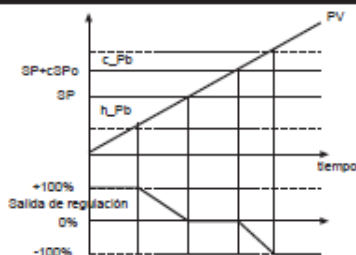
Se interrumpe en el caso de variación del setpoint y se reactiva automáticamente con un setpoint constante. Los parámetros calculados no son memorizados; en caso de apagado del instrumento, el regulador se reactiva con los parámetros que han sido programados antes de habilitar el autotuning.

El autotuning de acción de un solo impulso es útil para efectuar el cálculo en el entorno del setpoint; produce una variación en la salida de control correspondiente al 10% de la potencia de la corriente de regulación y evalúa los efectos del sobreimpulso en función del tiempo.

Estos parámetros son memorizados y reemplazan a los programados anteriormente.

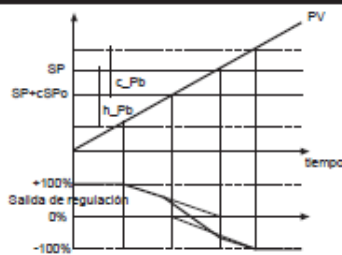
Después de esta perturbación el regulador reanuda el control en el setpoint con los nuevos parámetros. El parámetro activado en CFG es aceptado sólo en el caso de que la potencia de regulación esté comprendida entre 20 y 80%.

16 • REGULACIONES



Salida de regulación sólo con acción proporcional en el caso de banda proporcional de calentamiento separada de la de enfriamiento.

PV = variable del proceso
 SP+cSPo = setpoint de enfriamiento
 c_Pb = banda proporcional de enfriamiento



Salida de regulación sólo con acción proporcional en el caso de banda proporcional de calentamiento superpuesta a la de enfriamiento.

SP = setpoint de calentamiento
 h_Pb = banda proporcional de calentamiento

Regulación Calor/Frío con Ganancia Relativa

En esta modalidad de regulación (habilitada con el parámetro *Ctrl* = 14) es necesario especificar el tipo del enfriamiento.

Los parámetros PID de enfriamiento se calculan a partir de los de calentamiento, de acuerdo a la relación indicada (por ej.: $C.MEd = 1$ (aceite); $H_Pb = 10$; $H_dt = 1$; $H_lt = 4$, implican: $C_Pb = 12,5$; $C_dt = 1$; $C_lt = 4$).

Se aconseja aplicar en la programación de los tiempos de ciclo para las salidas los siguientes valores:

- Aire T Ciclo Enfriamiento = 10 seg.
 Aceite T Ciclo Enfriamiento = 4 seg.
 Agua T Ciclo Enfriamiento = 2 seg.

Nota. En esta modalidad los parámetros de enfriamiento no son modificables.

17 • FUNCIÓN DE CORRECCIÓN ENTRADA PRINCIPAL

Permite efectuar la corrección personalizada de la lectura de la entrada principal mediante la programación de cuatro valores A1, B1, A2, B2. Para habilitar esta función se debe seleccionar el código "Sens" +8 (menú "Hrd").

Ejemplo: Sens = 1 + 8 = 9 para sensor RTD con corrección entrada.

Usando esta función para las escalas lineales (50 mV, 10 V, 20 mA, Pot) es posible invertir la escala.

Los cuatro valores se programan en el menú "Lin" de la siguiente forma: A1 = St00; B1 = St01; A2 = St02; B2 = St03. La programación queda limitada dentro de la escala preestablecida ("LoS" ... "HiS" en el menú "InP").

La función de offset (parámetro "oFt" menú "InP") permanece habilitada.

Limitaciones:

B1 siempre mayor que A1;

B1 - A1 mayor en un 25% de plena escala de la sonda seleccionada.

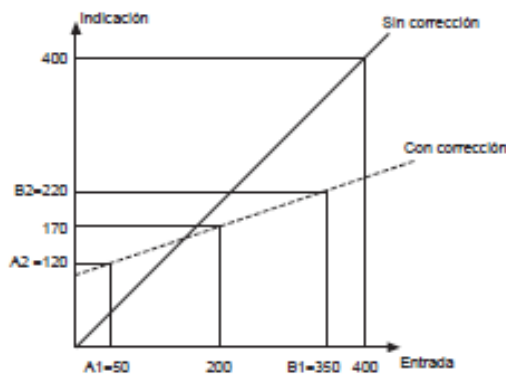
Ejemplo:

Sens = 9; TyPE = 0 (Pt100 escala natural -200 ... +600); dPS = 0

LoS = 0; HiS = 400; oFt = 0

Puntos de referencia en la curva real: A1 = St00 = 50; B1 = St01 = 350 (B1 - A1 = 300 mayor en un 25% de 800)

Puntos correspondientes en la curva corregida: A2 = St02 = 120; B2 = St03 = 220



• 3.6.-DIRECCIONES MODBUS EN EL CONTROLADOR

Addr.	Item	Description	R/W	Min	Max	Dp	Def.	Meas. Unit
0	P.V.	Process variable	R	Lo_S	Hi_S	dP_S	-	S.p.
1	_SP_	Active setpoint	R/W (*1)	Lo_L	Hi_L	dP_S	-	S.p.
2	-	Control output	R/W (*2)	-100.0	100.0	1		%
4	-	Deviation (S.P. - P.V.)	R			dP_S	-	
5	h_Pb	Heating proportional band	R/W (*3)	0.0	999.9	1	10.0	%
6	c_Pb	Cooling proportional band	R/W (*3)	0.0	999.9	1	10.0	%
7	h_It	Heating integral time	R/W (*3)	0.00	99.99	2	4.00	min
8	h_dt	Heating derivative time	R/W (*3)	0.00	99.99	2	1.00	min
9	_Ct.1	Cycle time if Out1 = logic or relay	R/W	0	200	-	10	sec
10	Lo_S	MAIN input scale minimum limit	R/W	-1999	9999	dP_S	0	S.p.
11	Hi_S	MAIN input scale maximum limit	R/W	-1999	9999	dP_S	1000	S.p.
12	-	Alarm point 1 (if relative)	R/W	Lo_L -999	Hi_L 999	dP_S	500	S.p.
13	-	Alarm point 2 (if relative)	R/W	Lo_L -999	Hi_L 999	dP_S	600	S.p.
14	-	Alarm point 3 (if relative)	R/W	Lo_L -999	Hi_L 999	dP_S	700	S.p.
16	-	Local setpoint	R/W	Lo_L	Hi_L	dP_S	400	S.p.
18	SP.rY	Remote setpoint selection	R/W	0	7	-	0	-
20	Lo_L	Lower limit for local setpoint and absolute alarms	R/W	Lo_S	Hi_S	dP_S	0	S.p.
21	Hi_L	Upper limit for local setpoint and absolute alarms	R/W	Lo_S	Hi_S	dP_S	1000	S.p.
22	Gr.SP	Set gradient	R/W	0.0	999.9	1	0.0	digit/min
23	oFSt	MAIN input offset correction	R/W	-999	999	dP_S	0	S.p.
24	FiLt	MAIN input digital filter	R/W	0.0	20.0	1	0.1	sec
25	Lo_L	AL1 low limit	R/W	Lo_S	Hi_S	dP_S	0	S.p.
26	Hi_L	AL1 high limit	R/W	Lo_S	Hi_S	dP_S	1000	S.p.
27	HYS.1	Hysteresis alarm 1	R/W	-999	999	dP_S	-1	S.p.
28	Lo_L	AL2 low limit	R/W	Lo_S	Hi_S	dP_S	0	S.p.
29	Hi_L	AL2 high limit	R/W	Lo_S	Hi_S	dP_S	1000	S.p.
30	HYS.2	Hysteresis alarm 2	R/W	-999	999	dP_S	-1	S.p.
31	S.num	Enabling self-tuning, auto-tuning and soft-start code	R/W	0	13	-	0	
39	c.SP.o	Cooling setpoint	R/W	-25.0	25.0	1	0.0	%
42	h.P.Hi	Heating maximum power limit	R/W (*3)	0.0	100.0	1	100.0	%
43	c.P.Hi	Cooling maximum power limit	R/W (*3)	0.0	100.0	1	100.0	%
44	LbA.t	Waiting time for L.B.A. intervention time	R/W	0.0	500.0	1	30.0	min
45	bAud	Baud rate	R/W	0	4	-	0	-
46	Code	Serial comm. identification code	R/W	0	247	-	1	-

Addr.	Item	Description	R/W	Min	Max	Dp	Def.	Meas. Unit
47	PAr	Parity selection	R/W	0	2	-	0	-
-								
49	Prot	Software protection code	R/W	0	127	-	1	
52	_AL.3	Alarm point 3 (if relative)	R/W	Lo_S -999	Hi_S 999	dP_S	700	S.p.
53	HYS.3	Hysteresis alarm 3	R/W	-999	999	dP_S	-1	S.p.
54	AL.3.t	Alarm 3 type	R/W	0	31	-	0	
55	AL.Hb	HB alarm	R/W	Lo.S2	Hi.S2	1	10.0	S.p.
56	Hb_t	Waiting time for HB alarm intervention	R/W	0	999	-	30	sec
57	Hb_F	HB alarm function	R/W	0	30	-	0	
-								
76	c_It	Cooling integral time	R/W (*3)	0.00	99.99	2	4.00	min
77	c_dt	Cooling derivative time	R/W (*3)	0.00	99.99	2	1.00	min
78	rSt	Manual reset	R/W	-999	999	dP_S	0	S.p.
79	A.rSt	Antireset	R/W	0	9999	dP_S	0	S.p.
80	_FFd	Feedforward	R/W	-100.0	100.0	1	0.0	%
85	Er.nr	P.V. self-diagnostic error code	R	0	4	-		
86	St.00	Custom scale point 0	R/W	Lo_S	Hi_S	dP_S	0	S.p.
87	St.01	Custom scale point 1	R/W	Lo_S	Hi_S	dP_S	31	S.p.
88	St.02	Custom scale point 2	R/W	Lo_S	Hi_S	dP_S	62	S.p.
89	St.03	Custom scale point 3	R/W	Lo_S	Hi_S	dP_S	94	S.p.
90	St.04	Custom scale point 4	R/W	Lo_S	Hi_S	dP_S	125	S.p.
91	St.05	Custom scale point 5	R/W	Lo_S	Hi_S	dP_S	156	S.p.
92	St.06	Custom scale point 6	R/W	Lo_S	Hi_S	dP_S	188	S.p.
93	St.07	Custom scale point 7	R/W	Lo_S	Hi_S	dP_S	219	S.p.
94	St.08	Custom scale point 8	R/W	Lo_S	Hi_S	dP_S	250	S.p.
95	St.09	Custom scale point 9	R/W	Lo_S	Hi_S	dP_S	281	S.p.
96	St.10	Custom scale point 10	R/W	Lo_S	Hi_S	dP_S	313	S.p.
97	St.11	Custom scale point 11	R/W	Lo_S	Hi_S	dP_S	344	S.p.
98	St.12	Custom scale point 12	R/W	Lo_S	Hi_S	dP_S	375	S.p.
99	St.13	Custom scale point 13	R/W	Lo_S	Hi_S	dP_S	406	S.p.
100	St.14	Custom scale point 14	R/W	Lo_S	Hi_S	dP_S	438	S.p.
101	St.15	Custom scale point 15	R/W	Lo_S	Hi_S	dP_S	469	S.p.
102	St.16	Custom scale point 16	R/W	Lo_S	Hi_S	dP_S	500	S.p.
103	St.17	Custom scale point 17	R/W	Lo_S	Hi_S	dP_S	531	S.p.
104	St.18	Custom scale point 18	R/W	Lo_S	Hi_S	dP_S	563	S.p.
105	St.19	Custom scale point 19	R/W	Lo_S	Hi_S	dP_S	594	S.p.
106	St.20	Custom scale point 20	R/W	Lo_S	Hi_S	dP_S	625	S.p.
107	St.21	Custom scale point 21	R/W	Lo_S	Hi_S	dP_S	656	S.p.
108	St.22	Custom scale point 22	R/W	Lo_S	Hi_S	dP_S	688	S.p.
109	St.23	Custom scale point 23	R/W	Lo_S	Hi_S	dP_S	719	S.p.
110	St.24	Custom scale point 24	R/W	Lo_S	Hi_S	dP_S	750	S.p.
111	St.25	Custom scale point 25	R/W	Lo_S	Hi_S	dP_S	781	S.p.
112	St.26	Custom scale point 26	R/W	Lo_S	Hi_S	dP_S	813	S.p.
113	St.27	Custom scale point 27	R/W	Lo_S	Hi_S	dP_S	844	S.p.
114	St.28	Custom scale point 28	R/W	Lo_S	Hi_S	dP_S	875	S.p.
115	St.29	Custom scale point 29	R/W	Lo_S	Hi_S	dP_S	906	S.p.
116	St.30	Custom scale point 30	R/W	Lo_S	Hi_S	dP_S	938	S.p.

Addr.	Item	Description	R/W	Min	Max	Dp	Def.	Meas. Unit
117	St.31	Custom scale point 31	R/W	Lo_S	Hi_S	dP_S	969	S.p.
118	St.32	Custom scale point 32	R/W	Lo_S	Hi_S	dP_S	1000	S.p.
119	LbA.P	Power limit for L.B.A. alarm condition	R/W	-100.0	100.0	1	25.0	%
120		Manufact trade mark (Gefran)	R	-	-	-	5000	-
121		Device ID (800)	R	-	-	-	800	-
122	UPdr	Software Version	R	-	-	-		-
124	SP.Pr	Programmer function	R/W	1	255	-	5	-
125	-At-	Actuator travel time	R/W	0.0	2000	-	60	sec
126	t_Lo	Impulse minimum time/actuator travel time	R/W	0.0	25.0	1	2.0	% At
127	-db-	Valve dead zone	R/W	0.0	25.0	1	0	% FS
-								
132	-	Control output	R/W (*2)	-100.0	100.0	1	-	%
133	butt	M/A key function	R/W	0	27	-	1	-
-								
136	SP.rY	Remote setpoint selection	R/W	0	3	-	0	-
137	-	Active setpoint	R					S.p.
138	-	Local setpoint	R/W	Lo_L	Hi_L	dP_S	400	S.p.
139	-	Auxiliary input (remote setpoint)	R	Lo.S2	Hi.S2	dP_S	-	S.p.
140	d.i.F.1	Digital input 1 (IN1) function	R/W	0	50	-	0	-
141	d.i.F.2	Digital input 2 (IN2) function	R/W	0	50	-	0	-
142	Lo_L	Lower limit for local setpoint and absolute alarms	R/W	Lo_S	Hi_S	dP_S	0	S.p.
143	Hi_L	Upper limit for local setpoint and absolute alarms	R/W	Lo_S	Hi_S	dP_S	1000	S.p.
-								
146	h.P.Hi	Heating maximum power limit	R/W (*3)	0.0	100.0	1	100.0	%
147	SoFt	Soft-Start time	R/W	0.0	500.0	1	0.0	min
148	h_Pb	Heating proportional band	R/W (*3)	0.0	999.9	1	10.0	%
149	h_b	Heating hysteresis (ON/OFF)	R/W (*3)	0.0	999.9	1	10.0	%
150	h_It	Heating integral time	R/W (*3)	0.00	99.99	2	4.00	min
151	h_dt	Heating derivative time	R/W	0.00	99.99	2	1.00	min
152	_Ct.1	Cycle time if Out1 = logic or relay	R/W	0	200	-	10	sec
159	_Ct.2	Cycle time if Out2 = logic or relay	R/W	0	200	-	10	sec
160	rL.o.1	Out1 Allocation of reference signal	R/W	0	65	-	0	-
163	rL.o.2	Out2 Allocation of reference signal	R/W	0	61	-	1	-
166	rL.o.3	Out3 Allocation of reference signal	R/W	0	61	-	2	-
169	_Ct.3	Cycle time if Out3 = logic or relay	R/W	0	200	-	10	sec
170	rL.o.4	Out4 Allocation of reference signal	R/W	0	61	-	3	-
173	_Ct.4	Cycle time if Out4 = logic or relay	R/W	0	200	-	10	sec

Addr.	Item	Description	R/W	Min	Max	Dp	Def.	Meas. Unit
177	-	Alarm point 1 (if relative)	R/W	Lo_L -999	Hi_L 999	dP_S	500	S.p.
178	-	Alarm point 2 (if relative)	R/W	Lo_L -999	Hi_L 999	dP_S	600	S.p.
179	FiLd	Digital filter on display process variable	R/W	0.0	9.9	1	0.5	S.p.
180	Ctrl	Control type	R/W	0	78	-	8	-
-								
187	HYS.1	Hysteresis alarm 1	R/W	-999	999	dP_S	-1	S.p.
188	HYS.2	Hysteresis alarm 2	R/W	-999	999	dP_S	-1	S.p.
189	HYS.3	Hysteresis alarm 3	R/W	-999	999	dP_S	-1	S.p.
-								
191	hrd.1	Hardware configuration 1: Auxiliary inputs Logic inputs Serial interface	R/W	0	95	-		-
192	hrd.2	Hardware configuration 2: Out1,..., Out4 outputs W1, W2 analogue outputs	R/W	0	127	-		-
193	SEnS	Probe type for MAIN input selection	R/W	0	15	-		-
194	SnS.2	Probe type for auxiliary input selection	R/W	0	7	-		-
195	AL.nr	Select number of enabled alarms	R/W	0	31	-		-
196	diSP	Defining SV display function	R/W	0	3	-		-
197	LEd.1	LED 1 function (MAN)	R/W	0	31	-		-
198	LEd.2	LED 2 function (AUX)	R/W	0	31	-		-
199	LEd.3	LED 3 function (REM)	R/W	0	31	-		-
200	HEAd	Set number of parameters in custom menu	R/W	0	14	-	0	-
201	PA.01	Identification code for parameter 1	R/W	0	241	-	15	-
202	PA.02	Identification code for parameter 2	R/W	0	241	-	16	-
203	PA.03	Identification code for parameter 3	R/W	0	241	-	17	-
204	PA.04	Identification code for parameter 4	R/W	0	241	-	18	-
205	PA.05	Identification code for parameter 5	R/W	0	241	-	19	-
206	PA.06	Identification code for parameter 6	R/W	0	241	-	20	-
207	PA.07	Identification code for parameter 7	R/W	0	241	-	21	-
208	PA.08	Identification code for parameter 8	R/W	0	241	-	22	-
209	PA.09	Identification code for parameter 9	R/W	0	241	-	23	-
210	PA.10	Identification code for parameter 10	R/W	0	241	-	24	-
211	PA.11	Identification code for parameter 11	R/W	0	241	-	25	-
212	PA.12	Identification code for parameter 12	R/W	0	241	-	26	-
213	PA.13	Identification code for parameter 13	R/W	0	241	-	27	-
214	PA.14	Identification code for parameter 14	R/W	0	241	-	28	-
215	AL.1.r	Reference signal for alarm 1	R/W	0	5	-	0	-
216	AL.2.r	Reference signal for alarm 2	R/W	0	5	-	0	-
217	AL.3.r	Reference signal for alarm 3	R/W	0	5	-	0	-
218	rYP.2	Auxiliary analogue input function	R/W	0	13	-		-
219	FLt.2	Digital filter on auxiliary input	R/W	0.0	20.0	1	0.1	sec
220	oF.S2	Auxiliary input offset correction	R/W	-999	999	dP_S		S.p.
221	L.An.1	W1 analogue signal output 1 minimum limit	R/W	-1999	9999	dP_S		S.p.

Addr.	Item	Description	R/W	Min	Max	Dp	Def.	Meas. Unit
222	H.An.1	W1 analogue signal output 1 maximum limit	R/W	-1999	9999	dP_S		S.p.
223	An.o.1	W1 assignment of signal or reference value	R/W	0	63	-		-
224	L.An.2	W2 analogue signal output 2 minimum limit	R/W	-1999	9999	dP_S		S.p.
225	H.An.2	W2 analogue signal output 2 maximum limit	R/W	-1999	9999	dP_S		S.p.
226	An.o.2	W2 assignment of signal or reference value	R/W	0	63	-		-
227	InP.2	Amperometric input value or remote setpoint	R	Lo.S2	Hi.S2	dP_S		S.p.
228	FAC.P	Power in fault action	R/W	-100.0	100.0	1	0.0	%
229	_rEL.	Fault action (sets state in case of probe fault)	R/W	0	7	-	0	-
230	_SP.1	Setpoint 1	R/W	Lo_L	Hi_L	dP_S	100	S.p.
231	_SP.2	Setpoint 2	R/W	Lo_L	Hi_L	dP_S	200	S.p.
232	_SP.3	Setpoint 3	R/W	Lo_L	Hi_L	dP_S	300	S.p.
233	_SP.4	Setpoint 4	R/W	Lo_L	Hi_L	dP_S	400	S.p.
234	Gr.SP	Set gradient	R/W	0.0	999.9	1	0.0	digit/min
235	_tS	Limit value of timer	R/W	0	9999	0	10	sec
236	_S.S.t	Start/Stop timer	R/W	0	63	-		-
237	_r.t	Reset timer	R/W	0	31	-		-
238	-At-	Actuator travel time	R/W	0.0	2000	-	60	sec
239	t_Lo	Impulse minimum time/actuator travel time	R/W	0.0	25.0	1	2.0	%At
240	t_Hi	Pulse alarm point	R/W	0.0	100.0	1	0.0	%FS
241	-db-	Valve dead zone	R/W	0.0	25.0	1	0.0	%
242	At.rY	Valves control type	R/W	0	11	-		-
243	SP.Pr	Programmer function	R/W	1	127	-	5	-
244	-	Active program number	R					-
245	-	Active step number	R					-
246	-	Active segment	R					-
247	-	Actual time program	R					-
248	-	Program actual setpoint (stop)	R					S.p.
249	SP.rY	Remote setpoint selection	R/W	0	7	-	0	-
250	-	Remote setpoint from serial	R/W	Lo_L	Hi_L			S.p.
251	-	Out W from serial (*4)	R/W	0	65535			
252	-	Manual control output	R/W	-100.0	100.0	1		%
253	SP.Pt	Programmer installation and resource selection	R/W	0	3	-	0	-
254	h.P.Lo	Heating minimum power limit	R/W	0.0	100.0	1	0.0	%
255	c.P.Lo	Cooling minimum power limit	R/W	0.0	100.0	1	0.0	%
256	hrd.3	** * ** key and bargraph installation	R/W	0	3	-	0	-
257	but.2	** * ** key function	R/W	0	11	-	0	-
258	BarG	Bargraph function	R/W	0	8	-	0	-
-								
400	fYPE	Probe type, signal, enable custom linearization, main input scale	R/W	0	21	-	0	
401	Lo_S	MAIN input scale minimum limit	R/W	-1999	9999	dP_S	0	S.p.

Addr.	Item	Description	R/W	Min	Max	Dp	Def.	Meas. Unit
402	Hi_S	MAIN input scale maximum limit	R/W	-1999	9999	dP_S	1000	S.p.
403	dP_S	Decimal point position	R/W	0	3	-	0	
404	Lo_S2	Auxiliary input minimum range	R/W	-1999	9999	dP_S	0	S.p.
405	Hi_S2	Auxiliary input maximum range	R/W	-1999	9999	dP_S	1000	S.p.
406	AL_1.t	Alarm 1 type	R/W	0	31	-	0	
407	AL_2.t	Alarm 2 type	R/W	0	31	-	0	
408	AL_3.t	Alarm 3 type	R/W	0	31	-	0	
513	C.MEd	Cooling medium	R/W	0	2	-	0	-
-								
516	P.rSt	Reset power	R/W	-100.0	100.0	1	0.0	%
519	oFSt	MAIN input offset correction	R/W	-999	999	dP_S	0	S.p.

Remarks:

- (*1) They can be written in local (controller) and STOP (programmer) mode only
- (*2) They can be written in manual mode only
- (*3) Read only for programmer with control parameter (SP.Pt = 2)
- (*4) 32.768 (8000 H) value is not allowed

Addr.	Item	Description	R/W	Min	Max	Dp	Def.	Meas. Unit
Program Selection								
2731 + 4*N 0 ≤ N ≤ 3	LEn	Number of program steps	R/W	1	8	-		
2732 + 4*N	P.tY	Program type and restart mode	R/W	0	127	-		
Steps Selection								
1530 + 12*m 0 ≤ m ≤ 11 (§)	SP.S m	" m " step setpoint	R/W	Lo_L	Hi_L	dP_S		S.p.
1532 + 12*m	rP.t m	" m " step ramp time	R/W	00.00	99.59	2	0	hh.mm o mm.sec
1534 + 12*m	So.t m	" m " step hold time	R/W	00.00	99.59	2	0	hh.mm o mm.sec
1535 + 12*m	Hbb m	" m " hold back band value	R/W	0	999	dP_S	0	S.p.
1536 + 12*m	St.Y m	Defines " m " step type	R/W	0	15	-		
1537 + 12*m	SL.S m	Setpoint slaved for second external channel	R/W	0.0	100.0	-	0	%
1538 + 12*m	EU.r m	" m " step 4 ramp events	R/W	0	15	-	0	
1539 + 12*m	EU.S m	" m " step 4 hold events	R/W	0	15	-	0	
1540 + 12*m	IPt m	2 enabling inputs	R/W	0	3	-	0	
1541 + 12*m	GrP. m	PID control parameter group selection	R/W	0	31	-	0	-

Remarks:

- (§) m step number: 0 ≤ m ≤ 15 setting SP.Pt = 3

800/1600/1800 MODBUS BIT

Software Version V3.20

Addr.	Description	R/W
0	Active self-tuning	R
1	Auto/Man	R/W
-		
4	AL1 status	R
5	AL2 status	R
8	AL_LBA status	R
9	Sbr faulty sensor	R
10	Local/Remote SP	R/W
11	ON (=0)/OFF (=1) software	R/W
12	Out1 status	R
13	Out2 status	R
14	Out3 status	R
15	Out4 status	R
16	(Run) active programmer	R
17	Running programmer	R
18	Start (=1)/Stop (=0) programmer or timer	R
19	Reset (=0)/Go (=1) programmer or timer	R
20	Program end (=1)	R
21	Program in ramp (=1)	R
22	Program in hold (=1)	R
23	Waiting enabling	R
24	Start/Stop program for serial line	R/W
25	Reset program from serial line	R/W
26	AL HB alarm status	R
27	HBB active	R
28	Auto-Tuning active	R
-		
46	AL1 direct/inverse	R/W
47	AL1 absolute/relativ	R/W
48	AL1 normal/symmetric	R/W
49	AL1 disabled when switching ON	R/W
50	AL1 with memory	R/W
-		
54	AL2 direct/inverse	R/W
55	AL2 absolute/relativ	R/W
56	AL2 normal/symmetric	R/W
57	AL2 disabled when switching ON	R/W
58	AL2 with memory	R/W
-		
62	AL3 status	R
63	Running soft-start	R
64	Hold input active	R
65	Timer at zero	R
66	Auto (=0)/Man (=1) programmer	R
67	Ramp type: up (=0)/down (=1)	R
68	Digital input 1 status	R
69	Digital input 2 status	R
-		

4.-BIBLIOGRAFIA:

[1] Instalaciones de telecomunicaciones para edificios (2013)

Carlos Fernández Valdivieso – Ignacio Raúl Matías Maestro – Ander Gabilondo Areta - Carlos Ruiz Zamarreño – Francisco Javier Falcone Lanas – Idoya Castells Portal – Ignacio Del Villar Fernández – Javier Militino Anchorena – Juan Antonio Nazabal Urriza

[2] Analog Electronics with Labview / Keneth L. Ashley (2003)

[3] Labview 2009 student edition / Robert H. Bishop (2010)

[4] www.ni.com

[5] www.gefran.com