ESCUELA SUPERIOR POLITECNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

PROYECTO DE LA MATERIA DE GRADUACIÓN

MICROCONTROLADORES AVANZADOS

“INTERFAZ CON RECEPTOR DE BUCLE DE 4-20mA EN APLICACIÓN CON DSPIC”

Previo a la obtención del título de:

INGENIERO EN ELECTRICIDAD ESPECIALIZACIÓN ELECTRÓNICA Y AUTOMATIZACIÓN INDUSTRIAL

Presentada por:

Pazmiño Reasco Israel Fabricio

Torres Palaquibay Andrés Fabián

GUAYAQUIL – ECUADOR

2009

DEDICATORIA

A nuestros padres

A nuestros hermanos y familiares

A nuestros profesores

A nuestros compañeros

AGRADECIMIENTO

Agradecemos a Dios.

Agradecemos a nuestros profesores,

por sus enseñanzas diarias en las aulas.

Agradecemos a nuestros padres y familiares,

por toda la comprensión y apoyo incondicional

en nuestras vidas.

A nuestros compañeros que fueron apoyo

durante nuestra vida universitaria.

TRIBUNAL DE GRADUACIÓN

Ing. Carlos Valdivieso
Profesor de la Materia de Graduación

Ing. Hugo Villavicencio
Delegado del Decano

DECLARACIÓN EXPRESA

"La responsabilidad del contenido de este proyecto de graduación nos corresponden exclusivamente, y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL"

(Reglamento de Graduación de la ESPOL)

Israel Pazmiño R.	Andrés Torres.

INDICE GENERAL

INDICE DE ILUSTRACIONES

FIGURAS

TABLAS

LISTA DE ABREVIATURAS

SIMBOLO	SIGNIFICADO
ºC	Grados Centígrados o Celsius
uA	Micro Amperios
mA	Mili Amperios
A	Amperios
V	Voltios
mV	Mili Voltios
MHz	Mega Hertz
KHz	Kilo Hertz
Seg	Segundos
pf	Pico Faradios
Hex	Hexadecimal
Vi	Voltaje de entrada
Vo	Voltaje de salida
VO1	Voltaje de salida opamp 1
VO2	Voltaje de salida opamp 2
Gnd	Tierra
VDC	Voltaje directo continuo
MΩ	Mega Ohmio
J1	Jumper 1
P1	Potenciómetro 1
P2	Potenciómetro 2
P3	Potenciómetro 3
IC2	Circuito integrado 2
IC3	Circuito integrado 3
Vref	Voltaje de referencia
Q1	Transistor 1
VCC	Alimentación
OUT	Salida

RESUMEN

Este proyecto fue implementado con el objetivo de utilizar las nuevas herramientas para el desarrollo de controladores electrónicos como son los DSPIC. En este caso lo utilizamos para el control del proceso de la Maceración de la malta para controlar y monitorear la variable física más importante en este proceso; la temperatura.

El Receptor Bucle implementado recibe la señal de los sensores a largas distancias para poder evitar la interferencia de señales parásitas. Para esto utilizamos el sensor de temperatura LM35 que entrega 10mV/ºC el cual será acondicionado, transmitido y receptado por un controlador que esta diseñado por el DSPIC30F4011 en una de sus entradas analógicas.

En el proceso utilizamos LabVIEW para monitorear la señal del sensor que es recibida por el controlador para luego ser enviada mediante UART del DSPIC hacia LabView que posee el driver VISA utilizado para realizar la lectura y escritura en instrumentación y además tuvimos que utilizar un CI MAX232 que es el encargado de convertir las señales TTL del DSPIC a señales CMOS que lee la computadora.

OBJETIVOS

- Dar a conocer los beneficios de utilizar microcontroladores como los dsPIC en aplicaciones a largas distancias para controlar temperatura.

- Desarrollar aplicaciones en donde se requiera el intercambio de información entre una aplicación desarrollada en LabVIEW y un dispositivo externo al ordenador.

INTRODUCCIÓN

En el ámbito industrial la mayoría de las ocasiones se tiene alejado el circuito que permite recibir la variable que interesa en cada proceso desde donde este ocurre, por lo general se lo tiene en un cuarto de control donde también existen ordenadores para visualizar cada parte del proceso.

En este proyecto, receptamos los datos que son leídos desde un sensor de temperatura en un horno, la señal de este sensor es acondicionada y luego enviada a través de un transmisor de corriente para finalmente llegar a un cuarto de control donde se encontrará un receptor de bucle y el controlador de la temperatura del horno.

Una de las salidas del receptor de bucle es enviada a una de las entradas analógicas del controlador dsPIC30F4011 el cual es programado con el software MikroBasic de Mikroelectrónica.

La curva de temperatura y el proceso tendrán un sistema SCADA y será visualizada en LabVIEW.

1. Sensor de Temperatura LM35

1.1 Descripción

El LM35 es un sensor de temperatura con una precisión calibrada de 1ºC y un rango que abarca desde -55ºC a +150ºC.

El sensor se presenta en diferentes encapsulados de igual forma que un típico transistor con 3 pines, dos de ellos para alimentarlo (VCC y GND) y el tercero (VOUT) nos entrega un valor de tensión proporcional a la temperatura medida por el dispositivo.

1.2 Características

- Precisión de ~1,5ºC (peor caso), 0.5ºC garantizados a 25ºC.
- No linealidad de ~0,5ºC (peor caso).
- Baja corriente de alimentación (60uA).
- Amplio rango de funcionamiento (desde -55º a + 150ºC).
- Bajo costo.
- Baja impedancia de salida.
- Factor de escala lineal de 10mV/ºC

- Su tensión de salida es proporcional a la temperatura, en la escala Celsius. No necesita calibración externa y es de bajo costo. Funciona en el rango de alimentación comprendido entre 4 y 30 voltios.

- La baja impedancia de salida, su salida lineal y su precisa calibración inherente hace posible una fácil instalación en un circuito de control.

1.3 Encapsulado

El más común es el TO-92, una cápsula comúnmente utilizada por los transistores de baja potencia, como el BC548 o el 2N2904.

Este sensor es fabricado por Fairchild y National Semiconductor.

1.4 Acondicionador de señal para LM35

El voltaje de salida en el LM35 es de 10mV por grados centígrados, el rango de salida va desde 0V hasta 1,5V; es decir que antes de ingresar la señal al transmisor de corriente debe ser amplificada. Para nuestro caso de estudio realizaremos una amplificación de 6.67 veces su valor de lectura así obtendremos los 10 voltios necesarios en la entrada del transmisor de corriente.

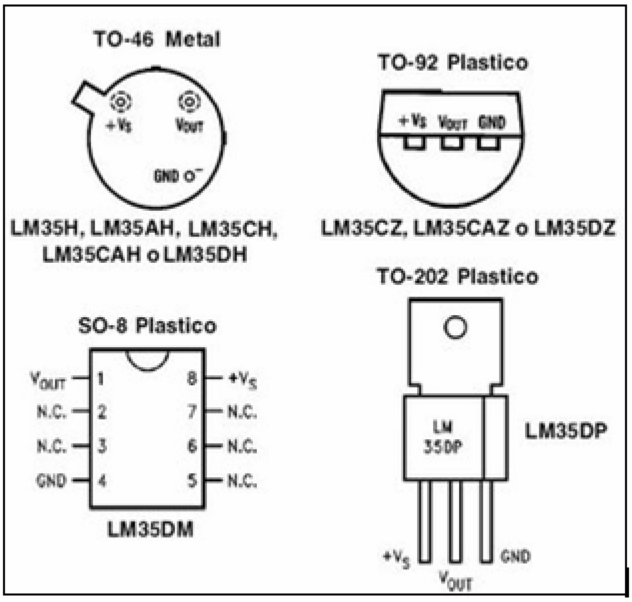


FIGURA 1

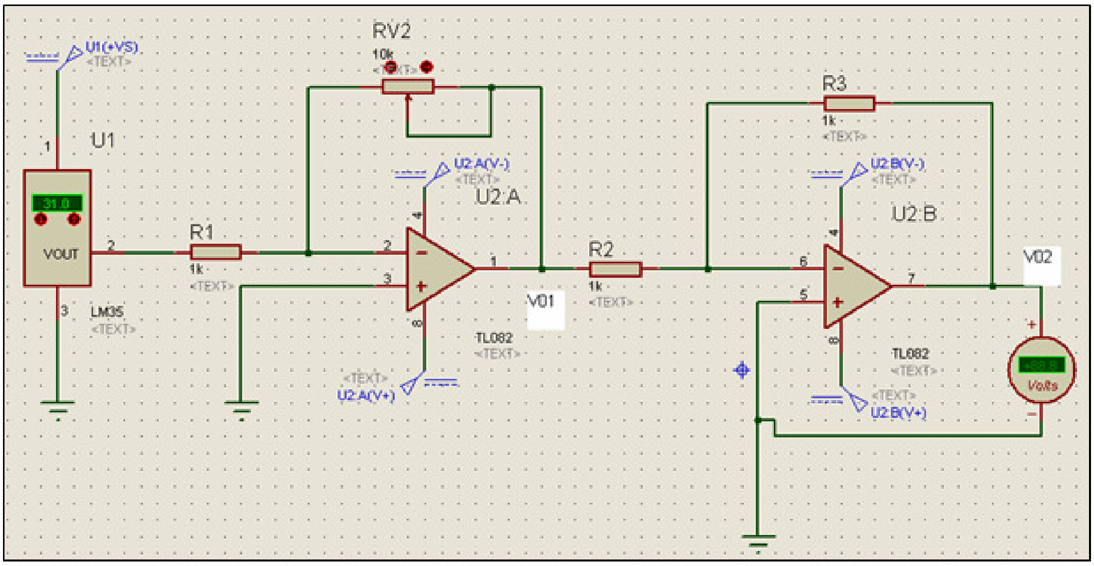


FIGURA 2

2. TRANSMISOR DE CORRIENTE

2.1 ¿Qué es un transmisor de corriente?

El transmisor de corriente es un circuito especial utilizado en la industria para comunicar transductores activos, con controladores PLCs y computadores, de manera confiable.

Se pueden transmitir corrientes normalizadas entre 0 y 25 mA, moduladas por tensiones de entrada también normalizadas, tales como 0- 10mV, 0- 100mV, 0-1V, 0- 5V y 0- 10V.

2.2 ESPECIFICACIONES Y CARACTERÍSTICAS TÉCNICAS

- Entrada de voltaje DC entre 0 y 10 VDC.
- Salida de corriente DC entre 0 y 20mA.
- Alta impedancia de entrada (mayor a 10MΩ).
- Selector de entrada por buffer y amplificador.
- Factor de amplificación variable entre 1 y 100.
- Niveles de escalas máxima y mínima ajustables.
- Máxima impedancia de carga 500Ω.

2.3 Diagrama esquemático del transmisor de corriente

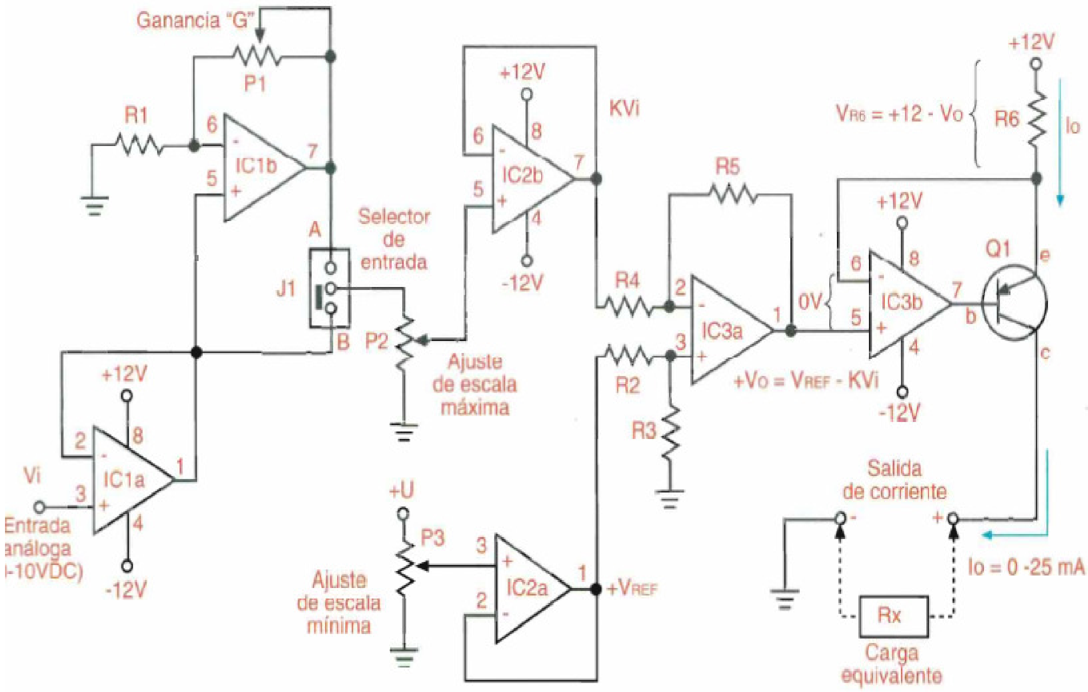


FIGURA 3

2.4 Funcionamiento

Cuando la señal de entrada varía entre 0 y 10VDC, el jumper selector de entrada **J1** debe ser colocado en la posición **B**. De esta forma, el voltaje se acopla a través de un seguidor que lo recibe con alta impedancia de entrada y lo transfiere con ganancia unitaria hacia la siguiente etapa del circuito.

Para todas las tensiones de entrada que estén por debajo de 10 VDC, el circuito cuenta con un amplificador no inversor. Este último se ajusta mediante **P1**.

Una vez que la señal de entrada pasa por el amplificador, se lleva hasta la resistencia variable **P2** que se utiliza para dividir la tensión y de esta manera permitir la calibración de la máxima corriente de salida, para el correspondiente voltaje de entrada.

Por medio de una tercera resistencia variable **P3** se puede calibrar el mínimo valor de corriente de salida asociado a la tensión de entrada más pequeña.

En los dos casos, la tensión de calibración positiva, proveniente de la terminal móvil de las resistencias variables, se acopla usando los seguidores de tensión implementados en el circuito integrado IC2.

El amplificador operacional IC3a opera como un nodo de diferencia y se ocupa de sustraer la tensión equivalente al ajuste de escala máxima (**+Vref**) de la señal variable de entrada y entregar la señal de salida (**+V0**).

La última etapa del circuito es una fuente de corriente constante controlada por voltaje que se ha diseñado para cargas conectadas a tierra, usando un amplificador operacional IC3b y un transistor bipolar PNP con realimentación del emisor.

Para generar la corriente constante en la salida del dispositivo, se conecta la salida del operacional con la base del transistor Q1 para que sea este el que entregue dicha corriente de salida a través de su colector y no el amplificador operacional, que solo tiene 10 mA.

2.5 Montaje y calibración

Una vez que se haya ensamblado el circuito, se debe calibrar con el fin de obtener la respuesta en corriente esperada para los correspondientes voltajes de entrada; para ello efectúe los siguientes pasos:

1. Alimente el circuito a través del conector de potencia con una fuente bipolar de + 12V, 0 y -12V.
2. Colocar el jumper en alguna de sus dos posiciones laterales.
3. Para calibrar el nivel inferior de la corriente de salida, cortocircuite la entrada del circuito y conecte un miliamperímetro entre las dos terminales de salida para leer la corriente, lo que puede estar entre 0 y 4 mA.

4. Finalmente, para calibrar el nivel superior de la corriente de salida, retire el cortocircuito de la entrada, conecte una fuente DC ajustada al valor máximo de voltaje de entrada esperado y ajuste el control de escala máxima hasta leer en el miliamperímetro la corriente de salida deseada

3.RECEPTOR DE BUCLE DE 4 A 20mA.

3.1 Que es un receptor de bucle

Existen diferentes sistemas para comunicar los sensores con los controladores, visualizadores pero dependiendo de la velocidad, el ancho de banda, el factor de rechazo a las interferencias electromagnéticas. También debemos saber elegir el medio de transmisión mas adecuado como por ejemplo: par trenzado, cable coaxial, línea telefónica, la fibra óptica entre otras.

La transmisión de voltaje solo se recomienda a cortas distancias y si a esto le sumamos las tensiones parásitas que se generan en los entornos industriales con lo cual se altera totalmente la medición. Los cables apantallados, los sistemas de tierra, la fibra óptica pueden solucionar este problema pero son sistemas muy costosos y complejos.

Los sistemas de bucle de corriente convierten el voltaje del sensor en una corriente continua que se envía hacia el receptor y este lo lee en forma de voltaje. La comunicación en los sistemas de bucle se han normalizado las corrientes a: 1 – 5mA, 0 – 5mA, 4 – 20mA, 0 – 20mA, 2– 10mA y 10 – 50mA.

Este circuito recibe una corriente continua entre 4 y 20mA la cual se acopla óptimamente y entrega un voltaje proporcional a la misma en dos magnitudes de 0 – 5V y de 0 – 10V, además tiene un led indicador de fallo que me indica si la corriente recibida es menor a 4ma o se rompió el cable de comunicación.

3.2 Especificaciones y características técnicas

- Entrada de corriente optoaclopada.
- Resistencia de entrada menor a 1 Ohmio.
- Corriente de medida entre 4 y 20 mA.
- Dos salidas análogas entre 0 y 5 VDC y entre 0 y 10 VDC.
- Salida digital de fallo.
- Factor de amplificación ajustable.
- LEDs indicadores de comunicación y fallo.

3.3 Diagrama de bloques del receptor de bucle

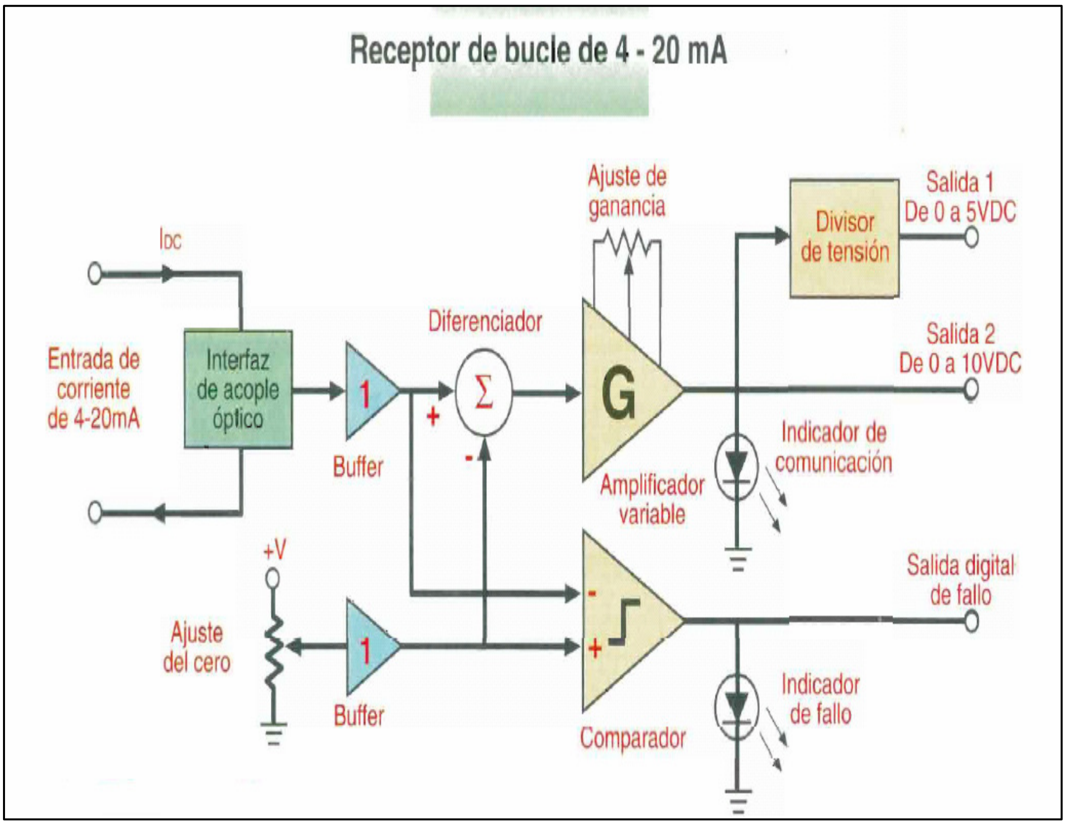


FIGURA 4

3.4 Diagrama esquemático del Receptor bucle

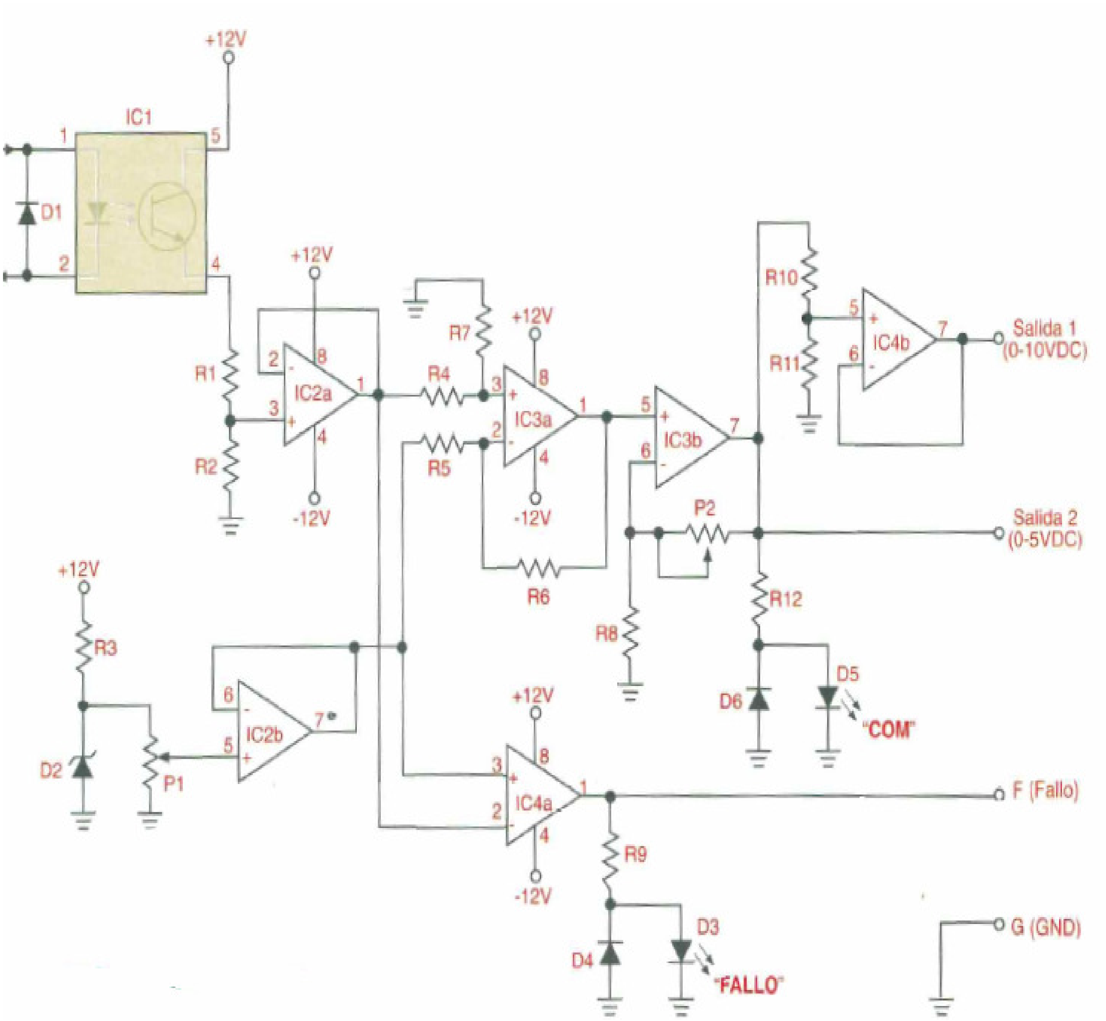


FIGURA 5

3.5 Funcionamiento

En el primer bloque está un optoacoplador que se encarga de aislar galvánicamente la corriente que ingresa y convertirla en un voltaje proporcional. Para esto el opto acoplador esta configurado como un seguidor de emisor y para que trabaje en zona lineal.

Los seguidores de tensión que se utilizan sirven para acoplar los voltajes entregados por el opto acoplador y por el trimmer P1 y su objetivo de este trimmer es para calibrar la salida a 0[V]. La siguiente etapa es un amplificador diferencial de ganancia unitaria y su objetivo es restar la tensión equivalente a 0 V del voltaje de entrada proporcional a la corriente medida.

Este es un voltaje que será amplificado de 0 – 10V por el IC3 de la figura porque está configurado como amplificador no inversor de ganancia variable de entre 1 y 10. Y esta salida del amplificador es la primera que tenemos de 0 – 10V y la segunda salida de 0-5V la obtenemos simplemente por un divisor de tensión que está colocado en otro amplificador que está configurado como un buffer.

El circuito también tiene un diodo Led detector de fallo que se activa en el momento que el bucle reciba una corriente menor a 4mA.

3.6 Calibración del Circuito

Lo primero que debemos hacer es colocar los trimmer al 50% de su valor. Entonces a la entrada del bucle es decir en el optoacoplador debemos colocar 4 mA y en cualquiera de las dos salidas la de 0 – 5V o 0 – 10V debemos medir con el multímetro 0V y todo esto se logra ajustando el trimmer P1.

El otro punto de calibración se hacen circular 20mA en la entrada del optoacoplador para obtener en las salidas un voltaje igual a 10V o 5V y esto se logra gracias al trimmer P2.

4. DSPIC 30F4011

4.1 INTRODUCCIÓN

Microchip, fabricante de los DSPIC, los ha bautizado con el nombre de DSC (Digital Signal Controller), que puede ser traducido como Controlador Digital de Señal.

Los Procesadores Digitales de Señales o DSP surgieron cuando la tecnología permitió su fabricación y las características de las aplicaciones lo necesitaron. La telefonía móvil, la electromedicina, la robótica, las comunicaciones, las reproducción y el procesamiento del sonido y la imagen, Internet, el control de motores, la instrumentación, son algunas de las áreas típicas de los DSP.

Cuando los microprocesadores y microcontroladores de 8, 16, 32 bits no fueron capaces de resolver eficientemente las tareas que el procesamiento digital de señales exigía se reforzaron sus arquitecturas, se amplió el repertorio de instrucciones y se les arropó con numerosos periféricos complementarios para dar lugar a los DSP.

4.2 ¿Qué es un dsPIC?

Los dsPIC nacen después de que los DSP hayan sido desarrollados durante años. En su diseño han participado expertos y especialista de muchas áreas. Los dsPIC se han aprovechado de la experiencia acumulada por otros fabricantes.

Los DSC poseen todos los recursos de los mejor microcontroladores embebidos de 16 bits conjuntamente con las principales características de los DSP, permitiendo su aplicación en el extraordinario campo del procesamiento de las señales analógicas y digitalizadas.

Un DSC ofrece todo lo que se puede esperar de un microcontrolador: velocidad, potencia, manejo flexible de interrupciones, un amplio campo de funciones periféricas analógicas y digitales, opciones de reloj, protecciones, perro guardián, seguridad del código, simulaciones en tiempo real, etc.

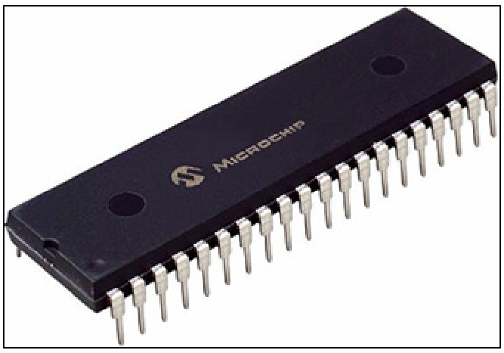


FIGURA 6

4.3 PRINCIPALES CARACTERÍSTICAS DE LOS DSPIC

Entre las aportaciones típicas de los DSP que se han implementado en la arquitectura básica de los DSPIC destacan:

- Multiplicación MAC 16 x 16 en un ciclo.
- 2 acumuladores de 40 bits.
- Registro de desplazamiento de 40 bits para el escalado.
- Acceso simultáneo de dos operandos.
- Bucles con estructura DO y REPEAT.
- Bloque de registros de trabajo.
- Juego flexible de interrupciones.
- Perro guardián.
- Emulación en tiempo real.
- Optimización para programación en lenguaje C.

4.4 RANGO DE FUNCIONAMIENTO

- DC – 30MIPS (30MIPS @ 4,5 – 5,5V, -40º a 85ºC).
- Voltaje de alimentación de 2,5 a 5,5V.
- Temperatura: interna de -40º a 85ºC y externa de -40º a 125ºC.

4.5 Estructura del DSPIC 30F4011

El núcleo del DSPIC es de 24 bits y el CP (Counter Program) es de 23 bits pero siempre tiene el bit menos significativo en clear y el bit mas significativo es ignorado durante la ejecución normal del programa.

El arreglo de los registros de trabajo es de 16x16 y cada uno puede trabajar como registro de datos, direccionamiento o de desplazamiento, el último registro W15 es el registro de pila o almacenamiento que guarda la dirección cuando el programa ejecuta llamada a subrutina, procedimiento o función.

El espacio de la memoria de datos es de 64 bytes o 32 word y esta dividida en dos bloques: memoria de datos X y memoria de datos Y. Cada bloque tiene su propia unidad generadora de dirección (AGU) es decir tiene un AGU X y un AGU Y. Además el núcleo puede ejecutar tres instrucciones en un solo ciclo de instrucción.

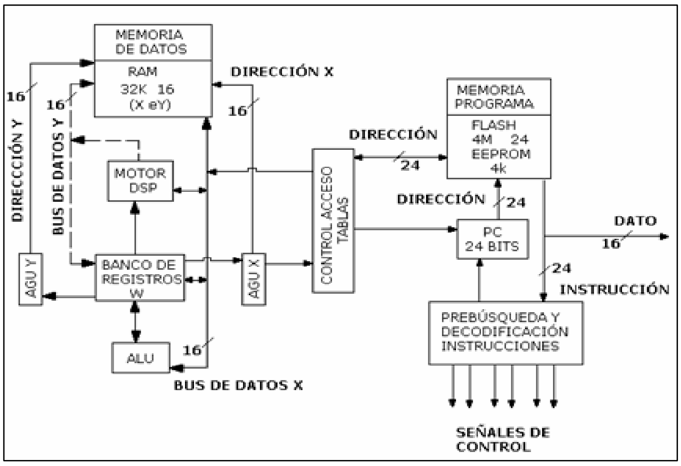


FIGURA 7

4.6 El modelo del Programador

El Registro Estado SR de DSPIC: el primer bit contiene la bandera de operación de la ALU y la parte más alta del byte del Registro Estado contiene estado de los bits acumuladores, restadores del DSP.

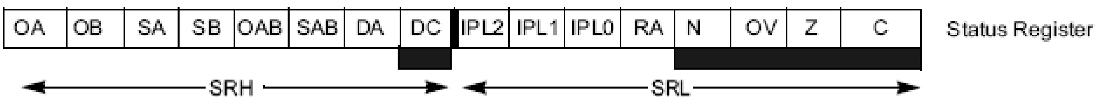


FIGURA 8

El DSPIC 30F4011 tiene: 16 registros de trabajo(W0…W15) de 16x16bit, dos registros acumuladores de 2x40 bit AccA y AccB , rsgistro STATUS(SR), registro de tabla de pagina, registros: Do Repeat, Do start Do End, Do Count y el Counter Program (PC).

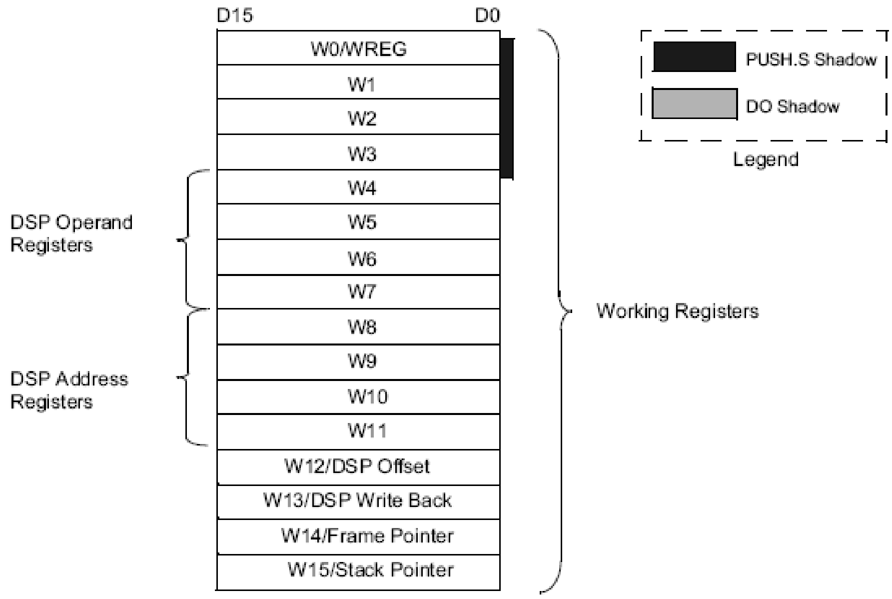


FIGURA 9

Organización de Memoria del DSPIC:

Esta memoria que es de 24 bits la direccionamos con el PC. Se puede leer/escribir en la memoria de programa a través de los siguientes registros:

_ TBLRDL: Leer parte baja

_ TBLRDH: Leer parte alta

_ TBLWTL: Escribir parte baja

_ TBLTH: Escribir parte alta.

Los Registros de Función Especial SFR son útiles a partir de la dirección 0x800.

Osciladores y Gestión de Energía de los DSPIC 30F:

Esta familia dispone de dos recursos fundamentales para el funcionamiento de los DSPIC:

1) El sistema de osciladores que generan las frecuencias necesarias para el clock y para los temporizadores que controlan el retardo para la estabilización de energía y de la frecuencia. Además puede seleccionar las señales de reloj y de los osciladores, temporizadores para la estabilización de la señal de reloj de la tensión y para la estabilización de la frecuencia.

2) El gestor de energía orientado a la administración del consumo de energía y a la tensión de alimentación. Además hace la detección de nivel bajo de energía (LVD), modos de bajo consumo SLEEP o IDLE, reset programable para la bajada de tensión en la alimentación (BOR), selección de la señal de suministro del clock.

4.7 El sistema Oscilador

Es el encargado de dar la señal de reloj a los elementos principales y auxiliares. Para realizar todas estas funciones el DSPIC tiene: tres osciladores primarios, uno secundario, dos osciladores internos uno externo y de un circuito PLL que sirve para elevar la frecuencia del oscilador interno

a) **Osciladores Primarios**: Son tres el XTL, XT y el HS. El XTL es un oscilador de cuarzo para un rango de frecuencias de 200KHz a 4MHz. El XT también utiliza un cristal de cuarzo o un resonador cerámico pero que va de 4MHz a 10MHz. Y el HS que es de cristal de cuarzo pero que desde los 10MHz hasta los 25MHz. Todos estos osciladores primarios emplean las patitas OSC1 y OSC2.

b) **Osciladores Secundarios**: Es un oscilador de baja potencia LP y es de un resonador de cristal o cerámico de 32KHz y se los conecta en los pines SOSC1 y SOSC0 que son los pines 16 y 17 respectivamente.

c) **Osciladores Internos**: Hay dos LPRC y un rápido RC (FRC) que trabaja a 8MHz para aplicaciones de alta velocidad el primero es de baja potencia y es 512KHz. Aquí en estos osciladores internos la frecuencia de oscilación depende de la temperatura y el voltaje del dispositivo.

d) **Oscilador Externo**: Es el único que tiene el DSPIC es gracias a una red RC conectada al pin 13 OSC1 y trabaja hasta una frecuencia máxima de 4MHz y este valor depende del valor del resistor, capacitor, voltaje y la temperatura del DSPIC.

En la siguiente figura se resume los sistemas osciladores que tiene el DSPIC según lo explicado anteriormente:

Primario, Secundario, Externo, Interno.

Oscillator Mode	Description
XTL	200 kHz-4 MHz crystal on OSC1:OSC2
XT	4 MHz-10 MHz crystal on OSC1:OSC2
XT w/ PLL 4x	4 MHz-10 MHz crystal on OSC1:OSC2, 4x PLL enabled
XT w/ PLL 8x	4 MHz-10 MHz crystal on OSC1:OSC2, 8x PLL enabled
XT w/ PLL 16x	4 MHz-10 MHz crystal on OSC1:OSC2, 16x PLL enabled(1)
LP	32 kHz crystal on SOSCO:SOSCI(2)
HS	10 MHz-25 MHz crystal
EC	External clock input (0-40 MHz)
ECIO	External clock input (0-40 MHz), OSC2 pin is I/O
EC w/ PLL 4x	External clock input (0-40 MHz), OSC2 pin is I/O, 4x PLL enabled(1)
EC w/ PLL 8x	External clock input (0-40 MHz), OSC2 pin is I/O, 8x PLL enabled(1)
EC w/ PLL 16x	External clock input (0-40 MHz), OSC2 pin is I/O, 16x PLL enabled(1)
ERC	External RC oscillator, OSC2 pin is Fosc/4 output(3)
ERCIO	External RC oscillator, OSC2 pin is I/O(3)
FRC	8 MHz internal RC oscillator
FRC w/ PLL 4x	8 MHz Internal RC oscillator, 4x PLL enabled
FRC w/ PLL 8x	8 MHz Internal RC oscillator, 8x PLL enabled
FRC w/ PLL 16x	7.5 MHz Internal RC oscillator, 16x PLL enabled
LPRC	512 kHz internal RC oscillator

FIGURA 10

4.8 Registros de Control de las Puertas de E/S

Todos los puertos del DSPIC tienen asociados directamente tres registros de control: TRIS, PORT y LAT.

Registros TRIS: Con este registro de control configuramos los pines de los puertos si son entradas o salidas. Por defecto cada vez que hay un reset en el DSPIC este configura al puerto como entrada.

Registros PORT: En este registro se encuentra el valor actual de entrada o salida del DSPIC.

Registro LAT: Con los registros LAT escribimos o leemos desde el registro cerrojo del DSPIC y no desde los pines físicos del mismo.

El DSPIC es de 40 pines y tiene multiplexado varios de sus pines:
Pines de Alimentación: VDD, VSS, AVDD, AVSS.

Señales Digitales de Entrada o Salida: Que están distribuidas en los o puertos B, C, D, E, F.

Señales Analógicas de Entrada: Que están distribuidas en el puerto B desde la AN0 hasta la AN8.

Señales del tipo de Oscilador: OSC1, OSC0, SOSC1, SOSC0.

Señales para comunicación tipo: CAN, UART, SPI, RS485.

4.9. Conversor Análogo Digital de 10/12 Bits

Este DSPIC nos permite convertir una señal análoga en digital con una alta velocidad y resolución de 10 o 12 bits. La conversión se realiza en tres etapas: muestreo de la señal, cuantificación de la señal que consiste en llevar el valor analógico a un valor de 2^n combinaciones y la codificación de la señal que consiste en asignarle los cero y unos a la señal convertida. El convertidor del DSPIC tiene las siguientes características:

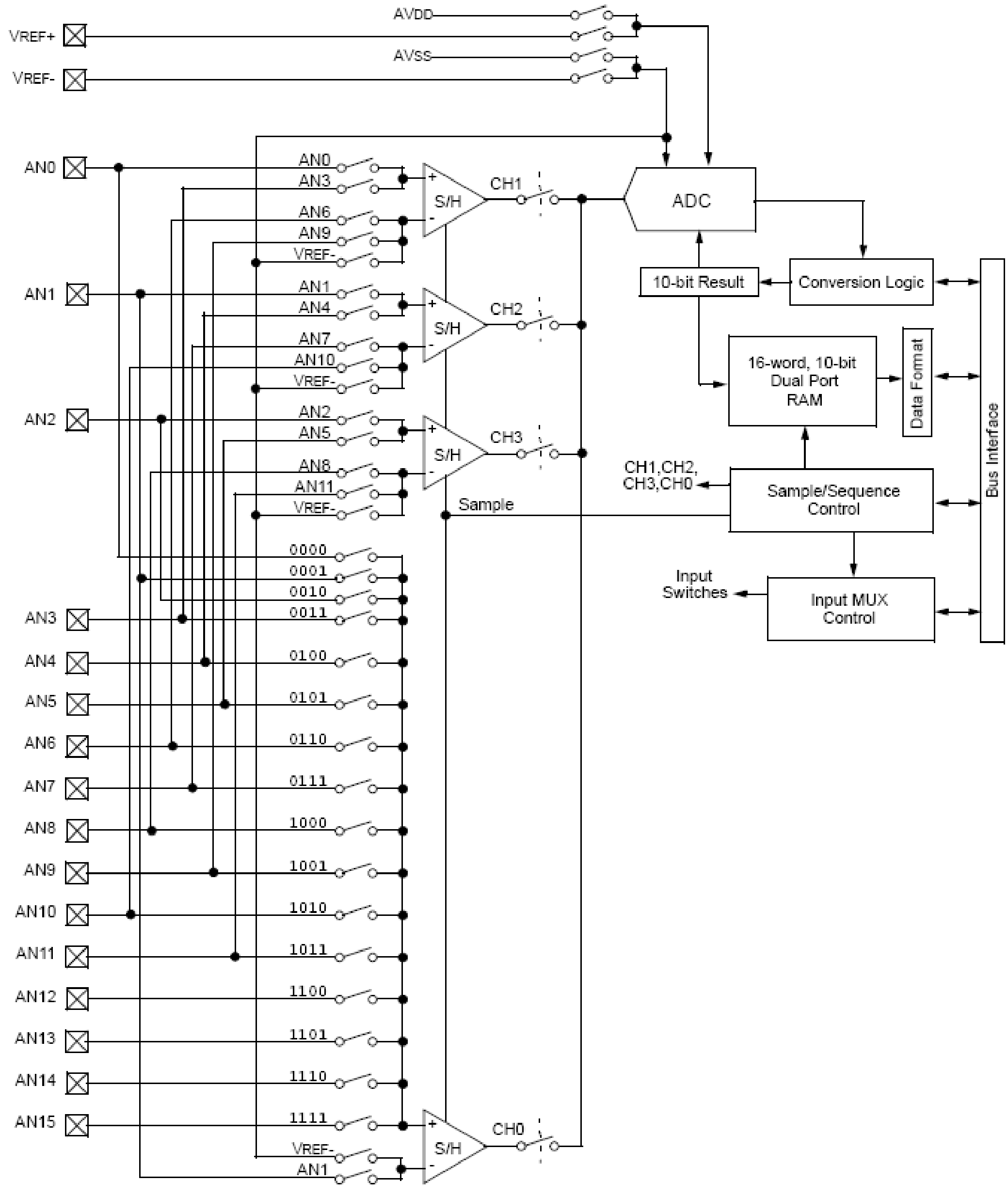
a) tiempo de muestreo de 154ns
b) conversión de aproximaciones sucesivas
c) velocidad de conversión de hasta 500Ksps
d) Modo automático de exploración de canal
e) La tensión analógica de referencia puede ser seleccionada por software

Para la configuración del módulo analógico debemos configurar a seis registros de control y de estado:

ADCON1, ADCON2, ADCON3, ADCHS, ADPCFG, ADCSSL.

Los tres primeros registros son los encargados de controlar al ADC, el registro encargado de seleccionar los pines que serán conectados a los cuatro amplificadores de retención y muestreo es el ADCHS ya que el módulo ADC tiene cuatro canales y por último tenemos el registro ADCSSL que es el encargado de seleccionar a los pines que serán muestreados, explorados por el ADC.

En la siguiente figura podemos ver el diagrama de bloques del módulo ADC que está formado por los pines de entrada analógica, cuatro amplificadores seleccionadores del canal, los mux para seleccionar el canal y el control para secuencia y muestro.



Note: VREF+, VREF- inputs may be shared with other analog inputs. See device data sheet for details.

FIGURA 11

El **ADCON1** sirve para controlar el funcionamiento del conversor, el funcionamiento en modo IDLE (desocupado), formato de salida del conversor, para indicar el inicio y secuencia de la conversión. En la siguiente figura veremos los bits encargados para realizar el trabajo que hemos dicho:

Register 17-1: ADCON1: A/D Control Register 1

Upper Byte:

R/W-0	U-0	R/W-0	U-0	U-0	U-0	R/W-0	R/W-0
ADON	—	ADSIDL	—	—	—	FORM<1:0>	
bit 15							bit 8

Lower Byte:

R/W-0	R/W-0	R/W-0	U-0	R/W-0	R/W-0	R/W-0 HC, HS	R/C-0 HC, HS
SSRC<2:0>			—	SIMSAM	ASAM	SAMP	DONE
bit 7							bit 0

FIGURA 12

ADCON2: Sirve para seleccionar el voltaje de referencia positivo o negativo del ADC, además nos indica que canales están siendo seleccionados en la conversión, para configurar el buffer de salida de conversión.

Register 17-2: ADCON2: A/D Control Register 2

Upper Byte:

R/W-0	R/W-0	R/W-0	U-0	U-0	R/W-0	R/W-0	R/W-0
VCFG<2:0>			reserved	—	CSCNA	CHPS<1:0>	
bit 15							bit 8

Lower Byte:

R-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
BUFS	—	SMPI<3:0>				BUFM	ALTS
bit 7							bit 0

FIGURA 13

ADCON3: Para seleccionar el tiempo de conversión TAD, el tiempo de muestreo, para seleccionar la fuente del reloj de conversión y las treinta dos posibilidades del tiempo del reloj de conversión.

Register 17-3: ADCON3: A/D Control Register 3

Upper Byte:							
U-0	U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
—	—	—	SAMC<4:0>				
bit 15							bit 8

Lower Byte:							
R/W-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
ADRC	—	ADCS<5:0>					
bit 7							bit 0

FIGURA 14

ADCHS: Es el encargado de seleccionar las entradas que van ha ser conectados a los amplificadores de muestreo y retención, además sirve para seleccionar a que entrada positiva o negativa de los amplificadores conectamos las entradas y esto lo hacemos gracias a un MUX A y MUX B y los bits 15,..,8 configuran al MUXB y los bits 7,…,0 configuran al MUX A.

Register 17-4: ADCHS: A/D Input Select Register

Upper Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH123NB<1:0>		CH123SB	CH0NB	CH0SB<3:0>			
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CH123NA<1:0>		CH123SA	CH0NA	CH0SA<3:0>			
bit 7							bit 0

FIGURA 15

ADPCFG: Este registro me indica cuales son los pines del DSPIC que trabajaran como I/O digital si colocamos un `1´ y para ser entradas analógicas debemos colocar un `0´ en este registro.

Register 17-5: ADPCFG: A/D Port Configuration Register

Upper Byte:

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG15	PCFG14	PCFG13	PCFG12	PCFG11	PCFG10	PCFG9	PCFG8
bit 15							bit 8

Lower Byte:

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
PCFG7	PCFG6	PCFG5	PCFG4	PCFG3	PCFG2	PCFG1	PCFG0
bit 7							bit 0

FIGURA 16

ADCSSL: Con este registro podemos seleccionar el orden en que estas entradas serán seleccionadas, al igual que el registro anterior utiliza todos sus bits para realizar esta función.

Register 17-6: ADCSSL: A/D Input Scan Select Register

Upper Byte:

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSSL15	CSSL14	CSSL13	CSSL12	CSSL11	CSSL10	CSSL9	CSSL8
bit 15							bit 8

Lower Byte:

R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0
CSSL7	CSSL6	CSSL5	CSSL4	CSSL3	CSSL2	CSSL1	CSSL0
bit 7							bit 0

FIGURA 17

4.10 MODULO UART

Este módulo permite la comunicación Universal Serial Asíncrona full duplex o serial asíncrona con ordenadores, con interfaces RS 232 y RS 485, entre las principales características son:

a) La transmisión de datos es de 8 o de 9 bits.
b) Trabajar con paridad impar, par o sin paridad.
c) Uno o dos bits de STOP
d) Tiene un generador de baudios con preescalador de 16 bits que se encarga de dar la frecuencia de trabajo al modulo.
e) Los Buffers de Transmisión y Recepción son de máximo cuatro caracteres.

Para el correcto funcionamiento del modulo UART debemos configurar algunos registros para el bloque Generador, Transmisor y Receptor.

USTA: Este registro produce una interrupción cuando enviamos un bit para el registro de desplazamiento TSR, podemos habilitar o deshabilitar la trasmisión, bits de error para saber si hay error en la transmisión o recepción.

Register 19-2: UxSTA: UARTx Status and Control Register

Upper Byte:							
R/W-0	U-0	U-0	U-0	R/W-0	R/W-0	R-0	R-1
UTXISEL	—	—	—	UTXBRK	UTXEN	UTXBF	TRMT
bit 15							bit 8

Lower Byte:							
R/W-0	R/W-0	R/W-0	R-1	R-0	R-0	R/C-0	R-0
URXISEL<1:0>		ADDEN	RIDLE	PERR	FERR	OERR	URXDA
bit 7							bit 0

FIGURA 18

URREG: Es el registro de recepción donde se encuentra el carácter recibido y como sabemos podemos recibir 8 o 9 bits.

Register 19-3: UxRXREG: UARTx Receive Register

Upper Byte:							
U-0	U-0	U-0	U-0	U-0	U-0	U-0	R-0
—	—	—	—	—	—	—	URX8
bit 15							bit 8

Lower Byte:							
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0
URX<7:0>							
bit 7							bit 0

FIGURA 19

UTREG: Es el registro de transmisión donde se aloja el carácter a enviar y como sabemos que hay dos modos el de 8 o9 bits.

Register 19-4: UxTXREG: UARTx Transmit Register (Write Only)

Upper Byte:							
U-0	U-0	U-0	U-0	U-0	U-0	U-0	W-x
—	—	—	—	—	—	—	UTX8
bit 15							bit 8

Lower Byte:							
W-x	W-x	W-x	W-x	W-x	W-x	W-x	W-x
UTX<7:0>							
bit 7							bit 0

FIGURA 20

5. Uso del Software de programación MiKrobasic for DSPIC

En el desarrollo del programa se debe cumplir una estricta organización de programa. Básicamente todo programa está dividido en dos secciones las Declaraciones y el Cuerpo de Programa, el modelo siguiente es una forma correcta de escribir el código de un programa:

'************************

'Declaraciones:

'************************

' Declaración de Símbolos:

symbol ...

' Declaración de Constantes:

const ...

' Declaración de Estructuras:

structure ...

' Declaración de Variables Globales:

dim ...

' Declaración de Procedimientos:

sub procedure *procedure_name*(...)

< Declaración de Variables Locales >

```
  ...
end sub

' Declaración de Funciones:
sub function function_name(...) as return_type
  <local declarations>
  ...
end sub

'************************
'* Cuerpo del Programa:
'************************
main:
  ' write your code here
end.
```

Lo primero que debemos hacer cuando empezamos ha realizar el codigo del DSPIC es configurarlo, para lo cual utilizamos sus registros y ademas debemos utilizar las funciones que nos brinda el Mikrobsic como por ejemplo: ADC Library, LCd Custom Library, PWM Library, y KeyPad Library y UART Library.

5.1 ADC Library

Este módulo Analog to Digital Converter es el encargado de convertir las señales análogas en señales discretas digitales y en el caso del DSPIC podemos escoger su resolución que puede ser de 10bits o 12bits. Por defecto este canal tiene un reloj de conversión de 32TCY es decir que depende del periodo de reloj del ciclo de instrucción (TCY) y de un tiempo de auto muestreo de 31TAD.

5.2 LCD Custom Library

Nos brinda algunas funciones para el manejo de la LCD que principalmente se dividen en dos: para la configuración de la LCD y para la presentación de datos que desde el DSPIC la podemos enviar en forma de string o de char, que en nuestro caso la más utilizada fue la segunda porque solamente debemos enviar el dato en forma de byte o Word; mientras que en la primera debíamos transformar la variable a string con lo que el código se hacía más extenso. Además el Mikrobasic nos brinda muchos comandos para mejorar la presentación de la LCD:

LCD_FIRST_ROW	Mueve el cursor a la primera línea
LCD_SECOND_ROW	Mueve el cursor a la segunda línea
LCD_THIRD_ROW	Mueve el cursor a la tercera línea
LCD_FOURTH_ROW	Mueve el cursor a la cuarta línea
LCD_CLEAR	Limpia la LCD
LCD_RETURN_HOME	Retorna el cursor a la primera fila y columna
LCD_CURSOR_OFF	Apaga el cursor de la LCD
LCD_UNDERLINE_ON	Muestra el cursor en firma de subguion
LCD_BLINK_CURSOR_ON	Hace parpadear el cursor
LCD_MOVE_CURSOR_LEFT	Mueve el cursor a la izquierda
LCD_MOVE_CURSOR_RIGHT	Mueve el cursor a la derecha
LCD_TURN_ON	Enciende el display de la LCD
LCD_TURN_OFF	Apaga el display de la LCD
LCD_SHIFT_LEFT	Desplaza la LCD a la izquierda
LCD_SHIFT_RIGHT	Desplaza la LCD a la derecha

5.3 PWM

Esta librería nos facilita el modulo CCP del DSPIC. La función que nos permite configurar el canal, la frecuencia, el prescalador del timer y el timer que se va a usar el 1 o 2 lo realiza la función PWM_INIT. La función que nos permite cambiar el rango que esta en alto con respecto al período es la función PWM_SET_DUTY.

5.4 KEYPAD

Con esta librería podemos trabajar teclados de 4x1, 4x2, 4x3 y 4x4 donde el primer número es la fila y el segundo es la columna. La función Keypad_Init nos sirve para definir en que puerto debe estar conectado el teclado y se debe escoger los bits menos significativos del puerto.

En el momento en que presionamos la tecla tenemos dos posibilidades: usar keypad_key_press() o keypad_key_click() pero era preferible la segunda porque esta enviaba el dato cuando habíamos soltado la tecla entonces nos permitía que el DSPIC lea valores deseados.

Se implementó una subrutina de teclado llamada Kypad() que hacia un barrido de cada fila colocando un cero y preguntando por cada columna si hemos presionado una tecla. Pero para mostrar la tecla presionada en la LCD tuvimos que enviar el Código ASCII de la tecla presionada.

5.5 UART LIBRARY

Nos permite realizar comunicación serial universal asíncrona. Si queremos que el DSPIC reciba información entonces debemos utilizar la función UART_READY_CHAR() UART_READY_TEXT() pero siempre antes de utilizar estas funciones debemos preguntar si el dato ha llegado al DSPIC en el buffer de recepción entonces esta listo para ser leído y la hacemos mediante la función UART_DATA_READY.

Igualmente el DSPIC puede enviar datos hacia un computador mediante las funciones UART_WRITE_CHAR() que le debemos ingresar la variable en forma de byte o word o la función UART_WRITE_STRING() pero la variable debe estar en string por lo que demos utilizar mas líneas de programa porque siempre trabajamos con variables tipo byte o word entonces debemos convertirlas.

En el DSPIC los pines que me permiten realizar la comunicación son el 25 que es la trasmisión y el pin de recepción que es el 26. Como ya es conocido el UART realiza comunicación serial y por lo tanto debemos utilizar un circuito conversor de lógica TTL a CMOS y esto la hace el circuito integrado MAX_232, que hablaremos sobre el mas adelante.

6.COMUNICACIÓN SERIE

Para la comunicación entre dispositivos podemos transmitir los datos de forma serial como RS232, USB y de forma paralela como GPIB o VIX.

Como ya sabemos la comunicación serie es menos costosa, puede transmitir a mayor distancia y necesita de poco hardware. La comunicación serie es en ráfagas y asíncrona por lo que suelen delimitarse con un bit de start y un bit de stop.

El RS232 se diseño y desde entonces ha tenido varias revisiones, comenzó a ser usado para la comunicación para un modem y una impresora de teletipo. Desde entonces se uso en ordenadores, impresoras, instrumentos de test y en casi todos los dispositivos electrónicos que necesitaban comunicarse.

6.1 RS232

Esta definido como la interfase entre un equipo terminal de datos (DTE) y un equipo de comunicación de datos (CDE) utilizando un intercambio binario en modo serie. Se acostumbra decir al ordenador que es el DTE y al modem como un CDE.

6.2 Circuito integrado para conversión de niveles

El **MAX232** es un circuito integrado que convierte los niveles de las líneas de un puerto serie RS232 a niveles TTL y viceversa. Lo interesante es que sólo necesita una alimentación de 5V, ya que genera internamente algunas tensiones que son necesarias para el estándar RS232. Otros integrados que manejan las líneas RS232 requieren dos voltajes, +12V y -12V.

El MAX232 soluciona la conexión necesaria para lograr comunicación entre el puerto serie de una PC y cualquier otro circuito con funcionamiento en base a señales de nivel TTL/CMOS.

El circuito integrado posee dos conversores de nivel TTL a RS232 y otros dos que, a la inversa, convierten de RS232 a TTL.

Estos conversores son suficientes para manejar las cuatro señales más utilizadas del puerto serie del PC, que son TX, RX, RTS y CTS.

TX es la señal de transmisión de datos, RX es la de recepción, y RTS y CTS se utilizan para establecer el protocolo para el envío y recepción de los datos.

6.3 Conectores

El RS 232 utiliza varios conectores siendo los más usados el de 25 pines, 68 pines, modular Jack de 10pines, RJ 45 de 8pines pero el conector más habitual es el de DB9 que tiene 5 líneas de recepción y 3 de transmisión. Para recibir o enviar la ráfaga de bits se utiliza un circuito llamado UART (Receptor Transmisor Universal Asíncrono) la tarea principal del UART es convertir los bits serie en paralelo. Siempre debemos evitar que los buffers de transmisión y recepción se saturen y LabView para evitar este problema utiliza un Hardware Handskaking, en la siguiente figura vemos el aspecto físico del DB9 y sus pines

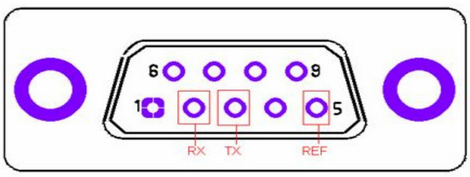


FIGURA 21

1	CD: Detector de transmisión
2	RXD: Recibir datos
3	TXD: Transmitir datos
4	DTR: Terminal de datos lista
5	GND: Señal de tierra
6	DSR: Ajuste de datos listo
7	RTS: Permiso para transmitir
8	CTS: Listo para enviar
9	RI: Indicador de llamada
	Protección

Figura 22

7. Comunicación Serial con LabView:

LabView nos brinda el driver NI VISA para la comunicación, en el cual podemos encontrar: VISA Configurate Serial Port, VISA Write, VISA Read y VISA Close. VISA es un API o una librería desarrollada por varios fabricantes que proporcionan un software estándar para la lectura y escritura en instrumentación.

VISA Configurate Serial Port: Sirve para configurar el puerto serie con los siguientes parámetros: velocidad, protocolo, paridad, bits de datos, habilitar terminación de char. En el visa resourece name me indica cuales son los puertos disponibles para la transmisión. Por lo general ya viene configurado con los valores por defecto.

VISA Write: Al igual que el anterior debemos indicarle desde que puerto el va escribir y lo hacemos con el Visa Resource Name y entonces enviara el dato que se encuentre en el buffer de escritura a través de write buffer.

VISA READ: Es lo contrario al anterior hay que indicarle el numero de bytes que debe leer desde el buffer, para darle un valor a este parámetro suele usarse la salida de la propiedad VISA Bytes at Serial Port.

VISA CLOSE: Cierra una sesión VISA. Para abrirlas se usa el VI Instrument I/O>VISA>VISA ADVANCED>VISA OPEN. Se puede cerrar automáticamente en el menú Tools>Options>Enviroment>Automatically Close <Visa.

VISA Bytes at Serial Port: Es una propiedad del driver VISA, que nos da el número de bytes que están en el buffer esperando para ser leídos. Aparte de esta propiedad hay otras que pueden cambiar la configuración del puerto. Algo muy importante es que para acceder a esta propiedad debemos ir a un control y que en este caso seria el VISA Resource Name en donde haremos clic derecho Create>Property> Property For Instr>Serial Settings.

7.1 Comunicación Serial entre DSPIC 30F4011 y LabView

En la comunicación entre el controlador y el visualizador tienen que estar en sincronismo entonces el programa fue hecho de manera secuencial en el DSPIC y en LabView el programa sigue la secuencia Data Flow es decir que el programa se ejecuta de acuerdo a como estén presentes los datos en los nodos.

Lo primero que hicimos en LabView fue configurar el puerto serial con los valores por defecto solo la velocidad fue cambiada a 57600 baudios.
Los siguientes 5 funciones se convierten en la parte esencial entre la comunicación del DSPIC y LabView :

String to Byte Array:

Convierte una cadena de caracteres es decir un string en byte pero en forma de arreglo ya que nosotros en el DSPIC enviamos los datos uno detrás del otro.

Search ID Array:

Esta función me retorna el índice del dato que yo estoy buscando en nuestro caso yo deseo saber el índice del número 120 que lo escogí porque es valor que en nuestro proceso no se va dar.

Split ID Array:

Con esta función yo estoy partiendo los datos que yo envió desde el DSPIC apartir del número 120 y como vemos esta función me devuelve dos subarreglos.

Build Array:

Aquí vuelvo a unir el arreglo que había divido apartir del 120 pero con esto número como primer elemento.

Index Array:

Con esta función Index Array obtenemos cada variable que enviamos desde el DSPIC individualmente y así podemos monitorear el estado de todos los sensores que tiene nuestro proceso.

Feedback Node:

Es utilizado en lazos for y while y transfiere el valor de una iteración a la siguiente.

VISA Close:

Y por último algo muy importante es no olvidarnos de cerrar la sesión VISA

El resto del programa en LabView es para encender o apagar los sensores de nivel alto y bajo del boiler y del mixer. Y los contadores son para simular el llenado y vaciado de los 2 tanques.

8. APLICACIÓN: MACERACIÓN DE LA MALTA

Los ingredientes tamizados (malta y el grit) se introducen en un tanque. La proporción entre la malta y el grit dependerá de la receta del maestro cervecero, pero generalmente suele ser aproximadamente de un 1/3 de malta. A la mezcla acuosa se la somete a cierta temperatura para favorecer el ataque sobre el almidón de las enzimas.

Se detiene la temperatura para activar los **enzimas** y luego pasa a un tanque mezclador. Principalmente, se trata de hacer pasar la mezcla por diversas etapas más o menos largas de temperatura, cada etapa siendo óptima para enzimas diferentes. De este proceso de maceración de la malta se obtiene, un líquido claro y azucarado que se denomina "mosto". El proceso completo dura unas horas.

CONCLUSIONES Y RECOMENDACIONES

Para simular y monitorear un proceso industrial no es necesario tener todos los elementos físicos como motores, sensores o electroválvulas ya que se lo puede lograr gracias a la plataforma de LabVIEW y teniendo un controlador en este caso el dsPIC.

Se llegó a desarrollar un sistema de gran utilidad para un ambiente industrial logrando integrar hardware y software de una manera muy práctica con elementos a los que se tuvo fácil acceso.

Se logró implementar el receptor de bucle aprovechando las características del microcontrolador mediante sus entradas analógicas y puerto de comunicación.

Los DSP son mucho mejores en cuanto a velocidad de procesamiento, ya que tiene incorporada una unidad MAC de un ciclo de máquina, con lo que se puede hacer muy fácilmente filtros y controladores digitales súper rápidos.

Mediante el desarrollo del proyecto se pudo mejorar los conocimientos adquiridos sobre electrónica tanto análoga como digital y conocer nuevos dispositivos de comunicación.

Al realizar la comunicación con LabView tuvimos que aumentar la velocidad de transmisión del valor por defecto del DSPIC que es de 9600 baudios a 57600baudios.

Como LabView lo estamos utilizando solo un programa de monitoreo se recomienda que los niveles de los tanques sean enviados desde el DSPIC para evitar la programación en LabView y así retardar la comunicación.

Se recomienda colocar filtros después de una fuente de alimentación ya que pudiera presentar ruidos en nuestras señales de adquisición y no lograríamos apreciar el verdadero comportamiento de la señal deseada.

Para realizar la comunicación tuvimos que eliminar los retardos DELAY que hacíamos en el DSPIC porque esto provocaba que no hubiera sincronización con LabView y por lo tanto no había comunicación por lo tanto nunca dejamos de enviar datos hacia LabView y que siempre estamos actualizando las variables.

ANEXO A

PROGRAMACION EN MIKROBASIC

```
program bucle2_20ma

'******* PortD:  Filas  ********
symbol fa=portd.0        'Pin_23
symbol fb=portd.1        'Pin_18
symbol fc=portd.2        'Pin_22
symbol fd=portd.3        'Pin_19
'******* PortC:  Columnas  ********
symbol c1=portc.13       'Pin_15
symbol c2=portc.14       'Pin_16
symbol c3=portc.15       'Pin_14


'******* Variables ********
dim cnt0 as byte
dim  n_current,n_current2,n_before,num as word 'Para Ingresar La Temp x
Teclado
dim txt1, txt2 as string[5]
dim  pot3,pot8 as word        'Para los LM35
dim f_pot3,f_pot8 as float    'f_lm35

'******* + Variables ********
dim  s1, s2,s3,s4,b1,b2,ev1,ev2, boiler, mix as byte
dim flag1 as boolean
dim cnt1,cnt2 as longword


'*******  Sub_Procedures  ********
sub procedure clear_lcd()
   lcd_cmd(lcd_clear)
end sub

sub procedure dly50()
   delay_ms(50)
end sub

sub procedure dly2000()
   delay_ms(1500)
end sub
```

```
'Function_KeyPad: Retorna el ASCII para la LCD
sub function kypad (dim ky as byte) as byte

        fa=0
        if flag1=0 then dly50() end if
        if c1=0  then  gosub ar ky=49   end if   '1
        if c2=0  then  gosub ar ky=50   end if   '2
        if c3=0  then  gosub ar ky=51   end if   '3
        fa=1

        fb=0
        if flag1=0 then dly50() end if
        if c1=0  then   gosub ar ky=52   end if   '4
        if c2=0  then   gosub ar ky=53   end if   '5
        if c3=0  then   gosub ar ky=54   end if   '6
        fb=1

        fc=0
        if flag1=0 then dly50() end if
        if c1=0  then   gosub ar ky=55   end if   '7
        if c2=0  then   gosub ar ky=56   end if   '8
        if c3=0  then   gosub ar ky=57   end if   '9
        fc=1

        fd=0
        if flag1=0 then dly50() end if
        if c1=0  then  gosub ar  ky=42  end if    '*
        if c2=0  then  gosub ar  ky=48  end if    '0
        if c3=0  then  gosub ar  ky=35  end if    '#
        fd=1

        result=ky
   ar:
        if c1=0 then    goto ar    end if
        if c2=0 then    goto ar    end if
        if c3=0 then    goto ar    end if
   return
end sub

'Function_Convert: Retorna el Verdadero Número Para la Temperatura
sub function convert1 (dim xx as byte) as byte
    select case xx
          case 48     result=0
```

```
        case 49     result=1
        case 50     result=2
        case 51     result=3
        case 52     result=4
        case 53     result=5
        case 54     result=6
        case 55     result=7
        case 56     result=8
        case 57     result=9
   end select
end sub

'Function_Convert: Retorna el Verdadero Número Para la Temperatura
sub function convert2 (dim xxx as byte) as byte
   select case xxx
        case 0    result=48
        case 1    result=49
        case 2    result=50
        case 3    result=51
        case 4    result=52
        case 5    result=53
        case 6    result=54
        case 7    result=55
        case 8    result=56
        case 9    result=57
   end select
end sub

'Subprocedure_Comunicacion: Realiza la trasmision del DSPIC
sub procedure comunicacion( dim pri as  word )
   dim primero, segundo as word

   primero = (pri div 10)             'Primer digito
   segundo = (pri mod 10 )             'Segundo digito
   uart_write_char(primero)  'Escribe en el usart el primer digito
   uart_write_char(segundo)  ' Escribe en el usart el segundo digito
   uart_write_char(mix)    'espacio_32_$20

end sub
'******* Programa_Principal  ********
main:

'*******   Configurando Registros  ********
'Configurando ADC:
```

```
  adpcfg  = $fef7      'RB3/RB8: Analog Inputs antes fff7

trisd=$00             'PORTD filas son salidas del kyboard
trisc=$f000            'PORTC columnas son entradas  del kyboard
trise=$00             'trise.0=0 trise.1=0  trise.2=0 Fan y Foco
'trisb.3=1  trisb.8=1   'los lm35
trisf.4=0  trisf.5=0   'electrovalvulas
trisf.0=0  trisf.1=0  trisf.6=0 ' Niveles Bajo y Alto
trise.8=0           'motor


'********   Configurando:   LCD_UART  ********
uart_init(57600)
lcd_init(portb,7,6,5,4, portb,0,2,1)
lcd_cmd(lcd_cursor_off)
clear_lcd()
dly50()

'********   Inicializando Variables  ********
cnt1=0 cnt2=0   porte=$00
s1=0  s2=1 s3=0 s4=1
ev1=0 ev2=0 boiler=0  mix=0 flag1=0

'*********************************************************
'  Presentando Mensajes del Proceso y Escoger Temperatura
'*********************************************************
   lcd_out(1,1," ***Proceso***") lcd_out(2,3,"Maceracion")
   porte.0=0  dly2000()     'On  1foco_ port2.0
   porte.8=0
s2:
   clear_lcd()
   lcd_out(1,2,"Temp de Mezcla:") ' Receta Del Maestro Cervecero
   lcd_out(2,5,"49<Tm<61")  dly2000()  clear_lcd()
   lcd_out(1,2,"Temp de Mezcla:")
   lcd_cmd(lcd_second_row)

'*********************************************************
' Inicializando Variables Para Ingresar La Temperaura
' por el Teclado: n_current/n_before/num
'*********************************************************


   n_before=0   n_current=0    num=0
'********    Ingresando Temperatura x Teclado:     ********
s3:
```

```
  do
    n_current=kypad(25)           'Retorna el ASCII para la LCD
    n_current2=n_current          'Respaldo del codigo ASCII
    n_current=convert1(n_current)  'Obtengo el verdadeero número
    if n_current2=25 then   goto  s3  end if
    if n_current2=35 then   break     end if
    if n_current2=42 then         'Tecla * es invalida
      clear_lcd()
      lcd_out(1,8,"*") lcd_out(2,2,"Tecla_InValida")
      dly2000() clear_lcd() lcd_out(1,2,"Temp de Mezcla:")
      lcd_cmd(lcd_second_row)  goto s2
    end if

    num=n_before*10+n_current   'num es un Acumulador
    n_before=num
    lcd_chr_cp(n_current2)
   loop until n_current2=35       'enter = #

'*********  Validacion de Temperatura  **********

  clear_lcd()
  if ((n_before<61) and (n_before>49))  then
     lcd_out(1,1,"Temp_Correcta:")
     wordtostr(num,txt1)
     lcd_out(2,3,txt1)
     lcd_chr(2,9,111)  lcd_chr(2,10,67)
     dly2000()
  else
     lcd_out(1,1,"Temp_Incorrecta:")
     wordtostr(num,txt1)
     lcd_out(2,3,txt1)
     lcd_chr(2,9,111)  lcd_chr(2,10,67)  'lcd_chr(2,5,num)
     dly2000()
     goto s2
  end if

  pot3=adc_read(3)
  f_pot3=pot3*0.146     '

fin:
'**************************************************************
'  Inicializando Comunicacion con LabView utilizando el UART
'**************************************************************
 uart_write_char(120)
```

```
uart_write_char(s1)  uart_write_char(s2)
uart_write_char(s3)  uart_write_char(s4)
uart_write_char(b1)  uart_write_char(ev1)
uart_write_char(b2)  uart_write_char(ev2)
uart_write_char(boiler)
comunicacion(f_pot3)

b1=1  b2=1            'Envio a LabView
uart_write_char(120)
uart_write_char(s1)  uart_write_char(s2)
uart_write_char(s3)  uart_write_char(s4)
uart_write_char(b1)  uart_write_char(ev1)
uart_write_char(b2)  uart_write_char(ev2)
uart_write_char(boiler)
comunicacion(f_pot3)

 flag1=1

do
  pot3=adc_read(3)
  f_pot3=pot3*0.146
  uart_write_char(120)
  uart_write_char(s1)  uart_write_char(s2)
  uart_write_char(s3)  uart_write_char(s4)
  uart_write_char(b1)  uart_write_char(ev1)
  uart_write_char(b2)  uart_write_char(ev2)
  uart_write_char(boiler)
  comunicacion(f_pot3)
loop until 51=kypad(25)    '#3= s2
  s2=0
  uart_write_char(120)
  uart_write_char(s1)  uart_write_char(s2)
  uart_write_char(s3)  uart_write_char(s4)
  uart_write_char(b1)  uart_write_char(ev1)
  uart_write_char(b2)  uart_write_char(ev2)
  uart_write_char(boiler)
  comunicacion(f_pot3)

do
  pot3=adc_read(3)
  f_pot3=pot3*0.146
  uart_write_char(120)
  uart_write_char(s1)  uart_write_char(s2)
  uart_write_char(s3)  uart_write_char(s4)
```

```
  uart_write_char(b1)  uart_write_char(ev1)
  uart_write_char(b2)  uart_write_char(ev2)
  uart_write_char(boiler)
  comunicacion(f_pot3)
loop until 50=kypad(25)    '#2= s1

  porte.0=1              'foco_ON
  s1=1 b1=0  b2=0  boiler=1
  uart_write_char(120)
  uart_write_char(s1)  uart_write_char(s2)
  uart_write_char(s3)  uart_write_char(s4)
  uart_write_char(b1)  uart_write_char(ev1)
  uart_write_char(b2)  uart_write_char(ev2)
  uart_write_char(boiler)
  comunicacion(f_pot3)

do
  pot3=adc_read(3)
  f_pot3=pot3*0.146
  uart_write_char(120)
  uart_write_char(s1)  uart_write_char(s2)
  uart_write_char(s3)  uart_write_char(s4)
  uart_write_char(b1)  uart_write_char(ev1)
  uart_write_char(b2)  uart_write_char(ev2)
  uart_write_char(boiler)
  comunicacion(f_pot3)
loop until f_pot3>=num
  boiler=0   porte.0=0   'foco_off
  uart_write_char(120)
  uart_write_char(s1)  uart_write_char(s2)
  uart_write_char(s3)  uart_write_char(s4)
  uart_write_char(b1)  uart_write_char(ev1)
  uart_write_char(b2)  uart_write_char(ev2)
  uart_write_char(boiler)
  comunicacion(f_pot3)

do
  pot3=adc_read(3)
  f_pot3=pot3*0.146
  uart_write_char(120)
  uart_write_char(s1)  uart_write_char(s2)
  uart_write_char(s3)  uart_write_char(s4)
  uart_write_char(b1)  uart_write_char(ev1)
  uart_write_char(b2)  uart_write_char(ev2)
```

```
  uart_write_char(boiler)
  comunicacion(f_pot3)
loop until f_pot3<=45
   ev1=1 s4=0  s1=0
do
  pot3=adc_read(3)
  f_pot3=pot3*0.146
  uart_write_char(120)
  uart_write_char(s1)  uart_write_char(s2)
  uart_write_char(s3)  uart_write_char(s4)
  uart_write_char(b1)  uart_write_char(ev1)
  uart_write_char(b2)  uart_write_char(ev2)
  uart_write_char(boiler)
  comunicacion(f_pot3)
  loop until 53=kypad(25)          '#5_sensor s3
  porte.8=1  cnt1=0
  s3=1   ev1=0   mix=1   s2=1

 do
  inc(cnt1)
  pot3=adc_read(3)
  f_pot3=pot3*0.146
  uart_write_char(120)
  uart_write_char(s1)  uart_write_char(s2)
  uart_write_char(s3)  uart_write_char(s4)
  uart_write_char(b1)  uart_write_char(ev1)
  uart_write_char(b2)  uart_write_char(ev2)
  uart_write_char(boiler)
  comunicacion(f_pot3)
 loop until cnt1>5000


porte.8=0
mix=0  ev2=1   s3=0
 do
  pot3=adc_read(3)
  f_pot3=pot3*0.146
  uart_write_char(120)
  uart_write_char(s1)  uart_write_char(s2)
  uart_write_char(s3)  uart_write_char(s4)
  uart_write_char(b1)  uart_write_char(ev1)
  uart_write_char(b2)  uart_write_char(ev2)
  uart_write_char(boiler)
  comunicacion(f_pot3)
```

```
  loop until 54=kypad(25)          '#6_sensor s4
   s4=1  ev2=0
   uart_write_char(120)
   uart_write_char(s1)  uart_write_char(s2)
   uart_write_char(s3)  uart_write_char(s4)
   uart_write_char(b1)  uart_write_char(ev1)
   uart_write_char(b2)  uart_write_char(ev2)
   uart_write_char(boiler)
   comunicacion(f_pot3)

   goto fin
end.
```

ANEXO B

PLACAS UTILIZADAS

ACONDICIONADOR DE SEÑAL

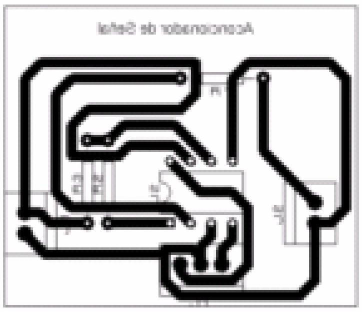


Figura b1

TRANSMISOR DE CORRIENTE

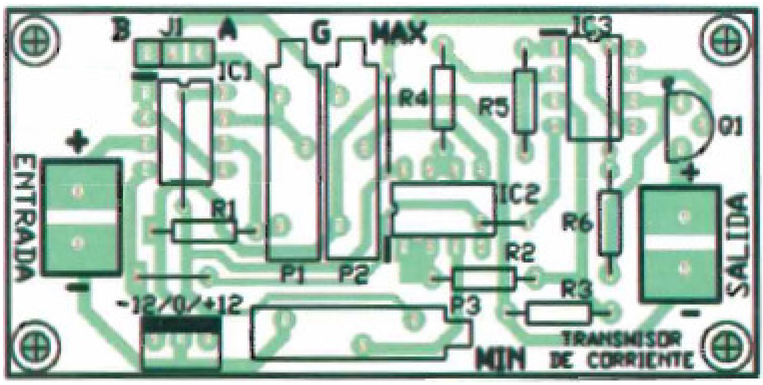


Figura b2

RECEPTOR DE BUCLE

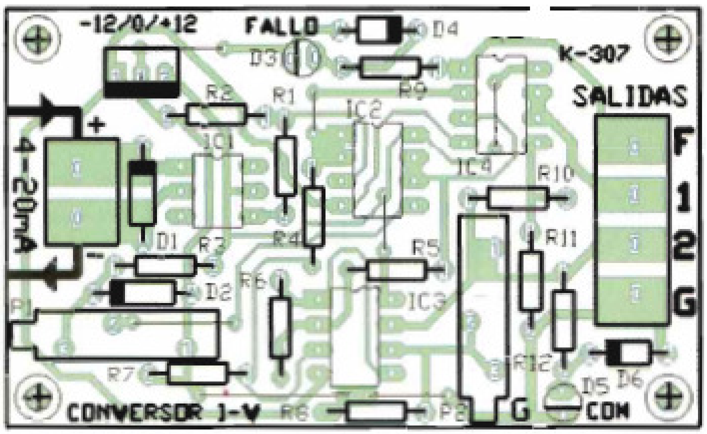


Figura b3

CIRCUITO PARA ENCENDER FOCO Y MOTOR

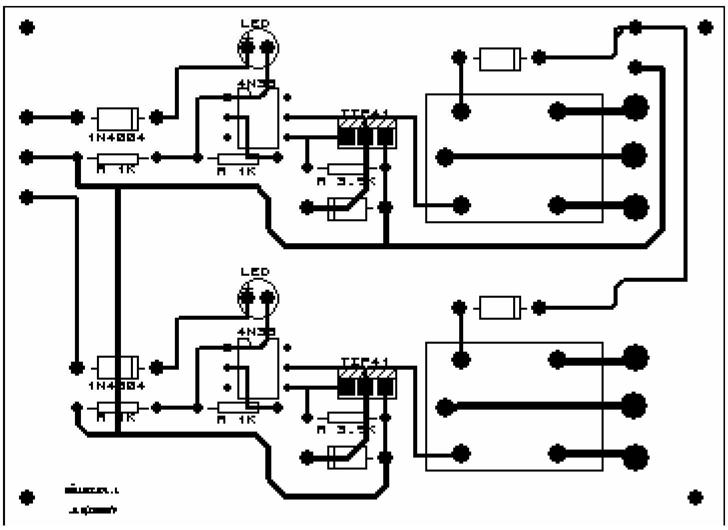


Figura b4

DIAGRAMA DE PINES DEL DSPIC30F4011

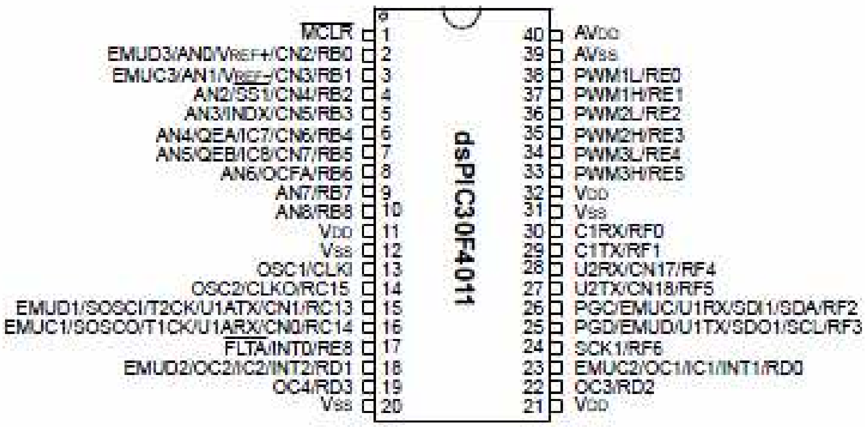


Figura b5

CONTROLADOR

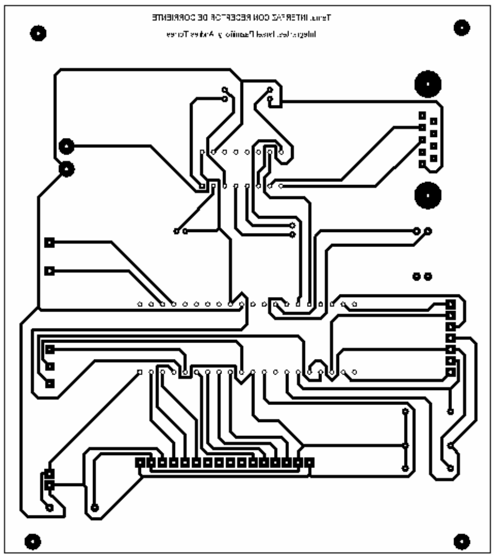


Figura b6

ANEXO C

TABLAS DE COSTOS

Acondicionador			
Descripción	**Cantidad**	**Precio Unitario**	**Precio Total**
Sensor LM35	1	$ 4.00	$ 4.00
Amplificadores Operacionales LF353	1	$ 0.50	$ 0.50
Resistencia de 1K -1/4 W	4	$ 0.05	$ 0.20
Trimmer Multivueltas de 10 K - 1/4W	1	$ 0.70	$ 0.70
Circuito impreso	1	$ 1.50	$ 1.50
Bases de 8 pines	1	$ 0.15	$ 0.15
Conectoras de tornillo de 2 pines para PBC	2	$ 0.20	$ 0.40
Conectoras de tornillo de 3 pines para PBC	1	$ 0.20	$ 0.20
Total			**$ 7.65**

Cabina de Simulación del Horno			
Descripción	**Cantidad**	**Precio Unitario**	**Precio Total**
Acrílico 20*20 cm	5	$ 1.35	$ 6.75
Ventiladores (VN 4051) 12 VDC	1	$ 5.00	$ 5.00
Foco de 100W	1	$ 1.00	$ 1.00
Silicona	1	$ 1.80	$ 1.80
Madera 60*20 cm	1	$ 4.00	$ 4.00
Total			**$ 18.15**

Transmisor de Corriente			
Descripción	**Cantidad**	**Precio Unitario**	**Precio Total**
Componentes Electrónicos			
Amplificadores Operacionales LF353	3	$ 0.50	$ 1.50
Resistencia de 1K- 1/4 W	1	$ 0.05	$ 0.05
Resistencia de 10K- 1/4 W	4	$ 0.05	$ 0.20
Resistencia de 470-1/4 W	1	$ 0.05	$ 0.05
Trimmer Multivueltas de 100 K	1	$ 0.70	$ 0.70
Trimmer Multivueltas de 50 K	2	$ 0.70	$ 1.40
Accesorios Electromecánicos			
Circuito Impreso	1	$ 3.00	$ 3.00
Conector Tipo Cerca de 3 pines	1	$ 0.30	$ 0.30
Bases para circuitos integrado de 8 pines	3	$ 0.15	$ 0.45
Conector en línea de 3 pines	1	$ 0.40	$ 0.40
Conectoras de tornillo de 2 pines para PBC	2	$ 0.20	$ 0.40
Jumper	1	$ 0.20	$ 0.20
Total			**$ 8.65**

Receptor Bucle de 4-20 mA			
Descripción	**Cantidad**	**Precio Unitario**	**Precio Total**
Circuito Integrado Fototransistor 4N25	1	$ 0.50	$ 0.50
Amplificadores Operacionales LF353	3	$ 0.50	$ 1.50
Resistencia de 390 - 1/4 W	1	$ 0.05	$ 0.05
Resistencia de 150- 1/4 W	1	$ 0.05	$ 0.05
Resistencia de 470 - 1/4W	1	$ 0.05	$ 0.05
Resistencia de 10K -1/4 W	6	$ 0.05	$ 0.30
Resistencia de 4.7K -1/4 W	2	$ 0.05	$ 0.10
Resistencia de 2.7K -1/4 W	1	$ 0.05	$ 0.05
Trimmer Multivueltas de 100 K	1	$ 0.70	$ 0.70
Trimmer Multivueltas de 50 K	1	$ 0.70	$ 0.70
Diodos rápidos de propósito General	3	$ 0.20	$ 0.60
Diodo zener de 5.1 V -1W	1	$ 0.30	$ 0.30
Diodos Led de 3mm, rojos	2	$ 0.30	$ 0.60
Circuito Impreso	1	$ 3.00	$ 3.00
Bases de 8 pines	4	$ 0.15	$ 0.60
Conector en línea de 3 pines	1	$ 0.40	$ 0.40
Conectoras de tornillo de 2 pines para PBC	1	$ 0.20	$ 0.20
Total			**$ 7.60**

Fuente de Alimentación de +5, +12 y -12 VDC/ 1A			
Descripción	**Cantidad**	**Precio Unitario**	**Precio Total**
IC-REG 7805	1	$ 0.50	$ 0.50
IC-REG 7812	1	$ 0.50	$ 0.50
IC-REG 7912	1	$ 0.50	$ 0.50
Puente Rectificador	1	$ 1.00	$ 1.00
Capacitor 2200uF/25V	2	$ 0.50	$ 1.00
Capacitor 0,1uF/50V	3	$ 0.50	$ 1.50
Jumper	1	$ 0.20	$ 0.20
Conectoras de tornillo de 3 pines para PBC	2	$ 0.20	$ 0.40
Transformador 14-014/2A	1	$ 5.00	$ 5.00
Cable 110V	1	$ 2.00	$ 2.00
Disipadores To-220	3	$ 0.70	$ 2.10
Circuito Impreso	1	$ 4.00	$ 4.00
Total			**$ 18.70**

Control Principal			
Descripción	**Cantidad**	**Precio Unitario**	**Precio Total**
Max232	1	$ 3.00	$ 3.00
Capacitores electrolíticos 10uF - 25V	4	$ 0.10	$ 0.40
Conector DB9 Hembra	1	$ 1.50	$ 1.50
DsPic 30F4011	1	$ 11.50	$ 11.50
Resistencia de 10K -1/4 W	3	$ 0.05	$ 0.15
Teclado	1	$ 5.00	$ 5.00
LCD	1	$ 10.50	$ 10.50
Conector en línea de 16 pines	1	$ 0.80	$ 0.80
Conector en línea de 7 pines	1	$ 2.80	$ 2.80
Conector en linea de 1 pines	1	$ 0.20	$ 0.20
Circuito Impreso	1	$ 9.20	$ 9.20
Total			**$ 35.85**

ANEXO D

Manual de Usuario

1. Conectar la cabina del horno a 110 Voltios 60 Hz.

2. Se debe conectar el cable en el puerto db9 al usb de la computadora para poder adquirir los datos de temperatura registrados por el sensor

3. En la LCD aparecerá un mensaje que le indica el rango de temperatura que debe ingresar para comenzar el proceso

4. Una vez ingresada la temperatura deberá presionar numeral para que empiece el proceso.

5. Ejecutar LabVIEW.

6. En LabVIEW usted observará como está llenándose el tanque donde ingresa el producto a calentarse. Usted podrá simular el sensor de nivel bajo presionando la tecla 3 y el sensor de nivel alto presionando la tecla 2 del teclado.

7. Inmediatamente se envía una señal para encender el foco (horno) el cual llegara hasta la temperatura que seteó en el principio.

8. Al llegar a esta temperatura se apagará y esperará que baje la temperatura y permitirá la apertura de una válvula para que pase al tanque mezclador.

9. Aquí podrá simular también el sensor de nivel alto con la tecla 5 lo cual activará el motor del mezclador y cerrará la válvula. Este proceso lo realizará por unos segundos y luego abrirá una válvula para extraer el producto.

10. Presionando la tecla 6 que es el sensor de nivel 4 indicamos que se ha vaciado el mixer con lo que la electroválvula 2 de cierra y el proceso se reinicia.

ANEXO E

GRAFICAS DE LABVIEW

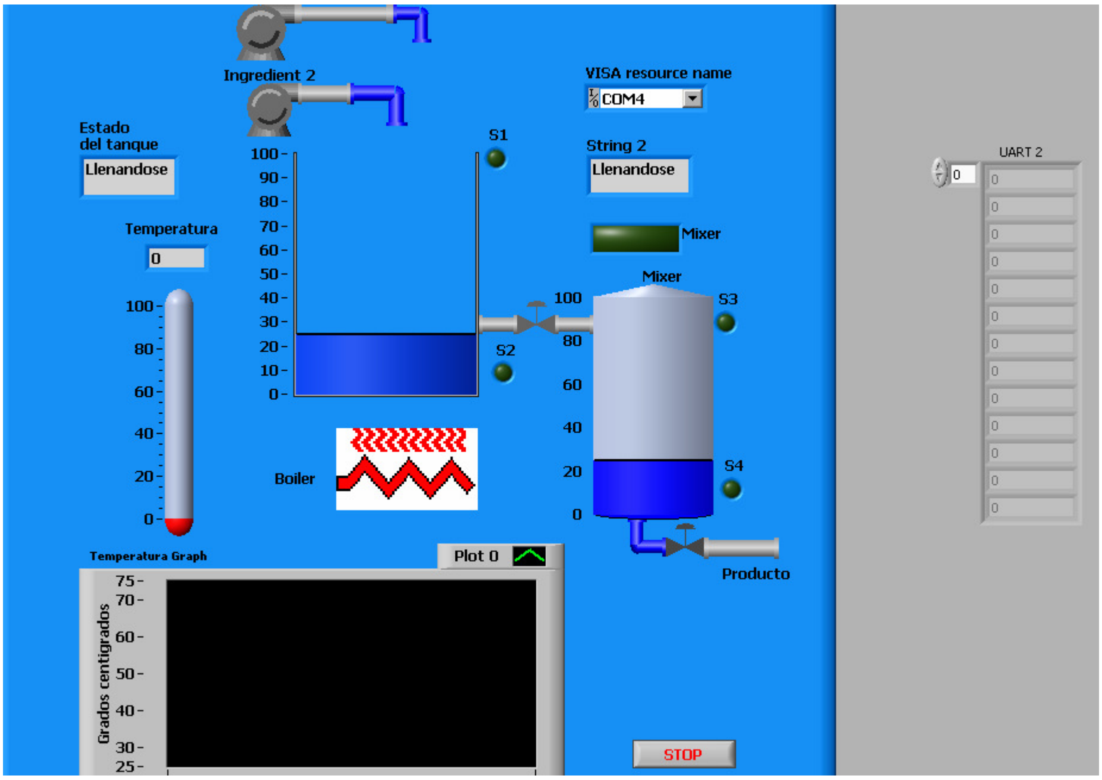
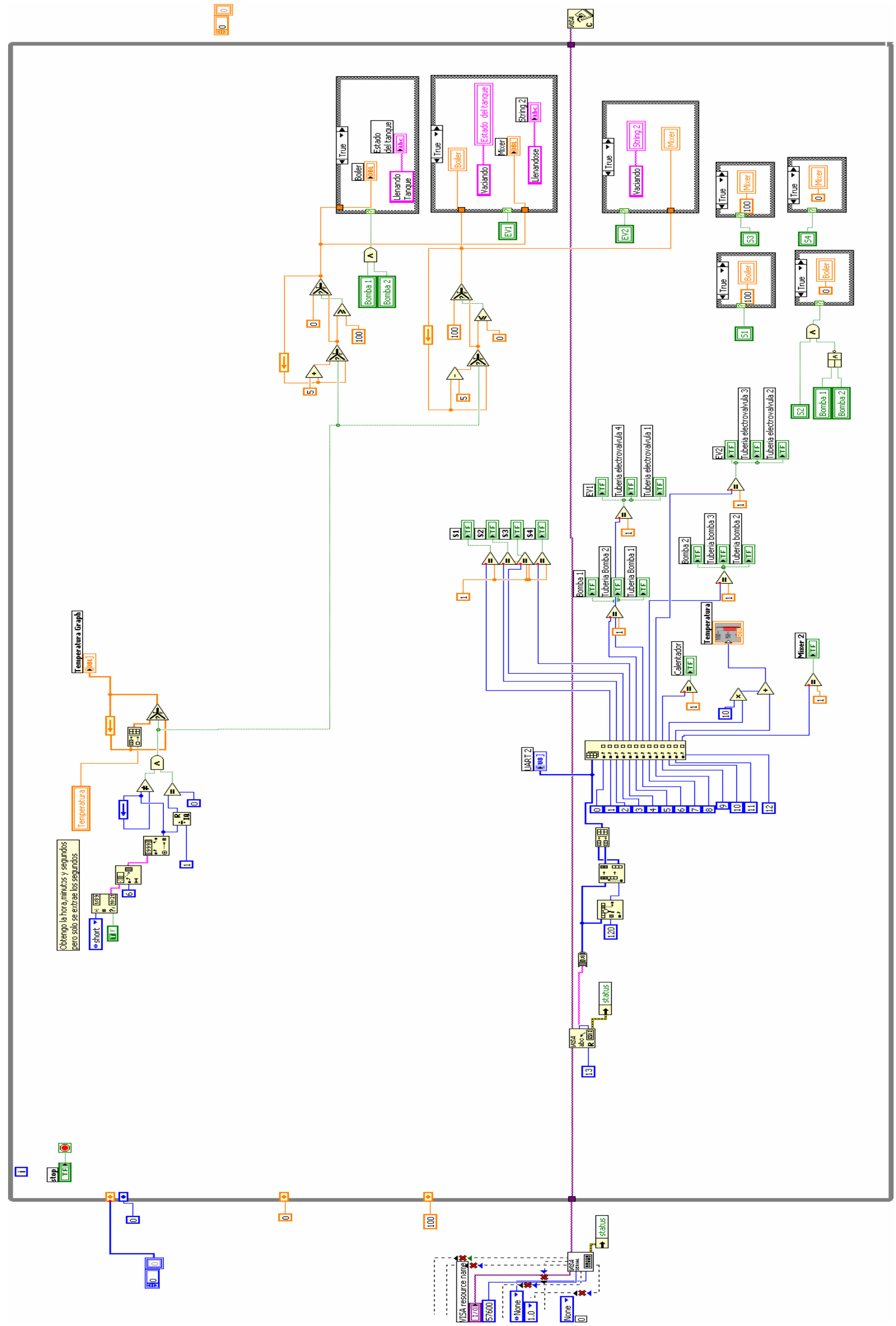


FIGURA E1


Temperatura Graph
Temperatura
Obtengo la hora,minutos y segundos pero solo se extrae los segundos
Boiler
Estado del tanque
Llenando Tanque
Vaciando
Mixer
String 2
Llenandose
EV1
EV2
Bomba 1
Bomba 2
S1
S2
S3
S4
Tuberia Bomba 2
Tuberia Bomba 1
Tuberia electrovalvula 4
Tuberia electrovalvula 1
Tuberia bomba 3
Tuberia bomba 2
Tuberia electrovalvula 3
Tuberia electrovalvula 2
Temperatura
Calentador
Mixer 2
UART 2
status
stop
VISA resource name
57600

BIBLIOGRAFIA

1. National Semiconductor, Manual del Sensor LM35: Precision Temperatura Sensors, (Edición mayo 1999) .

2. LabVIEW, Entorno Gráfico de programación, José Rafael Lajara Vizcaíno, José Pelegrí Sebastiá, Primera edición, MARCOMBO, Barcelona, España, 2007

3. URL: **www.datasheetcatalog.net**

4. Microchip, Hoja de datos del dsPIC30F4011, Feb 09, **http://ww1.microchip.com/downloads/en/DeviceDoc/70135F.pdf**

5. Microchip, Manual de referencia de la familia dsPIC30F, Feb 09, **http://ww1.microchip.com/downloads/en/DeviceDoc/70046E.pdf**.

6. Angulo Usategui José María, Begoña García Zapirain, Sáez Javie Vicente y Angulo Martínez Ignacio. Microcontroladores Avanzados dsPIC: Diseño práctico de aplicaciones, 1era edición. Ed. McGraw-Hill. 54-55, 308-310, 339-342, 364-368, 444-448.

7. Ing. Ramos Ramos Guillermo, Ing. Hernández Jorge Eduardo, Castaño Juan Andrés. Curso práctico de Electrónica Industrial y Automatización: Proyectos, Tomo 3. 193-196, 202-208.