PROYECTO REEMPLAZO V7S-00-ÍNDICE

R2021-09-27. (2 pág.)

Borrador comentarios preliminares para analizar en fábrica sobre voltímetro control.

01- CONTROL. (2 pág.)

Borrador idea genérica de análisis en fábrica de anexar posibles funciones de control.

P0794A\_HD.pdf (1 pág.)

Borrador idea hardware mediciones y control.

02- GENERALIDADES. (2 pág.)

Funciones como voltímetro. Cantidad canales y multiplexación. Promediación. Resolución mediciones. Resolución display. Loop mediciones. Escalas en Eeprom.

03- PROMEDIACIÓN. (1 pág.)

Algoritmos para promediar cada canal.

04- ESCALAS. (1 pág.)

Conversión de salidas de los adc a valores a mostrar en display. Datos no volátiles (Eeprom).

05- FORMATOS PRESENTACIÓN DISPLAY. (2 pág.)

Presentación para mediciones lineales, cuadráticas y cuadráticas con máxima resolución.

06- EEPROM ESCALAS. (2 pág.)

Parámetros guardados en eeprom para calibración entradas analógicas. Display en modo vúmetro.

07- EEPROM REFERENCIA. (3 pág.)

Muestra organización eeprom equipos anteriores para tener como referncia.

08- PROTOCOLO COMUNICACIÓN. (6 pág.)

Muestra protocolo comunicación con PC en otros equipos. Tenerlo presente para tratar de hacer similar o compatible.

09- DIAGRAMA CONTROL GENERAL. (1 pág.)

Descripción resumida de posibles funciones de control a desarrollar.

P0794-A\_DF-VOLT Y CTRL CON MICRO Rev1.pdf (1 pág.)

Diagrama funcionamiento Control General.

10- DETALLES CONTROL GENERAL. (6 pág)

Explicación Diagrama Control General.

continúa.....

Salida dac Ajuste de Potencia.

Salida pwm ventiladores.

Funciones Teclado y Leds.

Logbook eventos.

.......

La placa de mediciones digitales deberá encargarse también del Control General del equipo. Esto es fijar modo de encendido, realizar las distintas protecciones, ajustar la potencia del equipo, etc.

Para el diseño funcional de los bloques de Control General (del Tx), se tomará referencia el funcionamiento de las placas P0167, P0490 y P0534, y para implementarlo con el (mismo) microcontrolador, este recibirá las siguientes informaciones, y generará las salidas que correspondan:

Lecturas entradas Lógicas

-Ingreso señal eléctrica LÓGICA ENC AUT en entrada LÓGICA.

-Ingreso señal eléctrica LÓGICA falla >>ROE en entrada LÓGICA.

-Ingreso señal eléctrica LÓGICA falla >>TEMP en entrada LÓGICA.

-Ingreso señal eléctrica LÓGICA P50 EXT en entrada LÓGICA (equipos FM1K / 2K).

-Ingreso señal eléctrica LÓGICA falla -VGG en entrada LÓGICA (P0167-B en TRM2640).

-Ingreso señal eléctrica LÓGICA apagado externo (posible uso futuro).

Lecturas entradas Analógicas

-Ingreso señal eléctrica ANALÓGICA NTC1 en entrada ADC.

-Ingreso señal eléctrica ANALÓGICA NTC2 en entrada ADC (estas entradas ADC no irían multiplexadas con las ADC de mediciones a mostrar en el display porque tienen interfaz entrada distinta).

-(analizar posible entrada para un tercer NTC3 en entrada ADC).

-Ingreso señal eléctrica ANALÓGICA Ref Temp NTC (definir si LM35 local o Med Temp Aire).

-Compartir ingreso señales eléctricas ANALÓGICAS de PD, PR y PEXC para el AGC y PMOD y VFTE para MUTE del AGC.

Lectura configuraciones en Eeprom seteadas desde PC por Lie

-Lectura de Eeprom VALOR (seteado desde PC) tipo de P0167/490: si <<VENT apaga FTE (P0490-A), ó saca Low en HVGG (pin 9 P0490-C), ó saca High en <<VENT (pin 9 P0490-D), ó <<VENT2 saca High en <<VSUM (pin 9 P0490-E); ó falla -VGG apaga (pin 1 P0167-B para TRM2640). Cada una de estas opciones además afecta la lógica si <<VENT apaga o no Tx.

-Lectura de Eeporm VALOR número ADC para referencia MED TEMP AIRE para <<VENT.

-Lectura de Eeprom VALORES ajuste comparadores VNTC para T\_AIRE=0, para T\_AIRE=40 y valor HISTÉRESIS (para simpleza se considerará como si la VNTC fuese lineal con la temperatura).

-Lectura de Eeprom si la actuación de >>T debe enclavar o no.

-Lectura de Eeprom qué entradas analógicas se usaría como med. de Temp. en caso de amplificadores sin termostáto (con P0673) para usar como >>TEMP e histéresis.

-Lectura de Eeprom VALOR PNOMinal del Tx.

-Lectura de Eeprom ubicaciones (entradas analógicas) mediciones PD, PR, PEXC y PMOD, y valores comparadores para AGC (PD\_NOM, PE\_NOM, PR\_Foldback) y umbral PMOD, y rango (VMÍN, VMÁX), velocidad actuación AGC (pendiente V PIN respecto diferencia PD-PSET; también si deberán o no tener efecto c/u (PD, PR, PEXC y PMOD).

-Lectura de Eeprom VALORES Temp1 y Temp2 entre los cuales el PWM para turbinas (en caso de contemplarse en este diseño) dará un DC del 0% y del 100% respectivamente, y cuál entrada ADC se tomará como referencia (la del Aire, la del AMPL, etc).

-Lectura de Eeprom ubicaciones medición fuentes y valores de V a comparar para indicar Falla de Fuente (es una falla que no la venimos acusando pero suele ser bastante común en los equipos).

-Lectura de Eeprom retardo encendido automático al recibir señal del modulador, y retardo apagado al cesar.

Lecturas configuraciones en eeprom seteadas desde teclado por usuario Control en Tx

-Lectura de Eeprom estado LÓGICO (seteado desde teclado) Encendido Manual / Automático.

-Lectura de Eeprom estado LÓGICO (seteado desde teclado) Manual SI / NO.

-Lectura de Eeprom estados LÓGICOS (seteado desde teclado) cómo arranca al conectar 220 si se encontraba en MAN/SI o en AUT y con entrada de ENC AUT (APAG, ENC, ó como venía estando al cortarse 220).

-Lectura de Eeprom VALOR (seteado desde teclado) potencia deseada PSET (para AGC mediante microcontrolador).

Señales de Salida del uC para Control General del Tx

-Salida LÓGICA <<VENT/<<VENT2/LowHVGG/....Como ya se mencionó el pin 9 funciona de distintas maneras, y se indicó su lógica de acuerdo a la versión de la P0490 que corresponda (o la función homóloga en el pin 1 de la P0167). Quedará por definir si convendrá en el PCB hacer que sean pines distintos (y así podría evitarse ese seteo en la eeprom) pero consumiríamos más pines del microcontrolador.

-Salida LÓGICA Led Verde. Para indicar funcionamiento normal del Tx.

-Salida LÓGICA Led Rojo de APAG. Para indicar equipo apagado (por usuario).

-Salida LÓGICA Led Rojo. Para indicar fallas (ROE, TEMP, VENT, Fuente?, ....

-Salida LÓGICA Led Amarillo. Para indicar FoldBack ROE, o AGC controlado por EXC, o AGC al máx, ...(este led podría evitarse haciendo actuar simultaneamente el Verde y el Rojo de fallas ahorrando un pin del micro pero usando un led especial RO+VE).

-Salida LÓGICA energización relé HAB FTE.

-Salida LÓGICA HAB 12V.

-Salida LÓGICA HVGG.

-Salida LÓGICA energización NTC (por verificar si puede ser siempre la misma que habilita al relé HAB FTE).

-Salidas DAC para generar VAGC (hay uC con DACs de 5 bits; se utilizarían 2 DAC para 10 bits).

-Salida PWM VREF POT (opción en caso de no implementarse los DAC y AGC controlado).

-Salida PWM control ventiladores (opción a analizarse su implementación).

Otras conexiones del uC

-Entradas analógicas (cantidad dependerá de hardware de multiplexación).

-Display LCD (un LCD de 2 x 16 C utiliza unos 11 pines).

-Teclado (a definir, unos 5 pines).

-Comunicación con PC: dos pines Tx y Rx.

-Firmware: 3 pines.

GENERALIDADES

Sobre cada tema que se vaya presentando, habrá facilidad de realizar alternativas propuestas por los programadores, tanto en hardware como en software, y acordadas luego con la fábrica.

Consecuentemente a medida que se van generando, podrían tener sus variantes con el avance del proyecto, al tener presente que habrá una realimentación con los programadores, mientras se vayan evaluando el tipo de componentes y la complejidad del proyecto. En particular la decisión de incluir o no, o en qué medida, funciones de control (es decir aquellas que interactúan con el equipo donde vaya esta placa, aparte de las funciones de monitoreo que realiza como voltímetro). Un caso en particular afectado por lo mencionado, sería qué tamaño y cómo organizar la eeprom asociada (ya sea interna o externa al micro).

FUNCIONES COMO VOLTÍMETRO

-Adquisición (hasta) 16 señales analógicas, previo paso por circuitos atenuador y multiplexor (para 2 a 4 entradas ADC del micro).

-Conectado a teclado de unas pocas teclas y display preliminar posiblemente LCD 2 x 16.

-Acceso comunicación con PC para setear parámetros de cada medición (que se guardarán en eeprom), y acceder lectura de todas las mediciones.

-La eeprom será accedida para poder ser modificada, tanto por la PC (acceso total) como por el teclado (acceso limitado a algunos parámetros, como ejm. ajuste de potencia del equipo como función de CONTROL), y los datos allí guardados se utilizarán por ejemplo para indicar escala para cada medición, unidad a mostrar, tipo de promediación, etc.

-El teclado minimamente permitirá acceder a cada medición mostrándolo en el display LCD.

-El display mostrará posiblemente a lo sumo una medición por pantalla, pero con la PC se debiera ver actualizadas todas (las 16?) en su monitor, entonces una rutina principal será que el voltímetro esté barriendo continuamente todas las mediciones, guardandolas en RAM para el acceso de la PC, mientras una se estará mostrando en el display LCD.

-En la RAM se guardarán y actualizarán continuamente las mediciones como salen de los adc (unos 2 bytes por cada parámetro a medir considerando adc de 10 bits), y además otro conjunto (otros 2 bytes por parámetro) con valores promediados, según fórmulas a indicarse más adelante.

-La tarea de promediación de cada parámetro se realizará ni bien se adquiere la medición de ese canal, así mientras tanto se da tiempo para que la entrada del adc se estabilice con el próximo parámetro a medir. De tal manera que cumpla un loop de adquisición y promediación parámetro o canal MED(i), idem MED(i+1), ..., MED(16), MED(1), ...

-Para minimizar los retardos que demandan el settling time del los adc, es que si como ejm. se utilizan dos multipl. x8, que igresan a dos entradas adc, la MED(1) estará en el 1er multipl. y llegará a la adc1 del micro, y la MED(2) estará en el 2do multipl. y llegará al adc2 del micro. Y así alternativamente. Un multiplex. para las impares y el otro para las pares (suponiendo usar dos entradas adc del micro).

-La medición a mostrar en el display se tomará de la RAM promediada, y se convertirá a la escala que corresponda, según parámetros fijados en la eeprom.

-Las mediciones promediadas ubicadas en su RAM tendrán mayor cantidad de bits que las no promediadas; si estas tienen 10 bits útiles (según adc del micro), las promediadas podrán ocupar los 16 bits disponibles (el cálculo de promediación genera la mayor resolución).

-Pero desde la PC se deberá poder acceder a ambas RAM, tanto las sin promediar como las promediadas. En cada caso desde la PC se inviará un código distinto si se requiere el conjunto de las promediadas, o las sin promediar. Sin previa conversión a escalas.

-Más adelante se definirá si aparte se utilizarán otros códigos para requerir las mediciones mediante la PC, pero ya convertidas al valor a mostrar. En estos casos habrá que convenir el formato de datos (probablemente compatible con algún protocolo general, como el SNMP; en tal caso en su formato INTEGER utiliza 4 bytes....); minimamente el formato convenido deberá permitir comunicar cifras entre "0,001" hasta "1999".

-En el caso de que la transferencia de datos sea sin previa conversión (así tenemos implementado en equipos anteriores), la PC leerá en la eeprom del voltímetro (al iniciar la comunicación mediante programa correspondiente) cómo está seteado cada medición (escala, etc.) para luego hacer la conversión..

-El display mostrará cada medición a medida que se pulse alguna tecla (DN o UP). Aparte del formato general de mostar nombre de la medición, valor, y unidad, un segundo formato reducirá esos strings a una fila, y la otra del display lo mostrará como vúmetro. En la eeprom estará determinado qué formato se debe usar para mostrar cada una de las mediciones en el display.

-Un tercer formato, presentará una de las mediciones, la que estará definida como principal (también en la eeprom), con caracteres que ocupan todo el display, para tener una mejor visibilidad de la misma.

Para acceder a ella en el display, podría considerarse como la medición primera a mostrar (luego de alimentar la placa), y con el teclado (DN) luego se barrería en sentido MED1, MED2, .... o con otra tecla (UP) MED16, MED15, etc. (suponemos seteo en eeprom para adquirir los 16 canales dispinibles, caso contrario la MED de mayor número será la indicada en la eeprom).

Se dilucirán alternativas en el diseño, como por ejem. si después de un cierto tot estando observando la med. principal pero en formato general, esta pasa a mostrarse en el formato tamaño agrandado.

PROMEDIACIÓN CANALES ANALÓGICOS

Las mediciones analógicas (hasta 16 para mostrarse en display) se realizarán con una cierta cantidad de entradas al micro (2?, 4?, ....?) para lo cual estarían multiplexadas previamente.

El dato que sale de cada adc (0 a 1023 si fuesen 10 bits) se almacenará en RAM (16 canales x 2 bytes = 32 bytes) directamente, y un segundo juego ocupará otro espacio de RAM de 32 bytes, denominadas Promediadas.

Esta promediación se hará inmediatamente después de la adquisición de tal medición (antes de escalarla al valor a mostrar en display), mientras se le da tiempo al establecimiento del próximo canal a medir.

La promediación tiene el efecto de filtrar ruidos eléctricos en cables y conexiones, pero principalmente promediar cada canal ya que pueden tener una cierta variación en tiempos breves, según el tipo de canal a medir (como ejemplo, una medición de tensión de 220 Vca se realiza con rectificadores y capacitores, y produce una flecha o caída con período de 10 mseg). Entonces a veces si no se filtrara o promediara, se vería continuamente variación indeseable de algún dígito en el display. Pero además se requiere que el "tiempo Tau" de promediación pueda ser independiente en cada canal.

Se indica ahora un método conveniente que venimos realizando en nuestros equipos.

Supongamos que a ese canal lo llamemos PDANT, y que en un instante t(i) dado aparece válida una nueva salida de ese adc PDANT(i), ejemplo 800 en decimal (320 hexa ó 11 0010 0000 bin). Habrá en la ram de valor promediado de la anterior adquisición de dicho canal el valor PDANTprom(i-1), entonces el nuevo valor promediado seguirá la fórmula:

PDANTprom(i) = PDANTprom(i-1) + [(PDANT(i) - PDANTprom(i-1)] \* [t(i) - t(i-1)] / [TPROM]

esto equivale en un circuito eléctrico a una variación exponencial ante un salto discreto en la entrada, donde TPROM será la constante de tiempo Tau (tiempo en que la variación en la salida es el 63% del salto en la entrada).

El factor a la derecha del muliplicador "\*" es una fracción de la unidad, entonces el resultado dará mayor cantidad de bits que el de la medición instantánea.

Esta tiene 10 bits (como el adc), entonces es necesario que el resultado tenga mayor cantidad de bits. Restrigiendo a 16 bits, de forma de usar dos bytes, el factor mencionado no debiera ser menor a 1/64 (=1 / 2^6).

Para facilitar el cálculo en el microprocesador se utilizarán solo factores de la forma 1 / 2^n, y podríamos predifinir que los factores a utilizar serán 1/64, 1/16, 1/4 y 1/1 que significa no promediar (en este caso directamente se utiliza la MEDxx\_(i)).

Entonces las variables MEDxx\_prom(i-1) y MEDxx\_prom(i) deberán ocupar 16 bits c/u, lo mismo será lo necesario para su cómputo.

El factor antes mencionado lo denominamos (factor de) PROMEDIACIÓN.

Ni bien se alimenta la plaqueta, la MEDxx\_prom(i-1) de cada una de las (16) señales analógicas a medir estará en cero, entonces se las inicializará con la adquisición de la primer lectura sin promediar de cada (16) medición.

Para que el objetivo de que con los factores utilizados den una promediación prevista, convendrá que cada ciclo de lectura de las 16 mediciones, sea un tiempo prefijado (en base a las demoras del procesador este tiempo estaría en el orden de 15 a 25 mseg, cuando sean 16 los canales a barrer, y previendo demoras en otras tareas del micro no especificadas aquí). Será conveniente que sea bastante constante, por lo que al comenzar un nuevo período, podría incluirse un retardo tal que todos los ciclos tengan igual duración.

Si como ejemplo resultara 25 mseg, los Tau correspondientes serían 1,6 seg., 400 mseg. y 100 mseg. respectivamente, siendo estos valores adecuados para la necesidad de esta placa.

CONVERSIÓN SALIDA ADC AL VALOR A MOSTRAR EN DISPLAY

La calibración de la escala de cada canal a medir, se realizará por software según un conjunto de valores guardados en la eeprom.

Esto se hará mediante la PC conectada a la plaqueta (mas adelante se indicará si también será necesario poder hacer calibración desde el teclado).

El procedimiento sería como sigue:

Para cada canal en particular (16), una vez ingresado al menú de calibración, y habiendo leído desde la eeprom las características de LIN/CUAD, y de PROMEDIACIÓN del canal, se pondrá una tensión en la entrada correspondiente, para la que en la PC se mostrará en formato decimal un número correspondiaente a la salida promediada de ese adc (entre 0 y 1023, por ejemplo 789 y es para asegurar que la tensión de entrada esté en un valor razonable), y entonces en otro campo el usuario escribirá el valor que debiera mostrar el display una vez calibrado para esa entrada (digamos 1000), luego pulsando INTRO deberán quedar registrado en la eeprom para ese canal (2 bytes para CalADC y 2 bytes para CalDISPLAY), los valores CalADC (=789) y CalDISPLAY (=1000) respectivamente, finalizando así su calibración.

Luego para mostrar el valor adecuado en cada momento en el display, para ese canal, se deberá utilizar la fórmula que corresponda según sea lineal o cuadrática (atributo leído de la eeprom):

-para medición de escala cuadrática (como las mediciones de potencias) la fórmula a aplicar sería:

Px =[adc promediada / CalADC]^2 \* CalDISPLAY

Px =[adc promediada / 789]^2 \* 1000

si adc promediada fuese = 605, entonces Px sería = 588 W

-si la medición es de escala lineal (como mediciones de tensiones) la fórmula a aplicar sería:

Vx =[adc promediada / CalADC] \* CalDISPLAY

Vx =[adc promediada / 789] \* 1000

si adc promediada fuese = 801, entonces Vx sería = 1015 V

Al alimentar la plaqueta, el microcontrolador leerá de cada canal los valores en eeprom, y calculará para almacenar en ram CalDISPLAY / CalADC ^2 para las cuadráticas, y CalDISPLAY / CalADC para las lineales, con la resolución (cantidad de bits) suficiente.

USO DE LA EEPROM

La calibración ocuparía así, para cada canal (de 16) 4 bytes. Pero además al menos se dispondrían dos juegos de calibración (uno seteado antes de poner el equipo en marcha, y otro al momento de calibrar), sumando 8 bytes.

Además por canal consumirá 10 bytes para nombre y unidad de medición, un byte para modo LIN/CUAD y para PROMEDIACIÓN, y uno más para fondo escala vúmetro cuando corresponda. O sea unos 8 bytes más 12 bytes por canal.

FORMATOS PRESENTACIÓN DISPLAY

Al considerar el tipo de display a utilizar, se prevé que la presentación sea de las siguientes formas:

0,001 hasta 1,999; 0,01 hasta 19,99; 0,1 hasta 199,9; 1 hasta 1999.

El fondo de escala de 3 1/2 dígitos (apróx. 2000) debe permitir las cuatro posiciones posibles de ubicación del punto decimal indicados.

Y la resolución (ubicación del punto decimal) quedará determinado al momento de realizar la calibración para cada canal, y para mediciones lineales tendrá el punto en una única posición para cada canal sin importar el valor de salida de su adc, mientras que para mediciones cuadráticas habrá dos modos (fijados en eeprom desde la PC, para cada canal cuadrático en particular): uno con punto fijo como las lineales, y otro de máxima resolución. O sea resumiendo definiremos como Lineal, Cuadrática, y Cuadrática Máxima Resolución (este aplicable solo a las cuadráticas).

Aparte el hecho de no tener preset (mecánico o electrónico) antes de la entrada del adc, hará que la resolución del canal no sea aprovechada al máximo, porque para cierto fondo de escala en el display, la salida de ese adc puede estar cerca de 1023 (máximo aprovechamiento), o puede estar más cerca del 0; para el diseño supondremos que tenga capacidad para mostrar a fondo de escala cuando el adc esté en el orden de 128.

Veremos los cálculos y las posibles presentaciones en el display.

Mediciones Lineales con fondo de escala adc cercano a 1024:

ADC promediado Escala 2000 Escala 200,0 Escala 20,00 Escala 2,000

Eeprom CalDISPLAY p/ CalADC=1024 2000 200,0 20,00 2,000

ADC 1022 Cálculos 1996,094 199,6094 19,96094 1,996094

Mostrar en Display (Lineal) 1996 199,6 19,96 1,996

ADC 0001 Cálculos 1,953 0,1953 0,01953 0,001953

Mostrar en Display (Lineal) 2 0,2 0,02 0,002

Mediciones Lineales con fondo de escala adc cercano a 128:

ADC promediado Escala 2000 Escala 200,0 Escala 20,00 Escala 2,000

CalDISPLAY si CalADC fuese 128 2000 200,0 20,00 2,000

Eeprom CalDISPLAY p/ CalADC=1024 16000 1600 160 16

ADC 0127 Cálculos 1984,375 198,4375 19,84375 1,984375

Mostrar en Display (Lineal) 1984 198,4 19,84 1,984

ADC 0001 Cálculos 15,625 1,5625 0,15625 0,015625

Mostrar en Display (Lineal) 16 1,6 0,16 0,016

Mediciones Cuadráticas con fondo de escala adc cercano a 1024:

ADC promediado Escala 2000 Escala 200,0 Escala 20,00 Escala 2,000

Eeprom CalDISPLAY p/ CalADC=1024 2000 200,0 20,00 2,000

ADC 1022 Cálculos 1992,195 199,2195 19,92195 1,992195

Mostrar en Display (Cuadrático) 1992 199,2 19,92 1,992

Mostrar en Display (Cuad Máx Res) 1992 199,2 19,92 1,992

ADC 0023 Cálculos 1,009 0,1009 0,01009 0,001009

Mostrar en Display (Cuadrático) 1 0,1 0,01 0,001

Mostrar en Display (Cuad Máx Res) 1,009 0,101 0,010 0,001

ADC 0001 Cálculos 0,00191 0,000191 0,0000191 0,00000191

Mostrar en Display (Cuadrático) 0 0,0 0,00 0,000

Mostrar en Display (Cuad Máx Res) 0,002 0,000 0,000 0,000

Mediciones Cuadráticas con fondo de escala adc cercano a 128:

ADC promediado Escala 2000 Escala 200,0 Escala 20,00 Escala 2,000

CalDISPLAY si CalADC fuese 128 2000 200,0 20,00 2,000

Eeprom CalDISPLAY p/ CalADC=1024 128000 12800 1280 128

ADC 0127 Cálculos 1968,872 196,8872 19,68872 1,968872

Mostrar en Display (Cuadrático) 1969 196,9 19,69 1,969

Mostrar en Display (Cuad Máx Res) 1969 196,9 19,69 1,969

ADC 0003 Cálculos 1,0986 0,10986 0,010986 0,0010986

Mostrar en Display (Cuadrático) 1 0,1 0,01 0,001

Mostrar en Display (Cuad Máx Res) 1,099 0,110 0,011 0,001

ADC 0001 Cálculos 0,1221 0,01221 0,001221 0,0001221

Mostrar en Display (Cuadrático) 0 0,0 0,00 0,000

Mostrar en Display (Cuad Máx Res) 0,122 0,012 0,001 0,000

Observaciones:

* el decimal de menor peso mostrado en el display deberá estar redondeado según decimales no mostrados.
* Cuando el adc (promediado) da 1023 o mayor, se mostrará siempre >>>>.  
  También cuando la indicación calculada fuese mayor a 1999, ó 199,9, ó 19,99, ó 1,999 (según escala a mostrar en el display) se mostrará >>>>.
* Las diferencias de presentación entre Cuadrático y Cuad Máx Res se presentan solo con los valores bajos de salidas de los adc; justamente la última opción, para un canal dado, muestra diferente posición del decimal para mostrar decimales ocultos por la Cuadrática.
* La fila "Eeprom CalDISPLAY p/ CalADC=1024" es solo una referencia para el caso en que en la eeprom se anotara en cada canal el valor a mostrar si el adc fuese 1024; pero ese método no se utilizaría por requerir más espacio de eeprom. Más adelante se indica alguna forma posible de aprovechar la Eeprom.

GUARDADO ESCALAS EN EEPROM

Para cada canal analógico a medir (16 posibles), una vez promediada la medición (16 bits en ram), debe ser convertida al formato adecuado para presentar en el display.

Para eso realizará una operación de multiplicar esa ram una vez si se trata de canal analógico lineal, o dos veces (para computar el cuadrado) para canal cuadrático, por otro valor en ram que a la vez corresponde a una operación con bytes ubicados en la eeprom.

Dicha operación es el resultado de los valores CalDISPLAY / CalADC para lineales, o CalDISPLAY / CalADC ^2 para cuadráticos.

Los valores CalDISPLAY y CalADC están guardados en la eeprom para cada canal.

Valores para CalADC: será un valor entero comprendido entre 1 y 1022 (siempre en suposición de micro con adc de 10 bits), luego requiere 10 bits. Se utilizarán 2 bytes para ello.

Valores para CalDISPLAY: será una expresión que pueda indicar valores desde 0 hasta 1999 (11 bits) y además que pueda tener el punto decimal uno, dos o tres posiciones hacia la izquierda (2 bits). Incluir además un bit más (o sea 14 en total) para indicar que en el caso de medición lineal, pueda incluirse el signo - (menos), y en tal caso los 11 bits antes mencionados expresarían valores desde 0 hasta 1023 (000 0000 0000 hasta 011 1111 1111) y -1024 hasta -1 (100 0000 0000 hasta 111 1111 1111).

Lo mencionado hasta aquí representan 4 bytes para un juego de CalADC y CalDISPLAY. Pero en total comprenden 24 bits útiles. Por lo que pudieran acomodarse en conjunto en 3 bytes, en caso de tener que ahorrar espacio de eeprom.

Datos en EEPROM asociados a los canales analógicos

Para definir cada canal analógico se ocuparán por una parte los siguientes 12 bytes:

Nombre medición: 7 bytes (7 caracteres ASCII)

Nombre unidad: 3 bytes (3 caracteres ASCII)

Lineal/Cuad/Máx Res: 2 bits (00: Lin; 10: Cuad; 01: Lin Máx Res)

Promediación: 2 bits (00: sin promediar; 01: 1/4; 10: 1/16; 11: 1/64)

Caracteres Vúmetro Normal: 4 bits (0000\_1111: cant. caracteres formato izquierda)

Modo Numérico / Vúmetro: 1 bit (0: Numérico; 1: Vúmetro)

Fondo Escala Vúmetro: 7 bits (como valor display sin punto, dividido 16)

Y además los siguientes 6 u 8 bytes para la correspondiente escala:

1er juego CalADC: 2 bytes

1er juego CalDISPLAY: 2 bytes (o 3 bytes para los 4 en conjunto con CalADC)

2do juego CalADC: 2 bytes

2do juego CalDISPLAY: 2 bytes (o 3 bytes para los 4 en conjunto con CalADC)

En total 18 o 20 bytes por canal, o sea entre 288 y 320 bytes en eeprom para los 16 canales analógicos. Tener pen cuenta que se utilizará eeprom para otras definiciones también.

PRESENTACIÓN EN MODO VÚMETRO

Para mostrar en display la medición de cada canal analógico en particular, habrá dos modos (según esté seteado en la eeprom): uno numérico (se verá más adelante cómo organizarlo) y otro como vúmetro de barra horizontal, suponiendo display de 2 por 16 caracteres.

En el modo vúmetro:

La linea superior mostraría los 7 caracteres del Nombre de la medición, un espacio libre o con caracter ":" (dos puntos), los tres o cuatro o cinco caracteres que comprenden los números y cuando corresponda el punto decimal de la medición, y los últimos tres caracteres de la derecha corresponderían al nombre de la unidad.

En la línea inferior, simultaneamente con la indicación de arriba, se utilizarán los 16 caracteres de la siguiente manera:

Los 16 caracteres estarán divididos en dos grupos, uno de la izquierda y el otro de la derecha.

La cantidad de caracteres del de la izquierda (0 hasta 15) está definido por los 4 bits de "Caracteres Vúmetro Normal". Todos los restantes serán los de la derecha.

Además en cada grupo habrá posible dos tipos de caracteres.

El de la izquierda se subdivide en dos tipos que de izquierda a derecha son "...." (puntos) y "||||" (líneas verticales).

El de la derecha se subdivide en dos grupos: a la izquierda son"\_\_\_" (línea o guión bajo) y a la derecha "###" (numeral).

Esta definición es para dar el aspecto si la señal a mostrar pasó o no de tal límite (si invade solo el sector izquierdo o si alcanza el derecho).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| . | . | . | . | . | . | . | . | . | . | . | \_ | \_ | \_ | \_ | \_ |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | | | | | | | | | | | | . | . | . | . | . | \_ | \_ | \_ | \_ | \_ |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | | | | | | | | | | | | | | | | | | | | | | \_ | \_ | \_ | \_ | \_ |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | | | | | | | | | | | | | | | | | | | | | | # | # | # | \_ | \_ |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| | | | | | | | | | | | | | | | | | | | | | | # | # | # | # | # |

Para calcular cuántos caracteres cambian respecto del vúmetro vacío mostrado en la parte superior se procede así:

El Fondo Escala Vúmetro, supongamos que hemos definido que todos los 16 caracteres deben estar cambiados cuando esa medición indique 1000 en la línea superior, sin tener en cuenta la posición del punto decimal (esto es que 1000 vale igual para indicaciones 100.0, 10.00 o 1.000).

Calculamos 1000 / 16 = 62,5, redondeando los siete bits estarán en 64 (100 0000).

Si el valor 1000 corresponde a 16 caracteres, para una medición en un instante dado MED(i), menor de 1000, la cantidad de caracteres cambiados serán MED(i) / 16.

EEPROM SOLO REFERENCIA DE PLACA P0568

Lo mostrado en adelante se tomará solo como referencia de cómo tenemos implementado la organización en eeprom de una placa que desarrollamos una década atrás.

Ahora los datos y funciones a almacenar y acceder serán solo parecidos.

Para enumerar algunas de las diferencias:

En el proyecto actual se prevee 4 promediaciones distintas en lugar de 8.

Se preveen no más de 16 canales analógicos como voltímetro.

Las escalas ahora no será de valores discretos convenientes prefijados, sino adaptadas según nivel de señal de ingreso (que demandará entre 6 y 8 bytes cor canal).

Ahora se estima utilizar un display con menor resolución (en aquel en el display se muestran simultaneamente 4 mediciones en el sector izquierdo y una ampliada en el derecho.

Los rótulos serán de menor énfasis. Se destinaban a enunciar modelo del equipo e indiacción de la medición ampliada.

etc.....

Tener en cuenta que el uso de la eeprom hasta lo comentado en las notas anteriores, incluye solo lo referente a funciones como voltímetro; pero se prevee puedan desarrollarse también funciones de control como encendido y apagado del equipo, etc., a definirse más adelante y que requerirán su espacio.

Mapa EEPROM en P0568-Rev1

Se utilizan 1 KB (Kbytes) para indicar nombre de mediciones, etc, que se utilizan para mostrar en el display y para calcular según escalas, etc.

Están ordenados en la siguiente secuencia:

000\_ROTULO 1 (8 B)

008\_ROTULO 2 (23 B)

01F\_NPAN (1 B, solo los 3 lsb). Indica cuántos grupos x4 analógicas se utilizan.

020\_MED AMPLIADA (1 B, solo los 5 lsb). Indica nro. de la medición ampliada,

021\_MED1 NOMBRE (7 B)

028\_MED1 UNIDAD (3 B)

02B\_MED1 LINEALIDAD Y ESCALA (1 B. EX: Cuadrática. FX: Lineal. X0 hasta XF: indica para ADC 1000/1023 = 1000, 100, 10, 10000, 2000, 200, 20, 2, 4000, 400, 40, 4, 8000, 800, 80, 8)

02C\_MED1 PROMEDIACION (1 B, solo los 3 lsb. Promediación:1, 4, 8, 16, 32, 64, 128 o 256)

02D\_MED2 NOMBRE (7 B)

034\_MED2 UNIDAD (3 B)

037\_MED2 LINEALIDAD Y ESCALA (1 B)

038\_MED2 PROMEDIACION (1 B)

………………………………………………………………………………………………..

0D5\_MED16 NOMBRE (7 B)

0DC\_MED16 UNIDAD (3 B)

0DF\_MED16 LINEALIDAD Y ESCALA (1 B)

0E0\_MED16 PROMEDIACION (1 B)

0E1\_MED17 NOMBRE (7 B)

0E8\_MED17 UNIDAD (3 B)

0EB\_MED17 LINEALIDAD Y ESCALA (1 B)

0EC\_MED17 PROMEDIACION (1 B)

…………………………………………………………………………………………….…..

195\_MED32 NOMBRE (7 B)

19C\_MED32 UNIDAD (3 B)

19F\_MED32 LINEALIDAD Y ESCALA (1 B)

1A0\_MED32 PROMEDIACION (1 B)

1A1\_LOGICAS8-1\_SAL/ENT (1 B. Con “0” se indican las que son salidas desde el uC)

1A2\_LOGICAS8-1\_VAL\_INICIO (1 B. Con “0” inicializa con el último valor, 1 valor prefijado)

1A3\_LOGICAS8-1\_VAL\_PREFIJADO (1 B. Valor prefijado ALTO o BAJO)

1A4\_LOGICAS8-1\_VAL\_CORTE (1 B. Valor guardado antes del corte de energía)

1A5\_LOGICAS16-9\_SAL/ENT (1 B. Con 0 se indican las que son salidas desde el uC)

1A6\_LOGICAS16-9\_VAL\_INICIO (1 B. Con “0” inicializa con el último valor, 1 valor prefijado)

1A7\_LOGICAS16-9\_VAL\_PREFIJADO (1 B. Valor prefijado ALTO o BAJO)

1A8\_LOGICAS16-9\_VAL\_CORTE (1 B. Valor guardado antes del corte de energía)

1A9\_LOGICA1 NOMBRE (7 B)

1B0\_LOGICA2 NOMBRE (7 B)

………………………………………………………………………………………………..

20B\_LOGICA15 NOMBRE (7 B)

212\_LOGICA16 NOMBRE (7 B)

219\_CONFIG\_PWM1 (1B. 0 = 0V, 1 = Máx V, 2 = ultimo valor)

220\_VALOR\_PWM1 (1B.

221\_CONFIG\_PWM2 (1B. 0 = 0V, 1 = Máx V, 2 = ultimo valor)

222\_VALOR\_PWM2 (1B.

PROTOCOLO DE COMUNICACIÓN CON UNA PC

Se necesita acceder a la eeprom de la placa para diversas tareas que afectarán lo que se muestre en el display y otras como:

-Fijar eventualmente algún nombre referente al modelo y tipo de equipo

-Fijar nombre y unidad de cada medición.

-Fijar medición principal (que la mayor parte del tiempo se mostrará en tamaño ampliado).

-Fijar modo de presentación numérico o vúmetro a c/u.

-Fijar respuesta Lineal/Cuadrática a c/u.

-Fijar promediación a c/u.

-Calibrar su respectiva escala.

-Posibles valores para funciones de control sobre el equipo a indicarse más adelante.

-etc.

Algunos parámetros (calibraciones, etc.) es factible que más adelante definamos poder acceder y setear aparte desde teclado del voltímetro...

La manera de comunicación es tal que la placa siempre está a la espera de algún mensaje (desde la PC), y una vez detectado y reconocido un mensaje, devolverá la correspondiente respuesta. No se prevee que la placa deba iniciar por sí sola una comunicación (como los traps en SNMP).

Para la comunicación con la PC se tomará (solo) como referencia el utilizado en la placa P0568, el cual conforma un modo robusto al contener códigos de seguridad CRC16.

En general está conformado por bytes de comienzo de trama, destino (placa en cuestión), orígen (PC), función, CRC 2 bytes, y byte de final de trama.

También incluye en formato estandar, la inclusión de un byte a continuación de cada uno de los que se deben alterar para no coincidir con bytes de inicio o final de trama.

En el protocolo de referencia mostrado más adelante, la placa destino se definía como 01 hasta 08 (tenía capacidad simultánea de 8 placas), ahora indicaremos un número distinto (por ser distintas sus funciones), podría ser el número 09 ?.

Como código de origen de mensaje desde la PC se utilizaba el A0, y podemos seguir usándolo ahora.

El protocolo mencionado fija formato de la comunicación serie, velocidad, etc. Utilizamos 57600 (en otras ocasiones anteriores, para sistema similar utilizamos 19200).

La idea sería partir de dicho formato, conservando algunos códigos que serían compatibles (como los mensajes y formatos para escribir, o para leer la eeprom, etc.) pero también se agregarían nuevos códigos (distintos de aquellos) que contemplen las necesidades específicas de esta nueva placa (ejm. las calibraciones de escalas son de distinta manera).

Cuando la placa reciba una trama con un código de función distinta al de su conjunto, simplemente debe ignorarla y permanecer preparado para cualquier mensaje futuro.

También en el protocolo de comunicación se debe contemplar que las lecturas de mediciones posibles en el display (información externa a la eeprom), puedan ser leídas desde la PC, las 16 simultáneas con un solo requerimiento, y con códigos distintos para acceder las ram no promediadas, las promediadas, y factible también con un tercer código como quedan procesadas para presentar en el display (un formato ya convertidas de acuerdo a sus respectivas escalas).

El diseño del protocolo completo deberá ir haciendose teniendo en cuenta todas las funciones que se van previendo en la nueva placa.

Sistema conversión SERIAL – SNMP (plaqueta P0568)

El sistema realiza la conversión de una comunicación serial al protocolo TCP/IP servicio SNMP.

La entrada serial está configurada en:

Bits por segundo 57600

Bits de Datos 8

Paridad Ninguno

Bits de parada 1

Control de flujo Ninguno

La velocidad (bits por segundo) debe establecerse en el punto óptimo, debe realizarse mayor cantidad de pruebas, pero por ahora sería esa.

La tensión de la señal serial es de 3,3 [V] .

Las funciones implementadas tiene el siguiente formato:

Comienzo de trama 7E

Nº de Placa destino xx

Nº de Placa origen yy

Nº de Función zz

Checksun rr

Checksun pp

Fin de trama 7F

Nº de Placa: 01 a 08 para cada una de las placas, y A0 para el pedido desde SNMP

Checksun son dos byte CRC16 (Modbus)

7E + xx + yy + 01 + rr + pp + 7F Solicita el número de pantalla actual de la placa Nº

xx.

7E + yy + xx + valor\_pantalla + rr + pp + 7F Responde a la solicitud, con el

número de pantalla actual.

7E + xx + yy + 02 + rr + pp + 7F Solicita un up en el número de

pantalla de la placa Nº xx.

7E + yy + xx + valor\_pantalla + rr + pp + 7F Responde a la solicitud, con el

número de pantalla actual.

7E + xx + yy + 03 + rr + pp + 7F Solicita un down en el número de

pantalla de la placa Nº xx.

7E + yy + xx + valor\_pantalla + rr + pp + 7F Responde a la solicitud, con el

número de pantalla actual.

7E + xx + yy + 04 + rr + pp + 7F Solicita los valores de los 16 canales

digitales y los 32 canales analógicos

sin promediar de la placa Nº xx.

7E + yy + xx + digital\_8\_1 + digital\_16\_9 + analog\_1\_H + analog\_1\_L + …..+ analog\_32\_H + analog\_32\_L + rr + pp + 7F

Responde a la solicitud, con los 2

byte que son los 16 canales digitales

y las 32 mediciones analógicos (2B)

de la placa Nº xx.

7E + xx + yy + 05 + dato\_8\_1 + dato\_16\_9 + rr + pp + 7F

Escribe los canales digitales de la

placa Nº xx.

7E + yy + xx + digital\_8\_1 + dato\_8\_1 + digital\_16\_9 + dato\_16\_9 + rr + pp + 7F

Responde con los valores guardados en los canales digitales de la placa Nº xx, mas los registro de cómo están configurados

7E + xx + yy + 06 + dato\_8\_1 + mascara\_8\_1 + dato\_16\_9 + mascara\_16\_9 + rr + pp + 7F

Escribe los canales digitales Según

la mascara, de la placa Nº xx.

7E + yy + xx + digital\_8\_1 + dato\_8\_1 + digital\_16\_9 + dato\_16\_9 + rr + pp + 7F

Responde con los valores guardados en los canales digitales de la placa Nº xx, mas los registro de cómo están configurados

7E + xx + yy + 07 + Offset\_H + Offset\_L + cantidad\_byteH + cantidad\_byteL + rr + pp + 7F

Lee una cantidad de byte, a partir de

la dirección de Offset de la placa

Nº xx.

7E + yy + xx + Offset\_H + Offset\_L + cantidad\_byteH + cantidad\_byteL + byte\_1 + .... + byte\_n + rr + pp + 7F

Responde con cantidad de byte,

dirección de Offset de la y datos de

la placa Nº xx.

7E + xx + yy + 08 + Offset\_H + Offset\_L + cantidad\_byteH + cantidad\_byteL + byte\_1+….+ byte\_n + rr + pp + 7F

Escribe una cierta cantidad de byte,

a partir de la dirección de Offset

de la placa Nº xx.

7E + yy + xx + Offset\_H + Offset\_L + cantidad\_byteH + cantidad\_byteL + rr + pp + 7F

7E + xx + yy + 0A + rr + pp + 7F Lee los valores con que se

encuentran el PWM1 y PWM2 en la

placa Nº xx.

7E + yy + xx + valor\_pwm1 + valor\_pwm2 + rr + pp + 7F

Responde con los valores de los dos

PWM

7E + xx + yy + 0B + valor\_pwm1 + valor\_pwm2 + rr + pp + 7F

Escribe los valores en los PWM1 y

PWM2 en la placa Nº xx.

7E + yy + xx + valor \_pwm1 + valor\_pwm2 + rr + pp + 7F

Responde con los valores que se

escribieron los PWM de la placa N°

xx

7E + xx + yy + 11 + N°Canal\_analog\_leer\_prom + rr + pp + 7F

Solicita el número de canal a leer

(entre 0 y 31), de la placa xx

7E + yy + xx + N°Canal\_analog\_leer\_prom + Valor\_canal\_H + Valor\_canal\_L + rr + pp + 7F

Retorna con el número del canal

leído, y los dos byte pertenecientes

al valor del canal, (High – Low)

7E + xx + yy + 12 + rr + pp + 7F Solicita el valor promediado de los

32 canales analógicos de la placa xx

7E + yy + xx + analog\_1\_H + analog\_1\_L + …..+ analog\_32\_H + analog\_32\_L + rr + pp + 7F

Retorna con el valor promediado de

los 32 canales analógicos,

comenzando con el byte\_H y

después el byte \_L del canal 1

7E + xx + yy + 13 + leer\_o\_escribir + N°\_bit + Valor\_canal\_dig + rr + pp + 7F

Leer o escribe un canal digital (con

2 se lee, con 3 escribe) de la

placa xx. N°\_bit (de 1 a 16).

Valor\_canal\_dig (0 o 1)

7E + yy + xx + leer\_o\_escribir + N°\_bit + Valor\_canal\_dig + rr + pp + 7F

Retorna con un byte en el cual

comenta la operación realizada, un

byte con el número del bit trabajado,

un byte con el valor actual del canal

digital.

7E + xx + yy + 14 + rr + pp + 7F Lee el grupo digital conformado por

los bits 3, 4, 5 y 6.

7E + yy + xx + Valor\_grup\_dig + rr + pp + 7F Retorna con un byte con el valor

actual del grupo formado por los bits

3, 4, 5 y 6 (de 0 a 15). Con el

siguiente formato L3-L4-L5-L6

7E + xx + yy + 15 + Valor\_grup\_dig + rr + pp + 7F

Escribe el grupo digital

conformado por los bits 7, 8 y 11

Valor\_grup\_dig (0 o 7). Con el

formato (L7-L8-L11) según un

t (200 mseg).

7E + yy + xx + AA + rr + pp + 7F Retorna en un byte el valor AA

si la operación de escritura se

realizo con exito.

Si se realiza la operación a una esclava, internamente no realiza nada, pero envía la misma repuesta.

7E + xx + yy + 16 + rr + pp + 7F Leer el grupo digital conformado

por los bits 7, 8 y 11

7E + yy + xx + Valor\_grup\_dig + rr + pp + 7F Retorna con un byte el valor actual

del grupo formado por los bits 7, 8 y

11 (0 o 7). Con el formato (L7-L8-

L11)

7E + xx + yy + 17+ Valor\_grup\_dig + rr + pp + 7F

Escribe el grupo digital

conformado por los bits 9 y 10

Valor\_grup\_dig (0 o 3). Con el

formato (L9-L10).

7E + yy + xx + Valor\_grup\_dig + rr + pp + 7F

Retorna con un byte el valor actual

del grupo formado por los bits 9 y

10. Valor\_grup\_dig (0 o 3). Con el

formato (L9-L10)

7E + xx + yy + 18 + rr + pp + 7F Leer el grupo digital conformado

por los bits 9 y 10

7E + yy + xx + Valor\_grup\_dig + rr + pp + 7F

Retorna con un byte el valor

actual del grupo formado por los bits

9 y 10. Valor\_grup\_dig (0 o 3). Con

el formato (L9-L10)

7E + xx + yy + 19 + rr + pp + 7F Leer el grupo digital conformado

por todos los bits.

7E + yy + xx + Valor\_grup\_digH + Valor\_grup\_digL + rr + pp + 7F

Retorna con dos byte el valor

correspondiente a todos los canales

digitales. Con el formato (L16-L1)

DIAGRAMA CONTROL GENERAL (SIN CONSIDERAR FUTURAS FUNCIONES)

Se da una breve explicación sobre cómo se debe interpretar el diagarma del archivo "P0794\_A\_DF-VOLT Y CTRL CON MICRO.pdf".

Se agrupó en ocho bloques, los que están interrelacionados entre sí.

Aquí no se incluyen las funciones como voltímetro, solo las que hacen al Control que podrá realizar la placa.

Las señales indicadas dentro de un rectángulo con un lado menor reemplazado por un ángulo, corresponden con pines de entrada al micro (lógicas Low/High o analógicas adc10b), o con pines de salida del micro (lógicas Low/High).

Las señales encerradas con rectángulos corresponden en general a señales internas en la Ram, o de mensajes de entrada o salida hacia la PC, o bloques de ciertas funciones, o simplemente como referencia general.

Los círculos con signos "X" y "+" son para indicar funciones lógicas AND y OR respectivamente.

Los símbolos como OR Exclusiva se comportan como tal. Idem los símbolos INVERSOR.

Las señales encerradas en rectángulos con sus lados menores redondeados, representan cantidades de bits (b) o de Bytes (B) que deberán almacenarse en Eeprom, las que serán seteadas por funciones desde el teclado local, o desde una PC.

Más adelante se explicarán en detalle cada una de las señales ya mencionadas, como la posible implementación de nuevas funciones.

DIAGRAMA CONTROL GENERAL (EXPLICACIÓN DETALLADA)

1- Bloque ENCENDIDO:

EA\_IN: pin de entrada lógica al uC a un bloque con retardos. Si la salida Ram EA\_OUT está en Low, y EA\_IN permanece en High continuamente durante un tiempo EA\_TOT segundos (0 a 255), entonces EA\_OUT conmuta a High.

Si la salida Ram EA\_OUT está en High, y EA\_IN permanece en Low continuamente durante un tiempo AA\_TOT segundos (0 a 255), entonces la Ram EA\_OUT conmuta a Low.

FORZAR\_ON: si se recibe un mensaje desde la PC decodificado TCMD\_ON High mientras el bit en Eeprom HAB\_TCON está en High, entonces la señal que sigue será un High sin importar estado EA\_OUT.

Si se recibe un mensaje desde la PC decodificado TCMD\_ON Low, entonces la señal que sigue permanece en el estado EA\_OUT o en el estado que corresponda si está actuando TCMD\_OFF.

FORZAR\_OFF: si se recibe un mensaje desde la PC decodificado TCMD\_OFF High, mientras HAB\_TCOFF está en High, entonces la señal que sigue será un Low sin importar estado EA\_OUT.

Si se recibe un mensaje desde la PC decodificado TCMD\_OFF Low, entonces la señal que sigue permanece en el estado EA\_OUT salvo que esté siendo modificada por TCMD\_ON.

ENC SINO MODO ENC: los lugares en Eeprom E\_MODO y MAN\_SINO tienen 3 bits c/u: bit de inicio, bit memorizar y bit de estado. En cada caso, si al inicializarse el uC (alimentación energía) el bit memorizar se encuentra en Low, entonces el bit de inicio se copia en el bit de estado. En todo momento posterior, el bit de estado podrá ser modificado desde el teclado local (que acciona sobre el equipo y/o sobre el display), el correspondiente a E\_MODO se pone en High cuando desde el teclado se instruye MODO AUTOMÁTICO, y en Low cuando se instruye MODO MANUAL; el bit de estado de MAN\_SINO se pone en High cuando se instruye MAN\_SI, y se pone Low cuando se instruye MAN\_NO.

Lugar en Ram MODO\_AUT: tendrá el mismo valor (High o Low) que el bit de estado de E\_MODO.

Lugar en Ram ENC\_OUT: estará en Low en cualquiera de los siguientes casos:

a- pin de entrada VGGNEG y bit en Eprom HAB\_VGGN simultáneos en High;

b- pin de entrada ENC\_EXT estado inverso al bit Eeprom EEXT\_POL, mientras bit Eeprom HAB\_EEXT esté en High;

c- bit en Eeprom estado E\_MODO High simultáneo con salida bloque FORZAR\_ON/OFF en Low;

d- bit en Eeprom estado E\_MODO Low simultáneo con bit en Eeprom estado MAN\_SINO en Low;

Si no se cumple ninguno de estos cuatro casos, el bit en Ram ENC\_OUT será High.

Bit en Ram RES\_LATCH: tendrá polaridad inversa a bit en Ram ENC\_OUT.

Pin de salida lógica del uC HAB\_12V: será de la misma polaridad que el bit en Ram ENC\_OUT.

Espacio en Eeprom ocupado por el bloque: 2 Bytes + 11 bits.

2- Bloque >>TEMP:

>>TRM: pin entrada lógica al uC a un bloque Antirrebote.

Antirrebote: realiza la función de filtrar y cancelar el efecto de rebotes que puede tener un elemento mecánico como un pulsador (en este caso un termostáto). Comprende tiempos menores a 200 mseg. Su Ram estará en High o en Low cuando esté confirmado en tal estado, después de haber sido procesado por Antirrebote, la entrada >>TRM.

CMP\_>>T: bloque que realiza la comparación entre el valor decimal en una Ram correspondiente a una medición de temperatura como la mostraría el display (o sea promediada o no), y otro valor decimal ubicado en la Eeprom. Cuando el valor de la medición de temperatura es mayor o igual al de la Eeprom, dará una salida en High, caso contrario en Low.

ADC\_REF\_>>T: cuatro bits en Eeprom para designar cuál de los 16 canales analógicos a medir en el display corresponde a la temperatura. Los valores de esos cuatro bits de 0 a 15 corresponderán a canales analógicos desde 1 hasta 16 respectivamente.

>>T\_TEMP: un byte en Eeprom cuya representación decimal de 0 a 255 es el valor con el que se compara la medición en ADC\_REF\_>>T en el bloque CMP\_>>T.

En el caso de asignarse el valor "0", entonces CMP\_>>T entregará un Low independientemente del valor de la temperatura a medir y comparar.

>>T\_HIST: cuatro bits en Eeprom cuya representación de 0 a 15 asignará una histéresis de 0 a 30 °C (o sea valores decimales 0, 2, 4, ... 28, 30).

Tendrá efecto en caso que CMP\_>>T esté en High, y lo hará conmutar a Low cuando verifique que la medición de ADC\_REF\_>>T sea menor al valor de (>>T\_TEMP - >>T\_HIST).

Sumador de >>TRM y >>T\_TEMP: dará un High cuando Antirrebote y el bit en Eeprom HAB\_>>TRM estén simultaneamente en High, o cuando CMP\_>>T y el bit en Eeprom HAB\_>TADC estén simultaneamente en High.

HAB\_>>T: bit en Eeprom, que estando con el Sumador mencionado ambos en High, hará que la Ram >>T esté en High.

LATCH>>T: es un latch cuya salida se setea a High cuando >>T y el bit en Eeprom HAB\_L>T estén simultaneamente en High, y se resetea cuando RES\_LATCH esté en High (y cuando se inicializa el uC). RES\_LATCH tiene prioridad ante el seteo.

Espacio en Eeprom ocupado por el bloque: 1 Byte + 12 bits.

3- Bloque >>ROE:

>>R: pin entrada lógica al uC.

CMP\_>PR: bloque que realiza la comparación entre el valor decimal en una Ram correspondiente a una medición de PR como la mostraría el display (o sea promediada o no), y otro valor decimal ubicado en la Eeprom. Cuando el valor de la medición de PR es mayor o igual al de la Eeprom, dará una salida en High, caso contrario en Low.

ADC\_REF\_>PR: cuatro bits en Eeprom para designar cuál de los 16 canales analógicos a medir en el display corresponde a la PR. Los valores de esos cuatro bits de 0 a 15 corresponderán a canales analógicos desde 1 hasta 16 respectivamente.

PR\_>PR: un byte en Eeprom cuya representación decimal de 0 a 255 es el valor con el que se compara la medición en ADC\_REF\_>PR en el bloque CMP\_>PR.

En el caso de asignarse el valor "0", entonces CMP\_>PR entregará un Low independientemente del valor de la PR a medir y comparar.

Suma de >>R y CMP\_>PR: se obtiene un High cuando están simultaneamente en High >>R y el bit en Eeprom HAB\_>>R, o cuando están simultaneamente en High CMP\_PR y el bit en Eeprom HAB\_>PR.

HAB\_>>R: bit en Eeprom, que estando con el Sumador mencionado ambos en High, hará que la Ram >>R esté en High.

LATCH>>R: es un latch cuya salida se setea a High cuando >>R y el bit en Eeprom HAB\_L>R estén simultaneamente en High, y se resetea cuando RES\_LATCH esté en High (y cuando se inicializa el uC). RES\_LATCH tiene prioridad ante el seteo.

Espacio en Eeprom ocupado por el bloque: 1 Byte + 8 bits.

(Obs.: queda por definirse aún si PR\_>PR ocuparía un Byte más, para extender el rango de este a más de 255.

4- Bloque NTC BIAS:

El bit en Ram BIAS estará en High solo si se cumple simultaneamente todo lo siguiente:

a- que el bit en Ram ENC\_OUT esté en High;

b- que al menos esté en Low el bit en Ram >>T o el bit en Eeprom >T\_INH\_BIAS;

c- que al menos esté en Low el bit en Ram >>R o el bit en eeprom >R\_INH\_BIAS.

Luego BIAS va a un bloque de retardo de un seg. que funciona así:

Estando BIAS suficiente tiempo en Low, entonces también lo estarán el bit en Ram NTC\_BIAS y el pin de salida lógica del uC BIAS\_NTC.

Luego cuando pase el bit BIAS a High, inmediatamente pasa a High el pin BIAS\_NTC, y después de permanecer BIAS 1 seg. en High, conmutará a High el bit NTC\_BIAS.

En la ocasión en que el bit BIAS conmute a Low, inmediatamente pasará a Low el bit NTC\_BIAS, y 1 seg. después de permanecer BIAS en Low, conmutará a Low el pin del uC BIAS\_NTC.

Espacio ocupado en Eeprom: 2 bits.

5- Bloque <<VENT:

REF\_NTC: es un bloque que entrega una referencia de tensión (2 bytes en Ram) para usarse luego en los bloques CMP\_NT1 y CMP\_NT2. Tal referencia se obtiene de una curva a definir por tres puntos VT(i)= f [T(i)].

La entrada a la curva será luego una medición de temperatura (2 bytes en Ram) del canal ADC\_REF\_NTC con su valor procesado según promediación, como para mostrar en display.

En otra oportunidad se indicará cómo resolver la ecuación de tal curva.

ADC\_REF\_NTC: cuatro bits en Eeprom para designar cuál de los 16 canales analógicos a medir en el display corresponde a la temperatura de referencia. Los valores de esos cuatro bits de 0 a 15 corresponderán a canales analógicos desde 1 hasta 16 respectivamente.

T1; T2; T3: temperaturas que definen tres puntos en la ordenada de la curva en REF\_NTC. Un byte c/u en Eeprom.

VT1; VT2; VT3: tensiones que definen tres puntos en la abscisa de la curva en REF\_NTC. Dos bytes c/u en Eeprom.

NTC\_HIST: un byte en Eeprom para definir la histéresis (de tensión) de los comparadores.

NTC1; NTC2: dos entradas a adc independientes en el uC (no utilizar entradas multiplexadas de los 16 canales analógicos). Cada una mide la tensión de un termistor externo, que luego comparada con REF\_NTC indicará calentamiento excesivo de los NTC ante falla de refrigeración.

CMP\_NT1: bloque que compara la tensión del circuito del NTC1 con la del bloque REF\_NTC; cuando la primera es menor que la segunda, entonces genera un bit High en Ram CMP\_NT1. Necesitaría 3 bytes en Eeprom para programar la escala con que se mide y compara la tensión de NTC1 (y NTC2 por ser iguales los termistores).

NTC\_HIST: estando en High la salida del CMP\_NT1, si se da que que la tensión correspondiente a REF\_NTC más una tensión designada por NTC\_HIST fuese menor que la correspondiente a NTC1, entonces retorna la salida del CMP\_NT1 a Low.

HAB\_NTC: un bit en Eprom, que estando en Low fuerza a los comparadores CMP\_NT1 y CMP\_NT2 para que den un Low sin tener en cuenta la salida de REF\_NTC.

NTC\_BIAS: bit en Ram, que estando en Low fuerza a los comparadores CMP\_NT1 y CMP\_NT2 para que den un Low sin tener en cuenta la salida de REF\_NTC.

<<V1\_FALLA: bit en Eeprom, que estando simultaneamente con CMP\_NT1 en High, dará un High en bit de Ram <<V\_FALLA.

<<V1\_SAL: bit en Eeprom, que estando simultaneamente con CMP\_NT1 en High, dará un High en señal que luego multiplicada con bit de eeprom <<V\_OUT dará High en bit de Ram <<V\_SAL.

<<V\_OUT: bit en Eeprom que estando en High habilita la salida <<V\_SAL al pin del uC <<V\_OUT.

<<V\_POL: bit en Eeprom cuya polaridad indica la polaridad del pin <<V\_OUT en el caso de que <<V\_SAL esté habilitada.

<<V\_P50: bit en Eeprom que habilita un High en Ram <<V\_P50 cuando haya un High en <<V\_FALLA.

<<V\_INH: bit en Eeprom que habilita un High en Ram <<V\_INH cuando haya un High en <<V\_FALLA.

<<V\_TOT: timer totalizador de segundos que se inicializa si el bit en Eeprom HAB\_L<V está en High cuando el bit en Ram <<V\_INH pasa a High, y finaliza y da un High cuando alcanza el valor del byte en Eeprom <<V\_TOT. En los casos en que el byte <<V\_TOT haya sido configurado en 0x00, o que RES\_LATCH esté en High, o que el bit en Ram <<V\_INH esté Low, el timer permanece en "0" sin contar.

CMP\_NT2: bloque que compara la tensión del circuito del NTC2 con la del bloque REF\_NTC; cuando la primera es menor que la segunda, entonces genera un bit High en Ram CMP\_NT2. Comparte los mkismos 3 bytes en Eeprom para programar la escala con que se mide y compara la tensión de NTC1 (y NTC2 por ser iguales los termistores).

<<V2\_FALLA: bit en Eeprom, que estando simultaneamente con CMP\_NT2 en High, dará un High en bit de Ram <<V\_FALLA.

<<V2\_SAL: bit en Eeprom, que estando simultaneamente con CMP\_NT2 en High, dará un High en señal que luego multiplicada con bit de eeprom <<V\_OUT dará High en bit de Ram <<V\_SAL.

Espacio ocupado en Eeprom en este bloque: 14 bytes + 14 bits.

6- Bloque P50 TEMP

P50T1: pin entrada lógica.

CMP\_T50: bloque que realiza la comparación entre el valor decimal en una Ram correspondiente a una medición de temperatura como la mostraría el display (o sea promediada o no), y otro valor decimal ubicado en la Eeprom. Cuando el valor de la medición de temperatura es mayor o igual al de la Eeprom, dará una salida en High, caso contrario en Low.

ADC\_REF\_T50: cuatro bits en Eeprom para designar cuál de los 16 canales analógicos a medir en el display corresponde a la temperatura. Los valores de esos cuatro bits de 0 a 15 corresponderán a canales analógicos desde 1 hasta 16 respectivamente.

T50\_TEMP: un byte en Eeprom cuya representación decimal de 0 a 255 es el valor con el que se compara la medición en ADC\_REF\_T50 en el bloque CMP\_T50.

En el caso de asignarse el valor "0", entonces CMP\_T50 entregará un Low independientemente del valor de la temperatura a medir y comparar.

T50\_HIST: cuatro bits en Eeprom cuya representación de 0 a 15 asignará una histéresis de 0 a 30 °C (o sea valores decimales 0, 2, 4, ... 28, 30).

Tendrá efecto en caso que CMP\_T50 esté en High, y lo hará conmutar a Low cuando verifique que la medición de ADC\_REF\_T50 sea menor al valor de (T50\_TEMP - T50\_HIST).

Sumador de P50T1 y T50\_TEMP: dará un High cuando P50T1 y el bit en Eeprom HAB\_P50T1 estén simultaneamente en High, o cuando CMP\_T50 y el bit en Eeprom HAB\_T50 estén simultaneamente en High.

HAB\_P50T: bit en Eeprom, que estando con el Sumador mencionado ambos en High, hará que la Ram P50T esté en High.

Espacio ocupado en Eeprom en este bloque: 1 byte + 11 bits.

7- Bloque SUM P50

P50X: pin entrada uC nivel lógico Low o High.

P50X\_POL: bit en Eeprom que indica qué polaridad en P50X debe ser considerada para comandar reducción de potencia MODO\_P50.

HAB\_P50X: bit en Eeprom que estando en High habilita la entrada P50X como reductora de potencia.

TCMD\_P50: comando recibido desde una PC, tal que si en ese instante tanto el bit en Ram MODO\_AUT como el bit en Eeprom HAB\_TCP50 están en High, habilitará el bit en Ram MODO\_P50.

<<V\_P50: bit en Ram que en High habilita el bit en Ram MODO\_P50.

P50T: bit en ram que estando en High habilita el bit en Ram MODO\_P50.

P50\_MAN: bit en Eeprom seteable desde teclado local, que estando en High significa estado P50\_MAN y conduce al bit en Ram MODO\_P50 en High.

P50\_OUT: pin salida del uC que indica que se está en MODO\_P50.

P50\_POL: bit en Eeprom que indica la polaridad del pin P50\_OUT en el caso del estado MODO\_P50.

Espacio ocupado en Eeprom en este bloque: 5 bits.

8- Bloque HAB RELÉ

RES\_LATCH: bit en Ram que inhibirá HAB\_RELE.

<<V\_INH: bit en Ram que inhibirá HAB\_RELE.

>>T: bit en Ram que inhibirá HAB\_RELE.

>>R: bit en Ram que inhibirá HAB\_RELE.

HAB\_RELE: pin salida uC que estará en High solo si los cuatro bits en Ram mencionados están Low.

Espacio ocupado en Eeprom en este bloque: no utiliza.

Espacio total necesario en eeprom para ocho bloques del Control General aquí descripto: 19 bytes + 63 bits. o sea 27 bytes.