ESCUELA SUPERIOR POLITECNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

INTERFAZ CON TRANSMISOR DE CORRIENTE EN APLICACIÓN CON DSPIC

PROYECTO DE LA MATERIA DE GRADUACIÓN

MICROCONTROLADORES AVANZADOS

PREVIA A LA OBTENCIÓN DEL TITULO DE:

INGENIERO EN ELECTRICIDAD ESPECIALIZACIÓN ELECTRONICA Y AUTOMATIZACIÓN INDUSTRIAL

Presentado por:

CHELE PLUA DIANA JACQUELINE

RAMIREZ MONTENEGRO MARIUXI DIANA

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 constante.

TRIBUNAL DE GRADUACION

Ing. Carlos Valdivieso A.	Ing. Hugo Villavicencio
DIRECTOR DE TESIS	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)

_____________________ _____________________

Diana Chele Plùa Mariuxi Ramírez

INDICE GENERAL

INDICE DE ILUSTRACIONES

FIGURAS

TABLAS

LISTA DE SÍMBOLOS

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 opam 1
VO2	Voltaje de salida opam 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

GLOSARIO

- **Archivo hex.:** Representación ASCII del código de máquina. Un archivo HEX está compuesto de registros que le especifican al micro controlador datos o instrucciones que serán ubicados en un dispositivo de memoria programable.

- **Sensor de temperatura:** Dispositivo que permite determinar la temperatura de un ambiente cerrado.

- **Circuito integrado:** Dícese así, al conjunto de dispositivos que generan salidas en base al voltaje que se le aplica en sus entradas.

- **Código fuente:** Archivo de texto que es procesado por un lenguaje ensamblador o un compilador para producir un archivo de objeto intermedio, o código de máquina que pueda ejecutarse en un microcontrolador.

- **Hardware:** Partes o componentes físicos que integran una herramienta; inclusive ella misma como una unidad.

- **Microcontrolador:** Dispositivo electrónico que contiene todas las características de una computadora.

- **Metodología:** Es el proceso, técnicas o enfoques empleados en la solución de un problema o en la creación de algo: un procedimiento particular o un conjunto de procedimientos.

- **Quemador:** Es un circuito que permite cargar un programa al microcontrolador.

- **Software:** Es un programa de computadora que permite al programador interactuar con la computadora.

RESUMEN

El aporte del presente proyecto, es presentar al lector la forma que puede implementar un sistema para controlar y monitorear la temperatura que existe en una cabina de secado de pintura, a través de la interfaz con un transmisor de corriente y un DsPic, el mismo que actúa como controlador del sistema.

El circuito fue implementado con el sensor LM35, que, entre sus características tiene la capacidad de medir hasta un máximo de 150 [ºC]. La señal a la salida de este sensor es analógica la misma que es amplificada, para luego llegar al transmisor de corriente y pasar por un receptor de bucle a uno de los canales analógicos del DsPIC30F4011.

El DsPic se encarga de realizar el control de la cabina de secado de pintura a través de los motores de impulsión y extracción que forman parte del sistema.

OBJETIVOS

- Implementar un proyecto o aplicación capaz de tomar una decisión en base a la lectura de la temperatura registrada en un ambiente cerrado por medio de una interfaz con un transmisor de corriente y un DsPIC.

- Promover el uso de DsPic para toma de decisiones demostrando que su costo es bastante bajo en comparación con el beneficio que se podrá obtener para determinados procesos.

- Evaluar herramientas y técnicas para la implementación de este tipo de proyectos, de manera que se busque un costo beneficio apropiado.

INTRODUCCION

En el campo industrial existen diferentes procesos, que consisten en el tratamiento completo de una serie de entradas de un dispositivo dando como resultado una acción.

En este proyecto, transmitimos datos de un sensor de temperatura, que controla a una cabina de secado de pintura y realizamos el respectivo control para mantener las condiciones de temperatura adecuadas.

Como todo proceso tiene un controlador, en este caso hemos utilizado un DsPIC30F4011, que posee módulos analógicos y la facilidad de construir el control por medio del software Mikrobasic.

Los datos de temperatura van a ser transferidos a través de un transmisor de corriente que controla voltajes comprendidos entre 0 y 10V a la entrada y entrega corriente entre 4 y 20 mA. Luego se procede a la visualización y graficación de datos por medio de Visual Basic.

1. TRANSMISOR DE CORRIENTE

1.1 Que es un transmisor de corriente

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

El principio de la telemedida por corriente es de gran uso en la transmisión de señales débiles en los entornos industriales, por su relativa facilidad de implementación, confiabilidad y bajo costo.

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. Otra ventaja importante de la telemedida por corriente, que depende del tipo de transmisor, es la posibilidad de configurar la técnica de enlace modificando el número de hilos necesarios para transmitir la señal y la alimentación, permitiendo de esta forma, reducir considerablemente las inducciones parásitas sobre los cables cuando estos son de gran longitud.

1.2 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.

1.3 Diagrama de bloques del transmisor de corriente

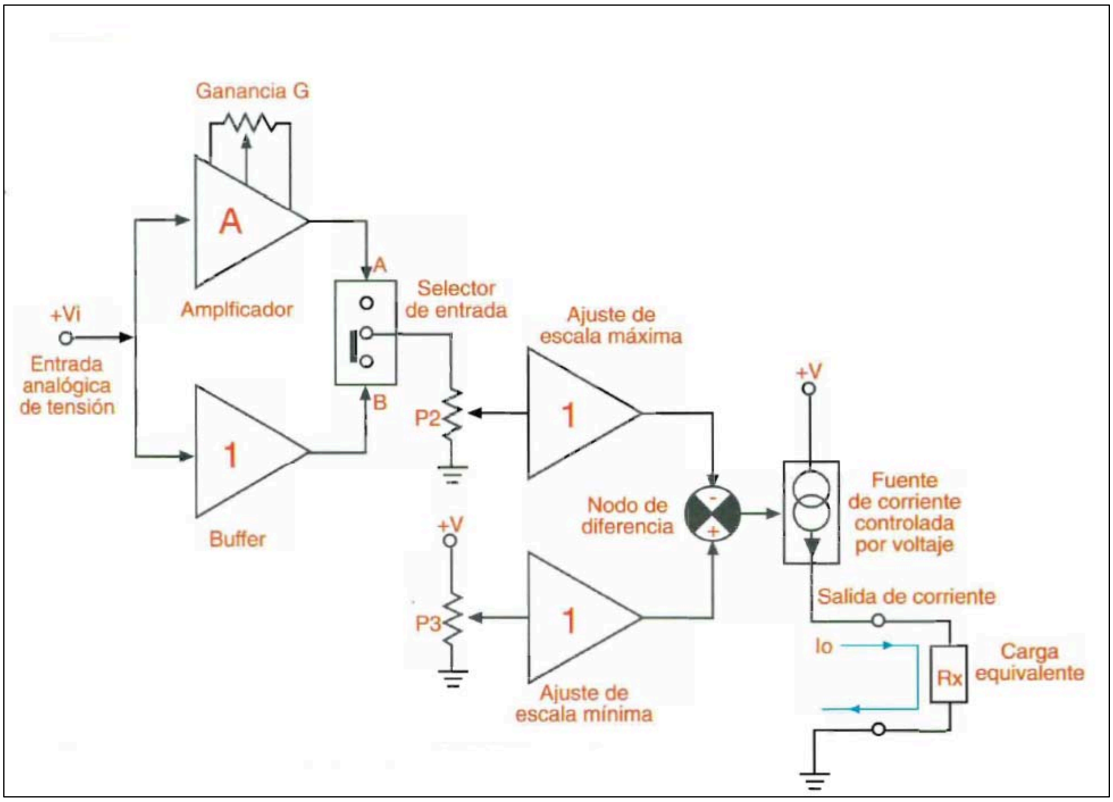


Figura 1

1.4 Diagrama esquemático del transmisor de corriente

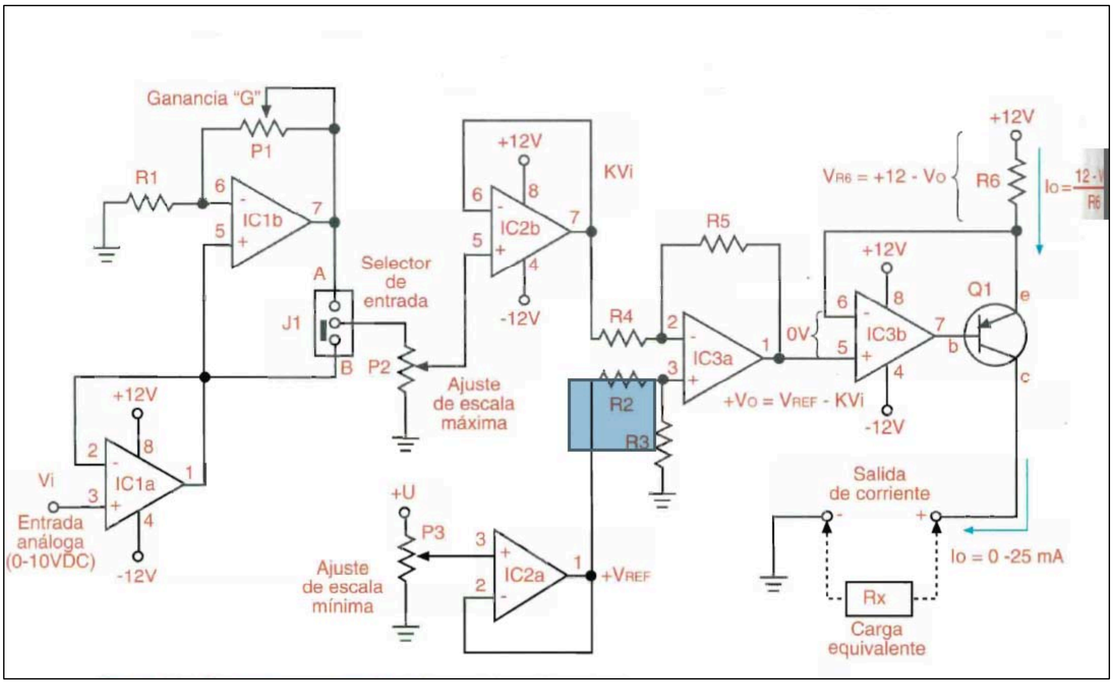


Figura 2

1.5 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.

1.5 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

2. RECEPTOR DE BUCLE DE 4-20mA

2.1 Que es un receptor bucle de 4-20mA

El receptor de bucle de 4-20mA son circuitos utilizados para comunicar los sensores con dispositivos de lectura y visualización cercanas o remotas.

La transmisión del voltaje continuo proporcional a la magnitud medida que los sensores entregan solo se recomienda para cortas distancias, ya que en entornos industriales los bucles formados por los conductores captarán tensiones parásitas inducidas, las cuales podrán alterar las magnitudes de la señal de medida.

La medición por medio de bucles de corriente se efectúa convirtiendo la magnitud medida por el sensor en una corriente continua proporcional, que se envía a través de un conductor y se lee en el extremo receptor en forma de tensión usando una resistencia conocida.

2.2 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.

2.3 Diagrama de bloques del receptor de bucle

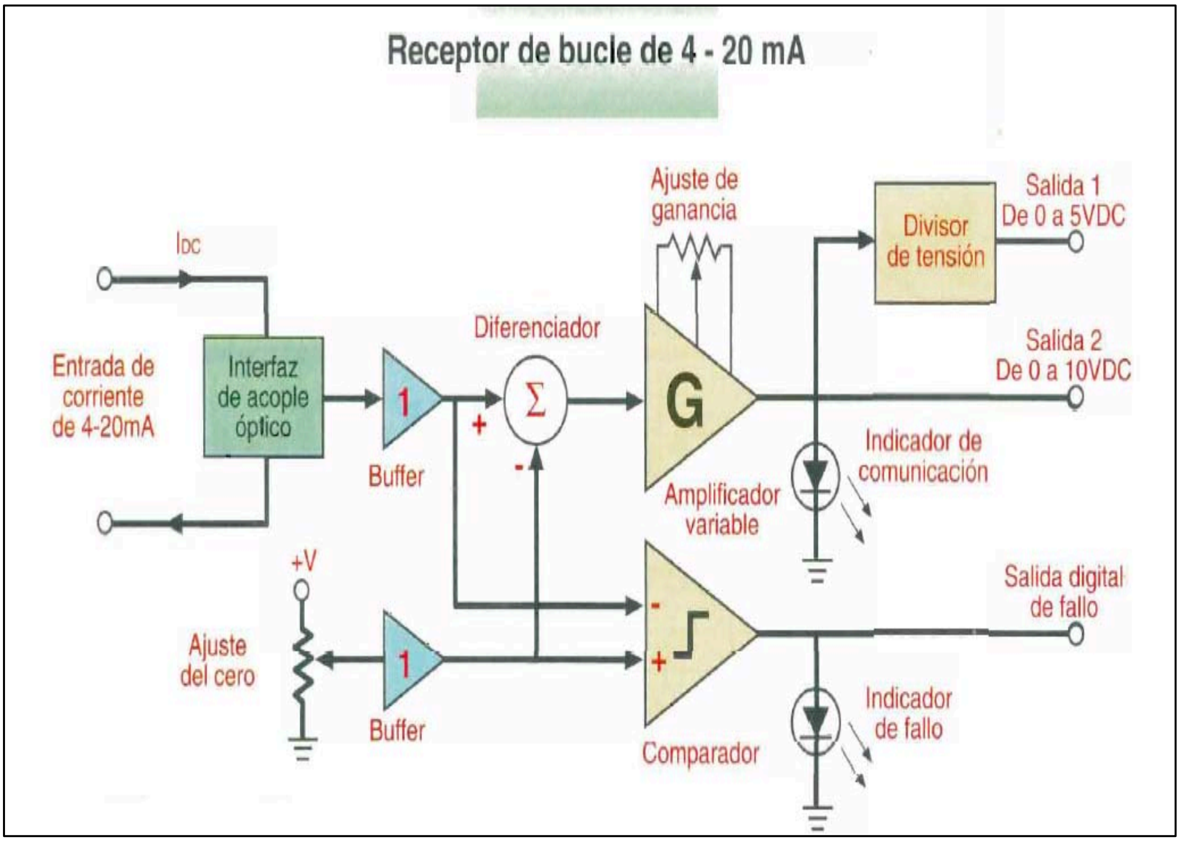


Figura 3

2.4 Diagrama esquemático del Receptor bucle

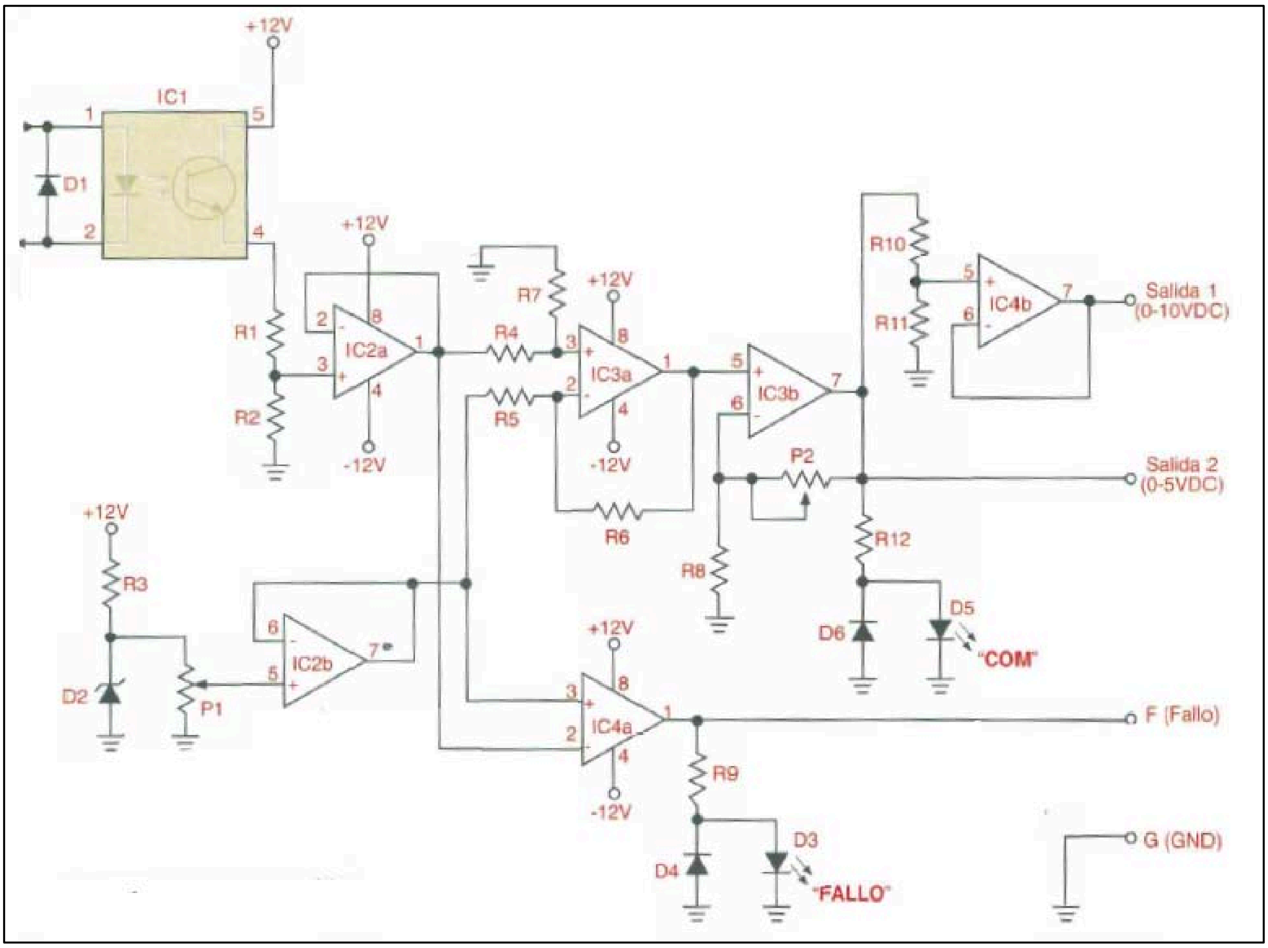


Figura 4

2.5 Funcionamiento

El primer bloque del circuito es la interfaz óptica tal como se muestran los diagramas en la **figura**. Este se encarga de aislar galvánicamente la corriente que entra y convertirla en una tensión continua proporcional a la misma. Para ello se utiliza un optotransistor NPN, configurado como seguidor de emisor y diseñado para operar en su zona lineal.

Los seguidores de tensión, implementados con **IC2**, se encargan de acoplar los voltajes entregados por el optotransistor y el trimmer **P1**. Este último, se usa para calibrar los 0V a la salida.

La siguiente etapa es un amplificador diferencial básico de ganancia unitaria, cuyo propósito es restar la tensión equivalente a 0V del voltaje de entrada proporcional a la corriente medida. Esto se hace para poder finalmente amplificarla usando el operacional **IC3b** configurado como amplificador no inversor de ganancia variable entre 1 y 10 tal como se puede ver en el diagrama de la figura. La salida 2, de 0 a 10VDC, se obtiene directamente del amplificador no inversor, para obtener la salida entre 0 y 5VDC se usa una red de resistencias y un buffer. Ambas salidas son monitoreadas por medio del led de comunicación **D5 (COM)**, cuya intensidad varía proporcionalmente con la corriente de entrada. Por medio de la salida de fallo **(F)**, se puede detectar la entrada de corrientes inferiores a 4mA, o la ruptura del cable de comunicación, mediante un valor digital igual a 12VDC proveniente del comparador inversor. El LED **D3** actúa como indicador del mismo.

2.6 Montaje y calibración

Una vez que se haya ensamblado el circuito, con los materiales indicados, se procede a calibrar el circuito de la siguiente manera.

1. Los trimmers P1 y P2 deben de colocarse en el punto medio.

2. Alimente el circuito a través del conector de potencia con una fuente bipolar de + 12V, 0 y -12V.

3. Aplique una corriente de 4 mA a través del conector de la entrada y mida con un voltímetro DC en cualquiera de sus salidas 1o 2, un voltaje igual a 0VDC mientras se ajusta el trimmer P1.

4. Para finalizar el proceso de calibración, haga circular a través de la entrada una corriente de 20 mA y varíe el trimmer P2 para modificar el factor de ganancia del amplificador hasta obtener en la salida 2 un voltaje igual a 10 VDC.

3. Sensor LM35

3.1 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 pero el más común es el TO-92 de igual forma que un típico transistor con 3 pines, dos de ellos para alimentarlo (VCC - GND) y el tercero (VOUT) nos entrega un valor de tensión proporcional a la temperatura medida por el dispositivo.

3.2 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.

- 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

- Como ventaja adicional, el LM35 no requiere de circuitos adicionales para su calibración externa cuando se desea obtener una precisión del orden de ±0.25ºC a temperatura ambiente, y ±0.75 ºC en un rango de temperatura desde 55 a 150 ºC.

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

- Debido a su baja corriente de alimentación (60uA), se produce un efecto de auto calentamiento reducido, menos de 0.1 ºC en situación de aire estacionario.

3.3 Encapsulado

El sensor se encuentra disponible en diferentes encapsulados pero el más común es el TO-92, una cápsula comúnmente utilizada por los transistores de baja potencia, como el BC548 o el 2N2904.
Tiene tres pines: alimentación (VCC), tierra (GND) y salida (OUT). Este sensor es fabricado por Fairchild y Nacional Semiconductor.

3.4 Acondicionador de señal para LM35

El viejo sensor de temperatura LM35 es un clásico en los circuitos comerciales y de aficionados.

Este pequeño sensor (y varios de sus "parientes") entrega diez milivoltios por cada grado centígrado.

Permite una precisión importante, pudiendo leerse fracciones de grado. Pero para ello es necesario hacer un adecuado tratamiento de la señal, ya que al trabajar con tensiones tan pequeñas, cualquier ruido o interferencia puede hacernos tomar una lectura errónea, o a veces, errática

El voltaje de salida es de 10mV por grado, 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

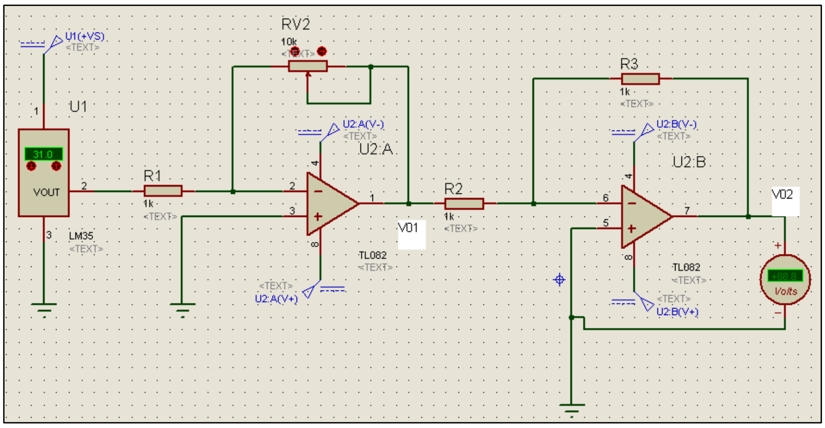


Figura 5

Sacando la ecuación de amplificación tendremos que

$$\frac{V01}{Vi} = -\frac{Rv2}{R1} = -\frac{6.5}{1}$$

$$\frac{V02}{V01} = -\frac{R3}{R2} = -1$$

$$\frac{V02}{Vi} = -\frac{Rv2}{R1} * -\frac{R3}{R2} = 6.5$$

Es decir que la ganancia obtenida seria de 6.5.

4. DsPic 30F4011

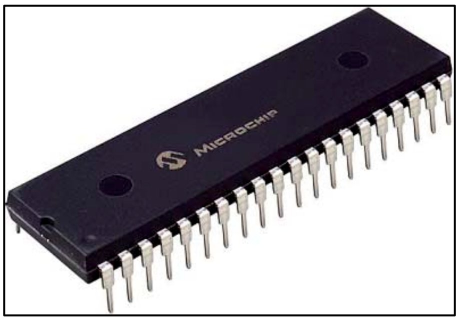


Figura 6

4.1 Introducción

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 electro medicina, la robótica, las comunicaciones, las reproducciones 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 micro controladores 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 los acoplo con la posibilidad de numerosos periféricos complementarios para dar lugar a los DSP.

Los DSP son procesadores digitales cuyo diseño ha sido enfocado para soportar las especificaciones del tratamiento de señales, que por sus implicaciones en los campos tecnológicos más demandados recientemente, supone disponer de un dispositivo programable con los recursos físicos y lógicos precisos para las exigencias de dicha áreas.

Microchip Technology Inc. en los últimos años, ha querido poner en manos de sus clientes unos nuevos componentes, llamados dsPIC, que le permita introducirse en las aplicaciones contemporáneas del proceso digital de señales de forma sencilla basada en arquitecturas y repertorios de instrucciones conocidos.

El DsPic es un Controlador Digital de Señales (DSC) que está basado en las características más potentes de los micro controladores de 6 bits y que se incorpora las principales prestaciones de los DPS, lo que facilita enormemente la transición entre los diseños clásicos y los complejos propios del procesamiento digital de señales.

4.2 Que 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.

Microchip, fabricante de los DsPic, los ha bautizado con el nombre de DSC (Controlador Digital de Señal). Un DSC es potente micro controlador de 16 bits al que se le han añadido las principales capacidades de los DSP. Es decir, los DSC poseen todos los recursos de los mejor micro controladores 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 micro controlador: 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. Además, su precio es similar al de los micro controladores.

4.3 Estructura del DsPic30F4011

El principal componente del sistema de control es el micro controlador dsPIC30F4011, el cual tiene un núcleo RISC con arquitectura Harvard modificada de 16 bits. Su estructura se puede separar en tres partes: microprocesador, integración del sistema y periféricos.

La CPU tiene palabras de instrucción de 24 bits con un campo variable para el código de operación. Tiene además 16 registros de trabajo que junto con instrucciones tipo MCU y DSP realizables en un ciclo de instrucción proveen una gran rapidez y complejidad de cálculo.

La memoria de datos puede contener 32 mil palabras o 64 mil bytes que se separan en dos bloques cada uno con su propia AGU (generadores de direcciones). La arquitectura DSP contiene un multiplicador de 17 bits, una ALU, dos acumuladores de 40 bits, un registro de desplazamiento bidireccional de 40 bits, todo esto permite realizar un procesamiento de señales en tiempo real de manera óptima.

4.4 Características del DsPic30F4011

CARACTERÍSTICAS	CAPACIDAD
Frecuencia de operación	EC a 40 MHz
Memoria de Instrucción (Bytes)	16 K
Memoria de Programa FLASH	48 K
Memoria de Datos RAM (Bytes)	2 K
Memoria EEPROM (Bytes)	1 K
Fuentes de Interrupción	30
Puertos de Entrada/Salida	5 (Puerto B,C,D,E,F)
Instrucciones Básicas	83
Temporizadores de 16 bits	5
Adicionalmente timer de 32 bits	2
Módulos de PWM	6
Comunicaciones Seriales	SPI,I2C,UART,CAN
Canales de Conversión Analógica Digital de 10 bits	9
Módulo Encoder de Cuadratura de 16 bits	1
MOTOR DSP	

- Multiplicador rápido 17 X 17 bits.
- Registro de desplazamiento de 40 bits.
- Sumador restador de 40 bits.
- Dos registros acumuladores de 40 bits.
- Modos de operación: redondeo y saturación lógica.
- Todas las instrucciones son de un solo ciclo.

Todos los procesadores de los DsPic incorporan en su procesador las siguientes características:

Arquitectura Hardware: posee dos memorias una para datos y otra para instrucciones.

Procesador RISC: posee un conjunto de instrucciones optimizadas para soportar el lenguaje C.

Cauce segmentado: permitiendo elevar el rendimiento del procesador, consiguiendo así que una instrucción empiece a ejecutarse antes de que hayan terminado las anteriores y, por tanto, que haya varias instrucciones procesándose simultáneamente.

Integración de Recursos propios de DSP: el aporte más considerable es el motor DSP, facilitando la resolución de operaciones matemáticas complejas, en los algoritmos para el rápido procesamiento de señales.

Esquema de la arquitectura Hardware del DsPic

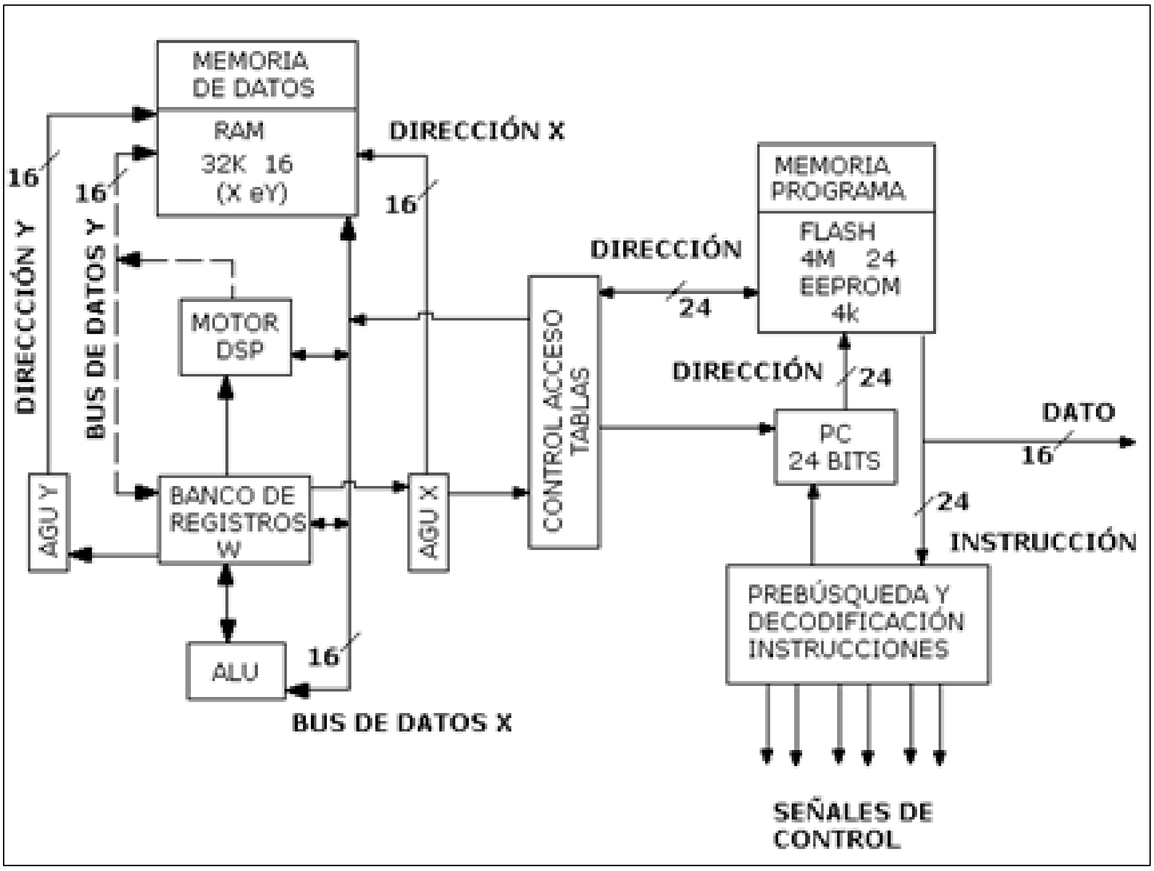


Figura 7

Se observa en el diagrama la división de dos grandes bloques de memoria, la memoria de programa FLASH direccionada por el contador de programa de PC, el segundo bloque formado por la memoria RAM de datos subdividida en dos partes X y Y, controlados por los generadores de direcciones AGU X y AGU Y. También existe un Banco de registros de 16 bits que alimenta al motor DSP.

4.5 Conversor Análogo Digital

Generalmente la señal analógica se obtiene de un sensor o transductor de la magnitud a medir, el cual proporciona en su salida una señal eléctrica cuyo valor está comprendido entre un mínimo y un máximo, pudiendo admitir infinitos valores intermedios.

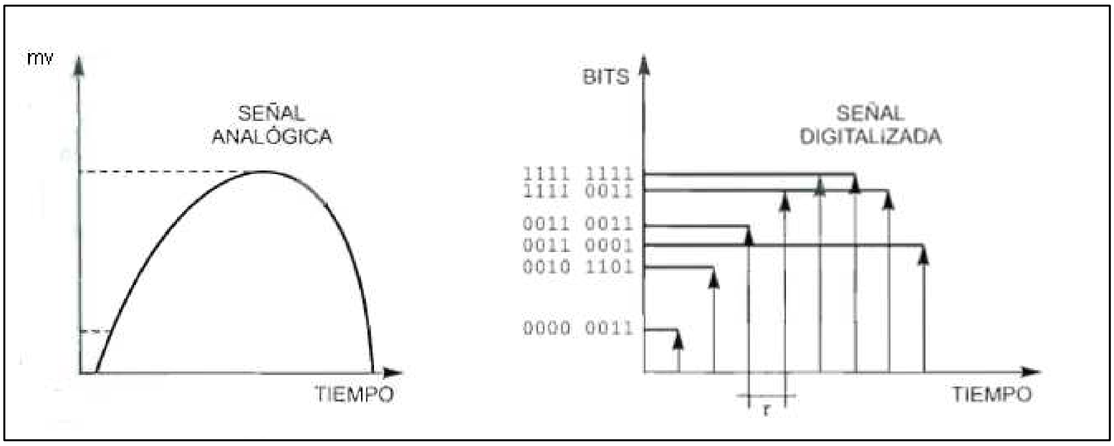


Figura 8

Adquisición de Señal Analógica

En la figura se representa el esquema clásico de la digitalización de la señal analógica procedente del transductor. Inicialmente un conmutador electrónico se cierra cada periodo de tiempo y captura una muestra que la aplica al condensador C que se carga y mantiene su carga mientras el conversor análogo digital realiza su transformación a un valor binario equivalente.

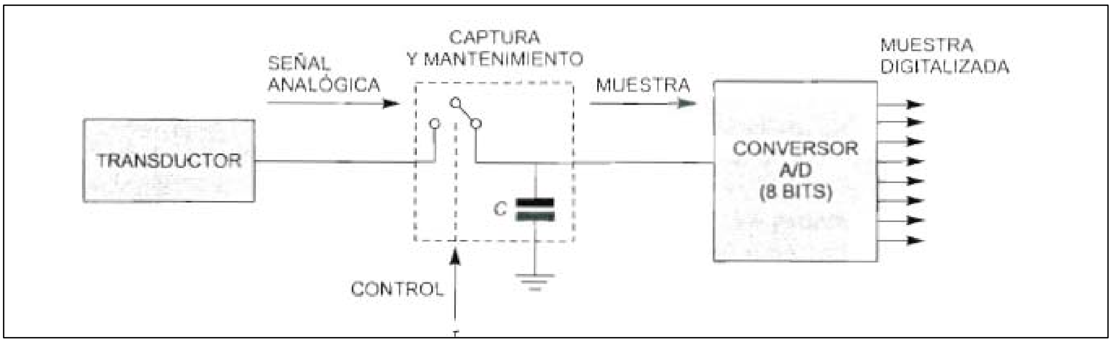


Figura 9

En este proceso de digitalización hay dos factores predominantes que son:

1. La frecuencia a la que se capturan las muestras. Para obtener unos resultados aceptables dicha frecuencia debe duplicar como mínimo la frecuencia máxima contenida en la señal analógica.

2. El número de bits que compone el valor digital de las muestras. Como se deduce fácilmente cuantos más bits tenga la muestra convertida, mayor será la precisión en los cálculos posteriores y disminuirán los errores generados.

El convertidor analógico digital tiene las siguientes características:

- Resolución de 10 bits.
- Tiempo de muestreo de 154 ns.
- Conversión por aproximaciones sucesivas.
- Velocidad de conversión de hasta 500 Ksps a 5V (2 us de tiempo de conversión) y 100 Ksps a 2,7V.
- Hasta 16 patitas analógicas de entrada.
- Patitas de entrada destinadas a soportar voltaje de referencia externo.
- Cuatro amplificadores unipolares diferenciales de muestreo y retención.
- Muestreo hasta de cuatro patitas de entrada analógica de forma simultánea.
- Modo automático de exploración de canal.
- Fuente seleccionable para relleno del buffer.
- Cuatro opciones de alineamiento del resultado.
- Funciona mientras la CPU se encuentra en los modos Sleep.

Tiene seis registros de control, los cuales se nombran a continuación:

- **ADCON1**: Registro de control analógico digital 1.
- **ADCON2**: Registro de control analógico digital 2.
- **ADCON3**: Registro de control analógico digital 3.
- **ADCHS**: Registro de selección de canal analógico de entrada.
- **ADPCFG**: Registro de configuración del puerto en analógico o digital.
- **ADCSSL**: Registro de selección de entradas para la exploración.

Los registros ADCON1, ADCON2 y ADCON3 son los encargados de controlar la operación del módulo analógico digital, mientras que el registro ADCHS selecciona las patitas de entrada que van a ser conectadas a los amplificadores de muestreo y retención.

El registro ADPCFG se encarga de configurar las patitas como entradas analógicas o como entradas o salidas digitales. Finalmente el registro ADCSSL se dedica a seleccionar las entradas que serán exploradas secuencialmente.

4.6 Tipos de osciladores

Existen dos osciladores internos. El FRC (RC rápido) trabaja a 8 MHz Esta diseñado para poder trabajar a frecuencias altas sin necesidad de conectar un cristal externo.

El segundo oscilador interno LPRC (RC de potencia baja) está conectado al Perro Guardián y trabaja a 512 KHz. Hace de fuente de reloj para el temporizador PWRT, Perro Guardián, y los circuitos de monitorización de reloj.

El único oscilador externo disponible (EXTRC) trabaja a frecuencias que llegan a los 4 MHz Utiliza una resistencia y un condensador externo conectado a la patita OSCI, la cual también puede conectarse a una señal de reloj externa (modo EC).

La frecuencia del oscilador RC es función de :

- Voltaje de alimentación.
- Valor resistivo externo (REXT).
- Valor capacitivo externo (CEXT).
- Temperatura de funcionamiento.

Diagrama del sistema oscilador

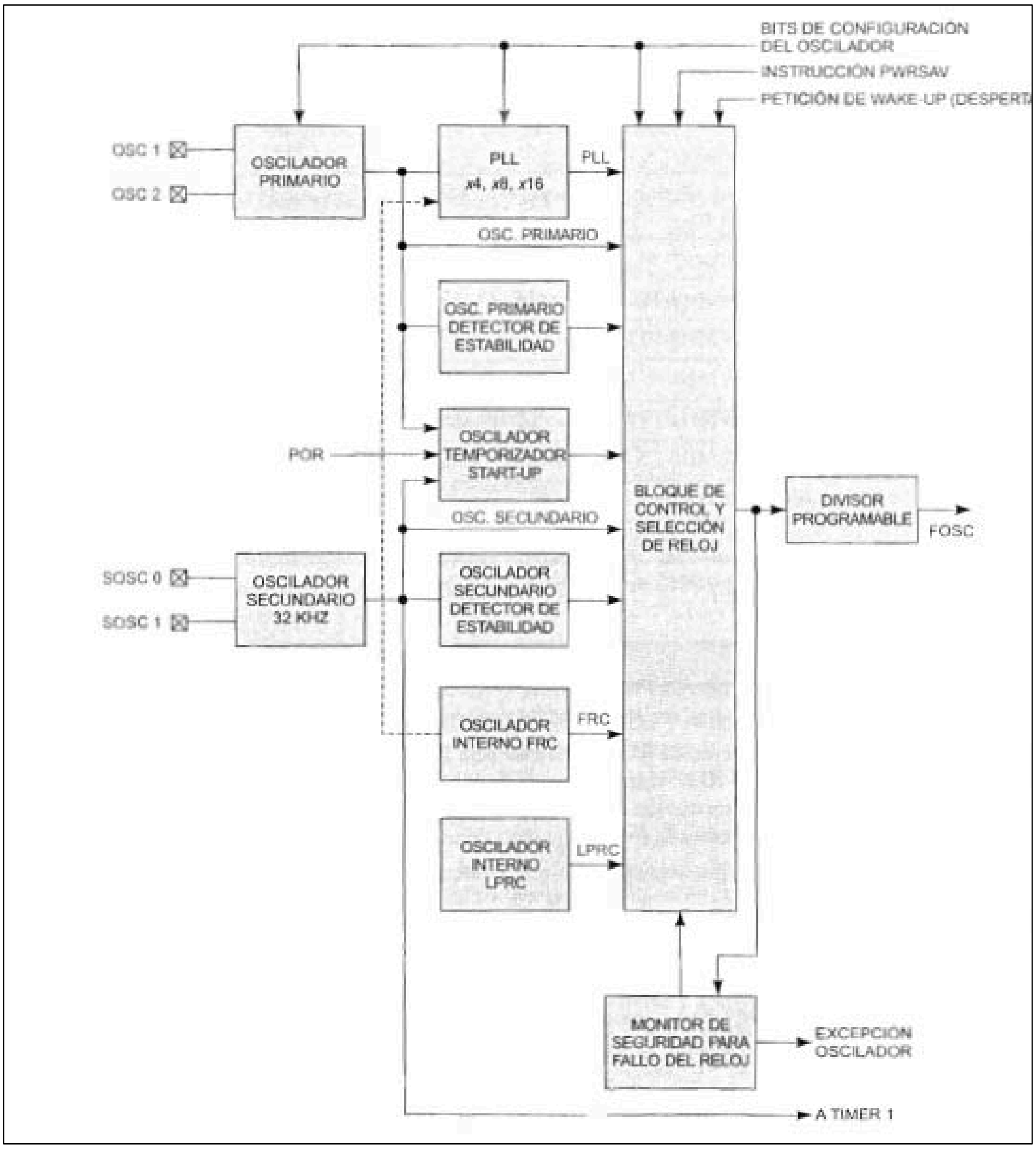


Figura 10

4.7 Modulo UART

El UART (Transmisor Receptor Universal Asíncrono) es un módulo para la comunicación serie. Funciona como un sistema de comunicación bidireccional asíncrono que puede adaptarse a multitud de periféricos, como ordenadores personales o interfaces RS-232 y RS-485.

Estas son las principales característica del módulo UART:

- Posibilidad de trabajar con paridad par, impar o sin paridad.
- Uno o dos bits de STOP.
- Contiene un generador de baudios con un prescaler de 16 bits que se encarga de generar la secuencia de trabajo del módulo.
- Buffer de transmisión con capacidad para 4 caracteres.
- Buffer de recepción con capacidad para 4 caracteres.
- Posibilidades de emplear interrupciones para indicar la finalización de la transmisión o de la recepción.
- Pines específicos TX y RX para transmitir y para recibir datos.

El módulo UART está formado por tres grandes bloques:

- Generador de baudios

- Transmisor asíncrono.
- Receptor asíncrono.

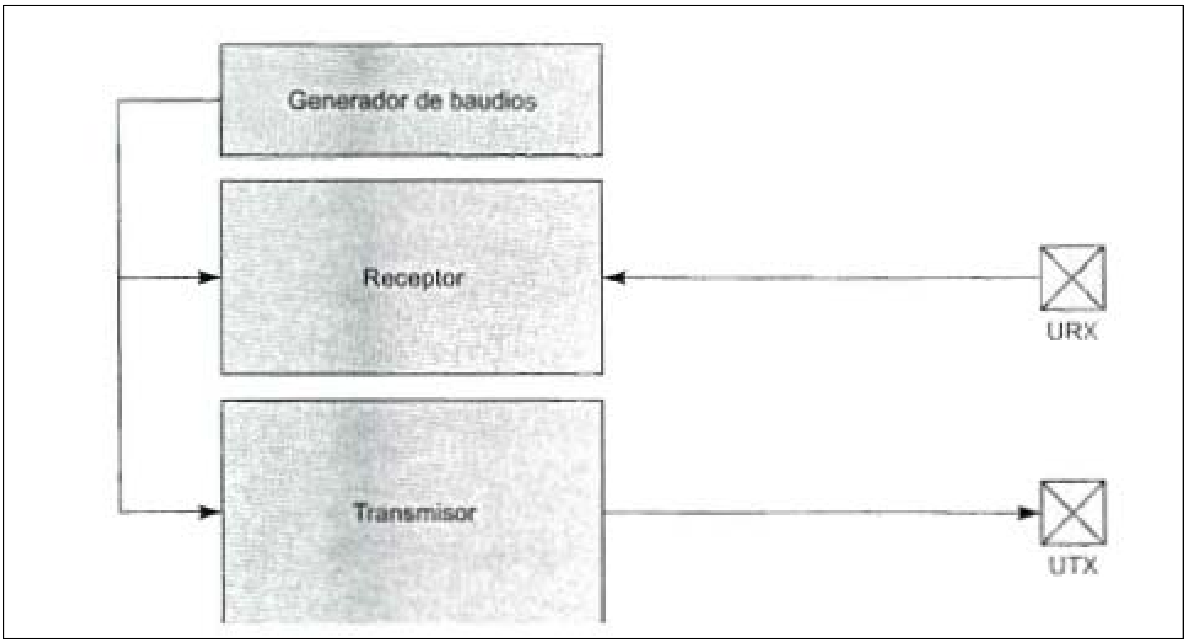


Figura 11

En el protocolo asíncrono RS-232 le frecuencia en baudios (bit por segundo) a la que se realiza la transferencia de información debe tomar un valor de baudios normalizado: 330, 600, 1200, 4800, 9600, etc. Para generar esta frecuencia el UART dispone de un generador de frecuencia en baudios, cuyo valor es controlado por el contenido del registro UBRG.

UBRG = (Fosc/(16- frecuencia en baudios))

5. Controladores

5.1 Objetivos

Aprender para qué sirve un regulador

Conocer y comprender los reguladores PD, PI.

Aprender técnicas de ajuste de los parámetros de estos reguladores .

5.2 Introducción

Un controlador automático compara el valor real de la salida de una planta con la entrada de referencia (el valor deseado), determina la desviación y produce una señal de control que reduce la desviación a cero o a un valor pequeño. La manera en la cual el controlador automático produce la señal de control se denomina acción de control. La figura siguiente muestra el diagrama de bloques de un sistema de control industrial que consiste en un controlador automático, un actuador, una planta y un sensor (elemento de mediación).

Esquema de bloques de un controlador

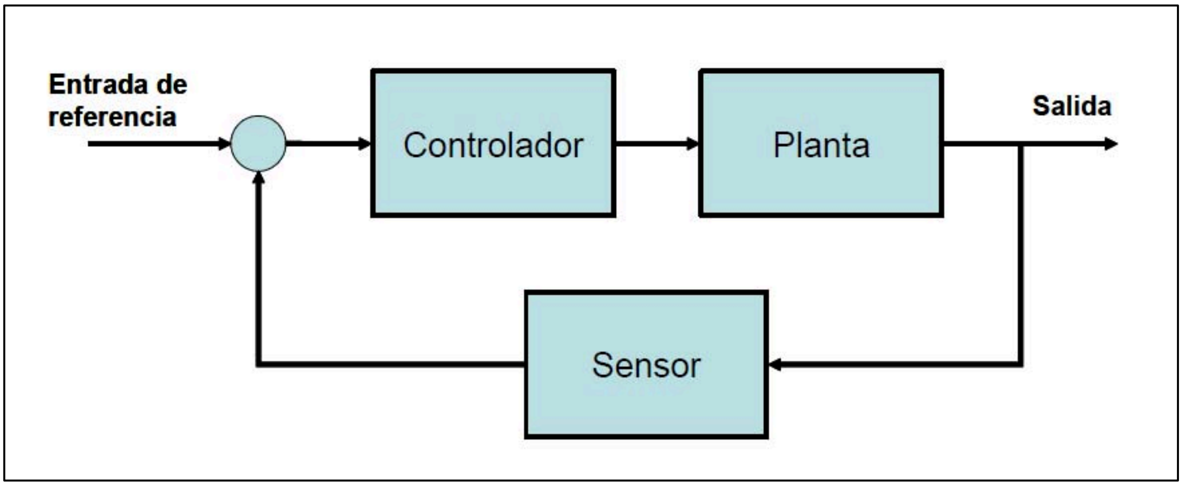


Figura 12

5.3 Tipos de Controladores

Los controladores industriales se clasifican, de acuerdo con su acción de control, como:

Controladores on-off.

Controladores proporcionales.

Controladores integrales.

Controladores proporcionales-derivativos

Controladores proporcionales-integrales.

Controladores proporcionales-integrales-derivativos.

5.4Controlador proporcional

Para un controlador con una única acción proporcional, la relación entre la salida del controlador u(t) y la señal de error e(t) es:

$$u(t) = K_p e(t)$$

la cual, utilizando la transformada de Laplace, se convierte en:

$$u(s) = K_p e(s)$$

De las ecuaciones anteriores se puede observar claramente que el controlador proporcional es utilizado para “controlar teniendo en cuenta el presente”, es decir, el error actual es multiplicado por una ganancia constante (Kp) y aplicado al actuador. Como es obvio, cuando el error es cero, la salida de este regulador también es cero, por lo que junto a la

señal de control proporcional habría que añadir un offset, o también conocido como vías que permitiese al valor de salida seguir a la señal de referencia.

5.5 Controlador integral

Para un controlador integral, el valor de la salida de la acción de control (u(s)) se cambia una relación proporcional a la integral del error, es decir:

$$u(t) = K_I \int_0^t e(t)\ dt$$

$$\frac{U(s)}{E(s)} = \frac{K_i}{s}$$

Al contrario que el controlador proporcional, el integral pretende “controlar teniendo en cuenta el pasado” debido a que el error es integrado (o sumado) hasta el tiempo actual, y entonces multiplicado por una ganancia. Si utilizásemos exclusivamente el controlador proporcional normalmente aparecería un error en estado estacionario, es por ello que se suele utilizar el término integral. Imaginemos por ejemplo un tanque en el que podemos controlar el flujo de entrada mientras que el de salida es constante. En esta situación, con un simple controlador proporcional más un término vías podríamos mantener una referencia de nivel. Sin embargo, si el flujo de salida variase (imaginemos que se obstruye la tubería de salida), el nivel de referencia no se mantendría existiendo un error en estado estacionario. Este problema se resolvería simplemente con cambiar el término bias con una acción integral la cual ajustaría su valor eliminando los errores que han permanecido durante cierto.

5.6 Controlador proporcional-integral.

En un controlador proporcional-integral, la acción de control se define mediante.

$$u(t) = K_p e(t) + \frac{K_p}{T_i} \int_0^t e(t)\ dt$$

siendo la función de transferencia del controlador:

$$\frac{U(s)}{E(s)} = K_p \left(1 + \frac{1}{T_i s} \right)$$

Donde Ti es el tiempo integral.

6. Visual Basic

6.1 Introducción

Visual Basic para aplicaciones es una combinación de un entorno de programación integrado denominado Editor de Visual Basic y del lenguaje de programación Visual Basic, permitiendo diseñar y desarrollar con facilidad programas. El término para "aplicaciones" hace referencia al hecho de que el lenguaje de programación y las herramientas de desarrollo están integrados con las aplicaciones del Microsoft Office (en este, caso el Microsoft Excel), de forma que se puedan desarrollar nuevas funcionalidades y soluciones a medida, con el uso de estas aplicaciones.

El Editor de Visual Basic contiene todas las herramientas de programación necesarias para escribir código en Visual Basic y crear soluciones personalizadas.

6. APLICACION CABINA DE SECADO DE PINTURA

6.1 Introducción

La característica de brillo sin pulir de las pinturas de dos componentes llamadas 2K (poliuretanos, barnices, etc.) que en la actualidad se esta utilizando en nuestro mercado, hacen indispensable realizar la operación de pintado y secado en un recinto presurizado libre de polvos, que brinde la posibilidad de aprovechar las ventajas de estos productos de secarse y endurecerse rápidamente con la ayuda de calor. Estas características son las que se consiguen al trabajar con una Cabina de secado de pintura, la cual aporta grandes ahorros de tiempo de secado, disminución de trabajos adicionales y el aumento significativo de la productividad y la calidad del taller demostrado a través de la gran cantidad de trabajos entregados en término que se pueden realizar en un mínimo lapso de tiempo.

6.2 Funciones de una cabina de secado

- Contrariamente a lo que se suele pensar, la función principal de estos equipos no es solo la de secar las piezas recién pintadas, sino la de tener una zona para pintar limpia y libre de polvos que generan incrustaciones en la pintura y obligan a pulir las piezas y/o retocarlas.

- La otra función de las Cabinas de Pintura es el secado de las piezas o de los vehículos pintados. Con este proceso se obtiene un secado parejo y controlado de toda la superficie pintada continuando este en un ambiente limpio y sin polvo.

- En caso de pintarse un vehículo con materiales acrílicos el pulido es obligatorio, más allá de las incrustaciones de basura que haya, ya que este proceso es necesario para obtener brillo. Las nuevas pinturas de dos componentes (2K) ya sean Poliuretanos (monocapa) o Barnices (para Bicapa) no necesitan pulido ya que tienen brillo propio. En caso de tener que realizar esta operación se requerirá un esfuerzo mucho mayor que para hacerlo en las pinturas acrílicas, debido a la mayor dureza de los productos.

6.3 Funcionamiento

Una vez aplicada la pintura de acabado, esta se puede secar a la temperatura ambiente, 20ºC aproximadamente, o acelerar el proceso de secado elevando la temperatura a unos 60-80 ºC en una cabina de secado aparte o en la misma cabina en la que se ha aplicado la pintura.

Por lo general, la propia cabina de pintura donde se ha aplicado la pintura actúa también como horno de secado, circulando el aire en su interior a una temperatura que oscila entre los 60 y los 80ºC. Por lo tanto, estas cabinas tienen dos fases de funcionamiento: una fase de pintado, con un determinado caudal de aporte de aire, a una velocidad determinada y calentando el aire introducido a unos 20 ºC; y una segunda fase de secado en la que el caudal y la velocidad pueden ser menores, y se eleva la temperatura a unos 60-80 ºC. En esta fase de secado, el aire aspirado del exterior antes de ser impulsado al "plenum" de distribución, es recirculado a través del intercambiador de calor en una proporción aproximada del 65% para un mayor aprovechamiento energético.

7. CONCLUSIONES Y RECOMENDACIONES

- El diseño implementado cumple con el objetivo principal que es controlar la temperatura en una cabina de secado de pintura utilizando un controlador PI, el mismo que nos permite mantener la temperatura constante aun cuando haya perturbaciones en el medio.

- El transmisor de corriente es muy utilizado en la industria para comunicar transductores con controladores, PLC's y computadores a grandes distancias, actúa como interfaz entre el sensor y el controlador, además permite que se puedan tomar decisiones dependiendo de la corriente generada en su salida.

- El receptor bucle se acopla perfectamente a un transmisor de corriente y es utilizado de acuerdo a las necesidades requeridas, en nuestro caso lo utilizamos para que la entrada que lee el canal análogo del DsPIC sea voltaje.

- El acondicionador de la señal del sensor es muy fácil de construir, tiene bajo costo y la ganancia es ajustable debido al potenciómetro que se está utilizando.

- La utilización de Visual Basic fue necesaria para la observación de datos además para comprobar que son transmitidos correctamente a través del puerto serial.

- Debido a las bondades que nos brinda el DsPic se pudo controlar el disparo del TRIAC para la regulación de voltaje alterno en la carga y de esta manera poder controlar de mejor forma la perturbación a la cual fue sometida el sistema.

- El uso de los modos de control, es siempre conforme a las características del proceso, lo cual significa que debemos entender bien la operación del proceso antes de automatizarlo y de proceder a las rutinas de los algoritmos de control.

- El modo On-Off, es un caso especial del modo proporcional aplicable solamente a un proceso estático, ya que la ganancia del On-Off es infinita (B.P.=0). Cuando solo deseamos dar estabilidad al proceso, el modo proporcional es suficiente.

- Por último, los parámetros de diseño por lo general son totalmente rígidos y basados en objetivos y criterios de diseño, con los que muchas veces no se pueden cumplir o que se requiere de soluciones extremadamente complejas, por lo tanto se debe saber y tener muy en cuenta entre lo que se quiere hacer y lo que los recursos tecnológicos nos pueden ofrecer.

ANEXO A

PROGRAMACION EN MIKROBASIC

```
program proyecto_3_3_09

'***********************************************************************************
'********************************TECLADO**************************************
'******************************* 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
'***********************************************************************************
dim n_current,n_current2,num,n_before as word 'Para Ingresar La Temp x
TECLADO
dim TXT1,F_SENSOR_TXT    as string[30]
dim SENSOR8              as word          'Para los LM35
dim F_SENSOR8            as WORD        'f_lm35
dim W_SENSOR8            as WORD         'W_lm35
DIM SET_TEMP             AS FLOAT
'DIM SET_TEMP_TXT         AS CHAR[20]
DIM FLAG1                AS BOOLEAN
DIM MARU                 AS FLOAT
'***********************************************************************************
dim cnt0 as byte
dim cnt1,cnt2 as longword
```

```
'****************************VARIABLES
TIMER1**********************************
DIM CONT       AS WORD             'CONTADOR DEL TIMER1 CADA 35useg
DIM temp       AS WORD             'VALOR FINAL DEL TIMER1, ANCHO DE
PULSO
DIM I,J,K      AS WORD             'I=TIMEMPO REAL,J=CRUCE POR CERO,
K=INT/.5SEG
DIM ESPERA     AS WORD             'DELAY NECESARIO PA COMPLETAR EL
CICLO
'****************************VARIABLES DE
PI2*********************************
DIM ER     AS FLOAT
DIM ERA    AS FLOAT

DIM P      AS FLOAT
DIM IN     AS FLOAT

DIM KP     AS FLOAT
DIM KI     AS FLOAT

DIM U      AS FLOAT
'*********************************UART***************************
*************
DIM   entero1  AS WORD
DIM   entero2  AS WORD
DIM   diana    AS WORD
'*********************************TIMER1*************************
*************
'*****************************************************************
***********
sub procedure Timer1Int org $1A
  '** it is necessary to clear manually the interrupt flag:
  IFS0 = IFS0 and $FFF7   ' Clear TMR1IF
  IF IFS0 <> $FFF7 THEN
    INC (CONT)           ' INCREMENTO DE CONTADOR
  END IF
  IF K>14286 THEN        ' 500MS
    K=0
  END IF
end sub
'*****************************************************************
***********
'*******************************ANCHO DE
PULSO*******************************
```

```
SUB PROCEDURE SHOOT()
J=0
 while (adc_read(3)<>512) and (j=0) 'CRUCE POR CERO
     nop
 wend

 CONT=0
 j=1
 IFS0  = IFS0 and $FFF7  ' Clear TMR1IF
 IEC0  = IEC0 or  $0008  ' Enable Timer1 interrupts

 WHILE CONT <temp        ' 4ms aprox. SI TEMP=114
     I=I+1
     INC(K)
     PORTE.1=1         ' envia pulsos al triac
 WEND

 IEC0.3  = 0            ' DESHABILITO EL TIMER1
 PORTE.1=0              ' detiene pulsos al triac
 ESPERA=229-TEMP
 IFS0  = IFS0 and $FFF7  ' Clear TMR1IF
 IEC0  = IEC0 or  $0008  ' Enable Timer1 interrupts

 WHILE CONT <ESPERA      ' DELAY
  NOP
 WEND

 IEC0.3  = 0            ' DETIENE EL TIMER1
END SUB
'*********************************************************************
***********
'*********************************FUNCION
PI2**********************************
'EL MUESTREO LO REALIZA CADA 0.5 SEGUNDOS
SUB PROCEDURE PI2()

sensor8=ADC_READ(8)
f_sensor8= sensor8 / 6.81
ER = set_temp- f_sensor8
P=KP*ER
ERA=ERA+ER
IF ERA>set_temp THEN
  ERA=set_temp
ELSE
```

```
   NOP
END IF
IN=KI*ERA
U=P+IN
END SUB

'**********************************************************************
***********
'*****************************FUNCION
TECLADO************************************
'****************Function_KeyPad: Retorna el ASCII para la
LCD******************

sub function kypad (dim ky as byte) as byte

      fa=0
      if flag1=0 then delay_ms(50) 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 delay_ms(50) 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 delay_ms(50) 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 delay_ms(50) 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
```

```
'********************************************************************
***********
'********************************************************************
***********

'********************************************************************
***********
'********************************ANCHO DE
PULSO*********************************
 sub procedure usart( dim temperatura as  word )
   dim entero1, entero2,diana as word

   entero1 = (temperatura div 10)             'Primer digito

   uart2_write_char(convert2(entero1))
   entero2 = (temperatura mod 10 )             'Segundo digito

    uart2_write_char(convert2(entero2))
   diana = 0x20

   uart2_write_char(diana)   'espacio

  if f_sensor8 >= set_temp   then
       f-sensor 8


 end sub



'********************************************************************
***********
'********************************ANCHO DE
PULSO********************************
main:
'**********************************TMR1****************************
*************
uart2_init(57600)
T1CON = $8000           ' Timer1 ON, internal clock FCY, prescaler 1:1
TRISE = $0000           ' PORTE as output
ADPCFG= $FeF7           ' Make PORTB digital input
IPC0  = IPC0 or  $1000  ' Interrupt priority level = 1

FLAG1 =0
```

```
porte =$00

trisb.3=1
trisb.8= 1
trisc=$f000            'PORTC columnas son entradas  del kyboard
trisd=$00              'PORTD filas son salidas del kyboard
trise=$00

KP=0.5
KI=0.02

lcd_init(portb,7,6,5,4, portb,0,2,1)
lcd_cmd(lcd_cursor_off)
lcd_cmd(lcd_clear)

'*****************************************************************
***********
'********************** Inicializando Variables
*****************************
 cnt1=0 cnt2=0   porte=$00
'*****************************************************************
***********
'*****************************************************************
***********
                                                            cnt1=0
cnt2=0   porte=$00
   lcd_out(1,1," ***Control***") lcd_out(2,3,"Temperatura")
   delay_ms(2000)
s2: lcd_cmd(lcd_clear)
   lcd_out(1,2,"Set temp:") '
   lcd_out(2,5,"49<Tm<61")
   delay_ms(2000)
   lcd_cmd(lcd_clear)
   lcd_out(1,2,"Temp set:")
   lcd_cmd(lcd_second_row)
   n_before=0   n_current=0    num=0

'******** Ingresando Temp x Teclado:       n_current/n_before/num
tecla:

   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  tecla  end if
      if n_current2=35 then   break     end if
      if n_current2=42 then
        lcd_cmd(lcd_clear)
        lcd_out(1,8,"*") lcd_out(2,2,"Tecla_Incorrecta")
        delay_ms(2000 )lcd_cmd(lcd_clear) lcd_out(1,2,"Temp Set:")
        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 = #

    lcd_cmd(lcd_clear)
    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)    ' grados centigrados
       delay_ms(2000)
    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)
       delay_ms(2000)
       goto s2
    end if

I= 0

DO
IF K=0 THEN
  SET_TEMP = num
  PI2()
ELSE
  lcd_cmd(lcd_clear)
  sensor8=ADC_READ(8)' lectura de señal analoga
  f_sensor8 = (sensor8 / 6.81)
  WORDTOSTR (f_sensor8,F_SENSOR_TXT)
  LCD_OUT(2,1, "READ=")
  LCD_OUT(2,7, F_SENSOR_TXT)
  lcd_chr(2,15,111)
```

```
   lcd_chr(2,16,67)    ' grados centigrados
   lcd_out(1,1,"SET:")
   wordtostr(num,txt1)
   lcd_out(1,6,txt1)
   lcd_chr(1,15,111)
   lcd_chr(1,16,67)    ' grados centigrados
   DELAY_MS(300)
   usart(f_sensor8)          'uart_write_char()
    delay_ms(500)
END IF

temp=(U*229)/((KP+KI)*set_temp)

loop until I>142857        'PROCURA MANTENER LA TEMPERATURA POR 5
MINUTOS
lcd_cmd(lcd_clear)
LCD_OUT(1,1, "DONE")
'*********************************************************************
***********
'********************FIN DEL PROCESO ENFRIANDO
CABINA***************************
DO
  PORTE.0 =1
  sensor8=ADC_READ(8)
  f_sensor8 = (sensor8 / 6.81)
  usart(f_sensor8)          'uart_write_char()
  delay_ms(500)
  FLOATTOSTR (F_SENSOR8,F_SENSOR_TXT)
  LCD_CMD(LCD_CLEAR)
  LCD_OUT(2,1, "READ=")
  LCD_OUT(2,7, F_SENSOR_TXT) lcd_chr_cp(223) lcd_chr_cp(67)
LOOP UNTIL f_sensor8<35

PORTE.0 =0
'*********************************************************************
***********

'*********************************************************************
***********
end.
```

CODIGO FUENTE DEL PROGRAMA REALIZADO EN PROGRAMACION EN VISUAL BASIC

```
Option Strict Off
Option Explicit On
Friend Class Form1
        Inherits System.Windows.Forms.Form

  Dim ExcelApp As Microsoft.Office.Interop.Excel.Application
        Dim ExcelCht As Microsoft.Office.Interop.Excel.Chart
        Dim ExcelSht As Microsoft.Office.Interop.Excel.Worksheet
        Dim ExcelWkb As Microsoft.Office.Interop.Excel.Workbook
  Dim MyExcel As Boolean
  Dim i As Integer
  Dim Dato As New ArrayList




        Private Sub Command1_Click(ByVal eventSender As System.Object, ByVal eventArgs
As System.EventArgs) Handles Command1.Click

                On Error Resume Next

                Err.Clear()

                ExcelApp = GetObject( , "Excel.Application")
                If Err.Number <> 0 Then
                        Err.Clear()
                        ExcelApp = CreateObject("Excel.Application")
                        If Err.Number <> 0 Then
                                MsgBox("Error: " & Err.Description)
                        Else
                                MyExcel = True

                        End If

                Else
                        MyExcel = False

                End If

                ExcelApp.Visible = True



        End Sub
```

```
        Private Sub Command2_Click(ByVal eventSender As System.Object, ByVal eventArgs
As System.EventArgs) Handles Command2.Click

                ExcelWkb = ExcelApp.Workbooks.Add
                ExcelSht = ExcelWkb.Worksheets(1)



        End Sub

        Private Sub Command3_Click(ByVal eventSender As System.Object, ByVal eventArgs
As System.EventArgs) Handles Command3.Click



     For index As Integer = 0 To Dato.Count - 1
       ExcelSht.Cells._Default(index + 1, 1) = Dato.Item(index)
     Next


   End Sub



        Private Sub Command5_Click(ByVal eventSender As System.Object, ByVal eventArgs
As System.EventArgs) Handles Command5.Click


     Dim n As Microsoft.Office.Interop.Excel.Range = ExcelSht.Rows.CurrentRegion
     Dim num As Integer = n.Count
     Dim rango As String = "A1:D" + CType(num, String)

                ExcelCht = ExcelWkb.Charts.Add
                ExcelCht.ChartType = Microsoft.Office.Interop.Excel.XlChartType.xlLineMarkers
     ExcelCht.SetSourceData(ExcelSht.Range(rango),
Microsoft.Office.Interop.Excel.XlRowCol.xlColumns)
                ExcelCht.HasTitle = True

                ExcelCht.Axes(Microsoft.Office.Interop.Excel.XlAxisType.xlCategory,
Microsoft.Office.Interop.Excel.XlAxisGroup.xlPrimary).HasTitle = True

        ExcelCht.Axes(Microsoft.Office.Interop.Excel.XlAxisType.xlCategory,
Microsoft.Office.Interop.Excel.XlAxisGroup.xlPrimary).AxisTitle.Characters.Text = "X-Axis"

        ExcelCht.Axes(Microsoft.Office.Interop.Excel.XlAxisType.xlValue,
Microsoft.Office.Interop.Excel.XlAxisGroup.xlPrimary).HasTitle = True
                ExcelCht.Axes(Microsoft.Office.Interop.Excel.XlAxisType.xlValue,
Microsoft.Office.Interop.Excel.XlAxisGroup.xlPrimary).AxisTitle.Characters.Text = "Data Series"



        End Sub
```

```
        Private Sub Command6_Click(ByVal eventSender As System.Object, ByVal eventArgs
As System.EventArgs) Handles Command6.Click

                ExcelCht.ChartArea.Select()
                ExcelCht.ChartArea.Copy()

                                Image1.Image = My.Computer.Clipboard.GetImage()


        End Sub



        Private Sub Form1_Load(ByVal eventSender As System.Object, ByVal eventArgs As
System.EventArgs) Handles MyBase.Load
     MSComm1.CommPort = 5
     MSComm1.Settings = "57600,n,8,1"
     MSComm1.RThreshold = 1
     MSComm1.PortOpen = True
                Randomize()

        End Sub

  Private Sub Recibir_Click(ByVal eventSender As System.Object, ByVal eventArgs As
System.EventArgs) Handles Recibir.Click
     recibido.Text = recibido.Text + MSComm1.Input
  End Sub



  Private Sub Splite_Click(ByVal eventSender As System.Object, ByVal eventArgs As
System.EventArgs) Handles Splite.Click



     Dim valor() As String
     valor = Split(recibido.Text, " ")
     For index As Integer = 0 To valor.Length - 1
       Dato.Add(valor(index))
       rtbDatos.Text = rtbDatos.Text + valor(index)
       If index < valor.Length - 1 Then
         rtbDatos.Text = rtbDatos.Text + ","
       End If

     Next

  End Sub

  Private Sub Image1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Image1.Click

  End Sub
```

```
    Private Sub recibido_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles recibido.TextChanged

    End Sub

End Class
```

ANEXO B

DETALLE DE PLACAS REALIZADA PARA EL

PROYECTO

1. Diagrama del Transmisor de corriente

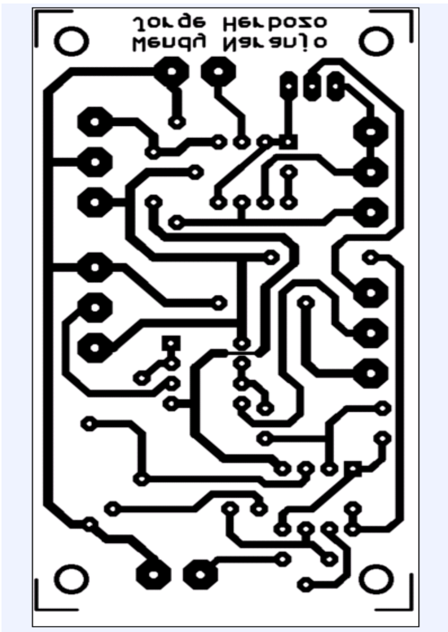


Figura B.1

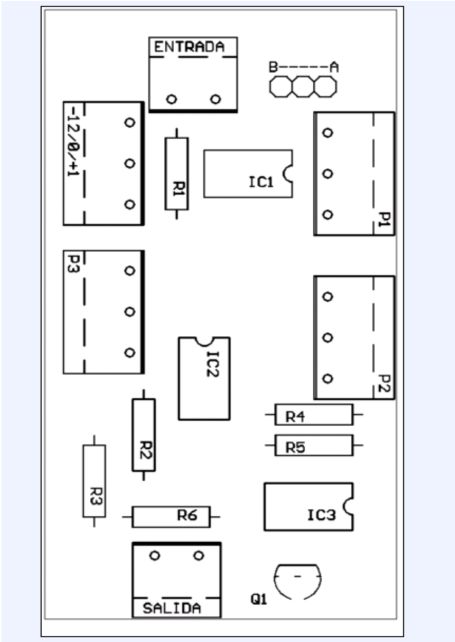


Figura B.1

2. Diagrama del Receptor bucle

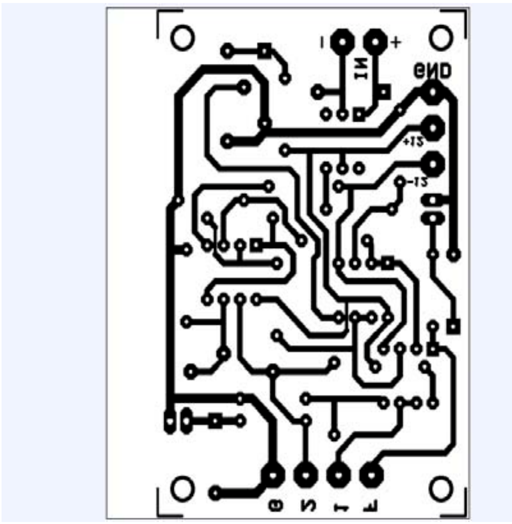


Figura B.2

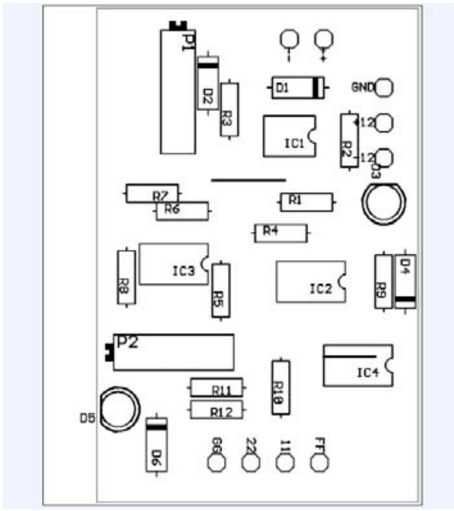


Figura B.2

3. Diagrama del sensor LM35

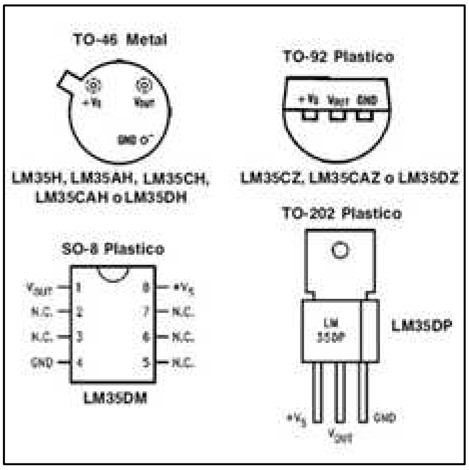


Figura B.3

4. Diagrama del acondicionador de señal

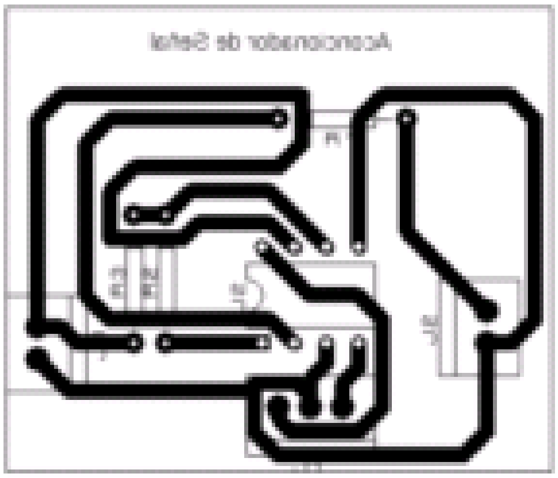


Figura B.4

5. Diagrama del circuito del ventilador

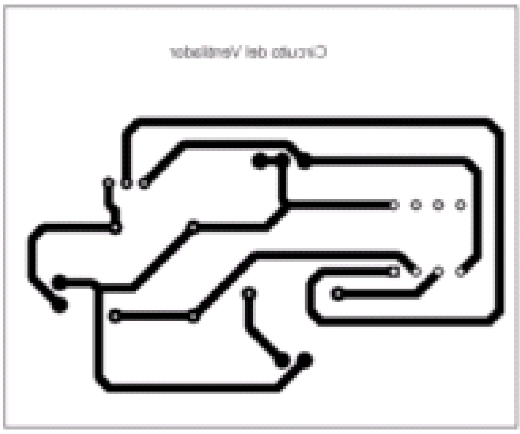
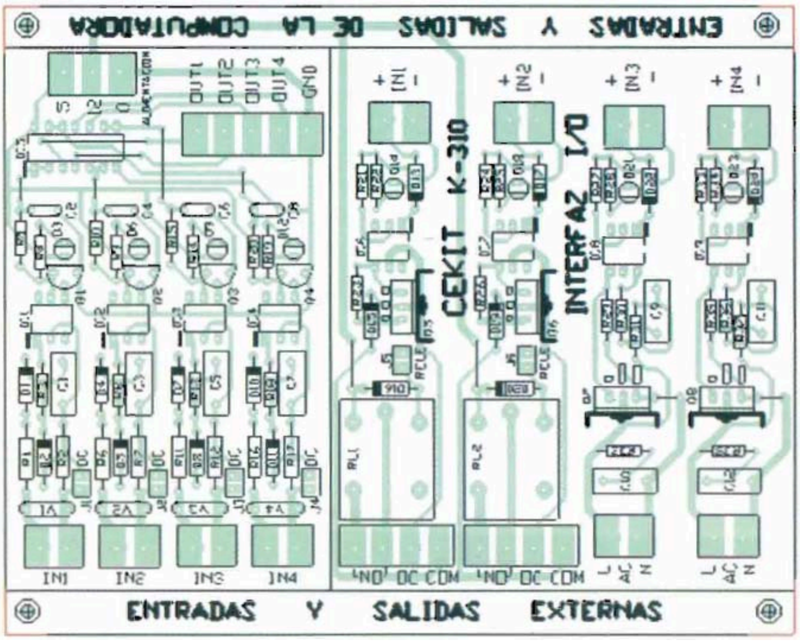


Figura B.5

6.Diagrama del circuito del foco



7. Diagrama de pines del DsPic30F4011

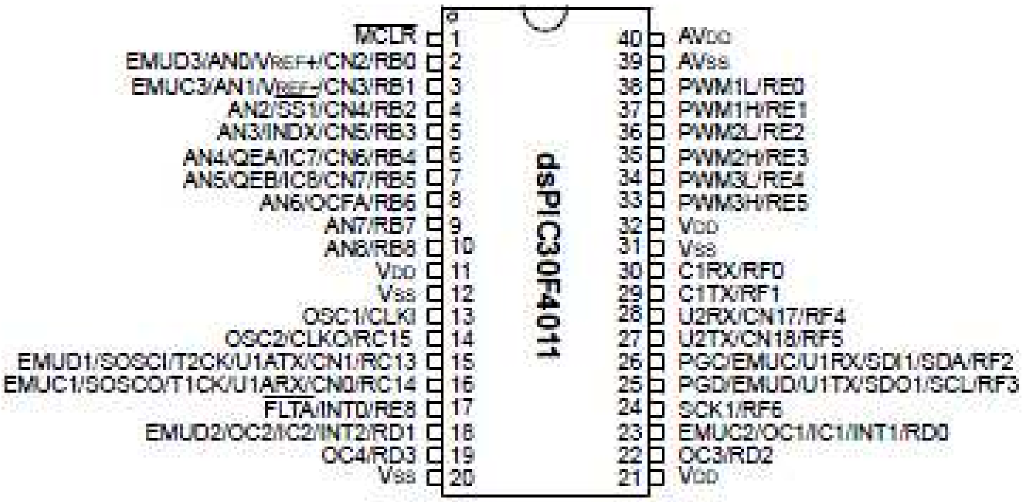
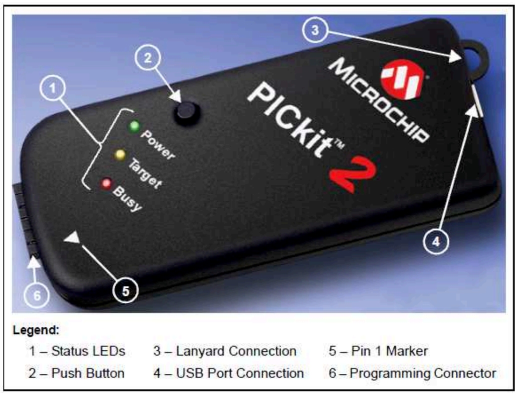


Figura B.7

8. Diagrama de pines del quemador



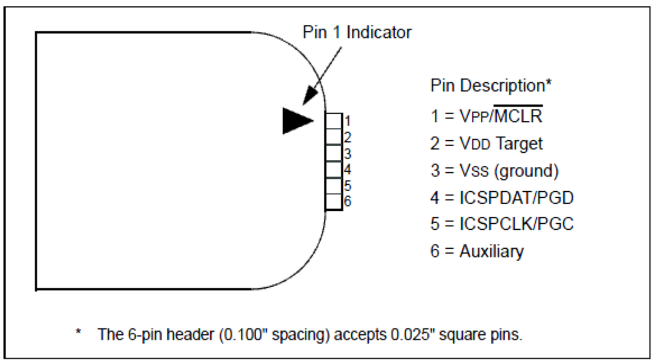


Figura B.8

ANEXO C

ESQUEMATICO DE COMUNICACION SERIAL

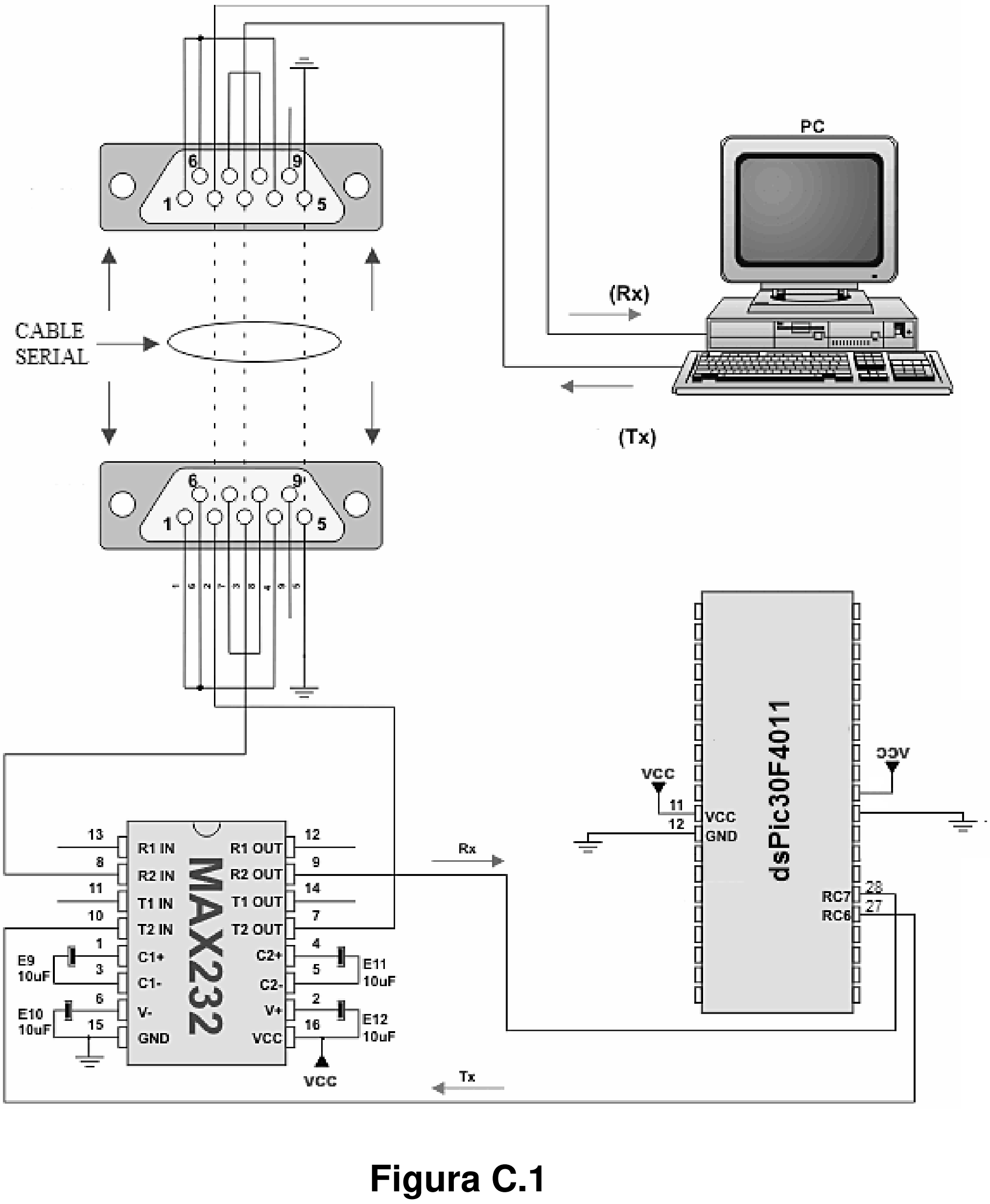


Figura C.1

DIAGRAMA DE BLOQUES

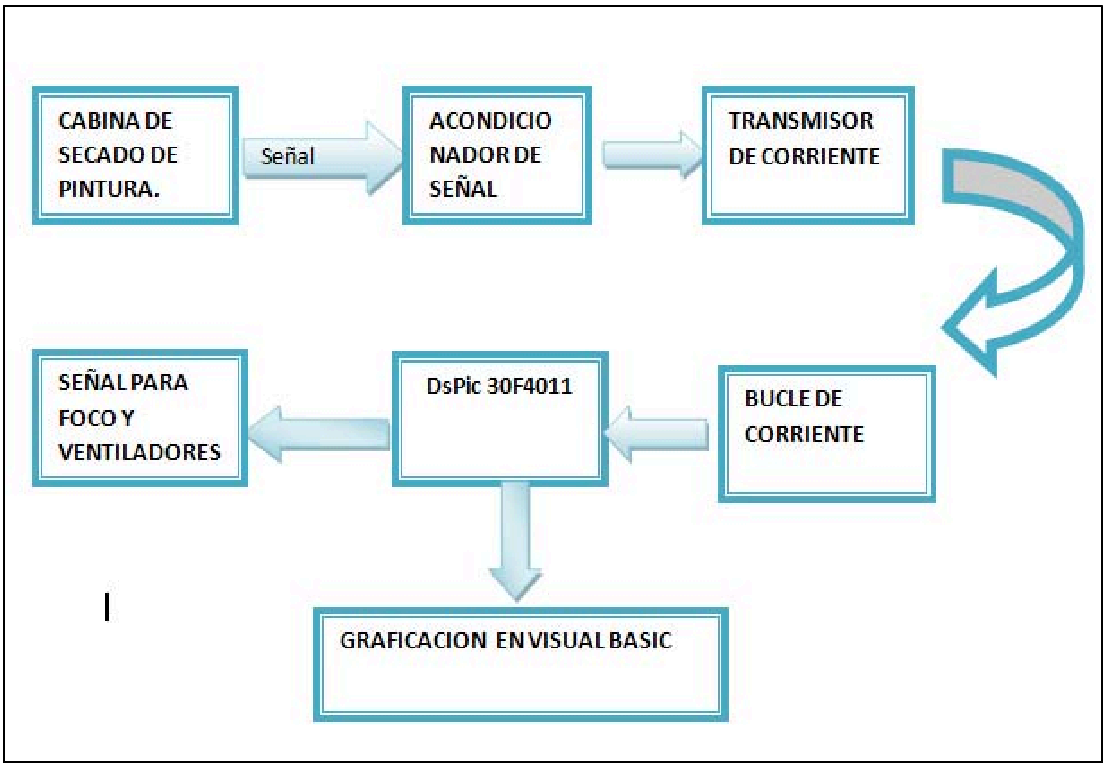


Figura C.2

ANEXO D

TABLA DE RUBROS PARCIALES Y TOTALES

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	$ 1.80	$ 5.40
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			**$ 13.00**

Acondicionador			
Descripción	**Cantidad**	**Precio Unitario**	**Precio Total**
Sensor LM35	1	$ 4.00	$ 4.00
Amplificadores Operacionales LF353	1	$ 1.80	$ 1.80
Resistencia de 1K -1/4 W	4	$ 0.05	$ 0.20
Trimmer Multivueltas de 10 K - 1/4W	1	$ 0.07	$ 0.07
Circuito Impreso	1	$ 3.00	$ 3.00
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			$ 9.82

Cabina de Secado			
Descripción	**Cantidad**	**Precio Unitario**	**Precio Total**
Acrílico 25*25 cm	5	$ 1.75	$ 8.75
Ventiladores (VN 4051) 12 VDC	2	$ 5.00	$ 10.00
Foco de 100W	1	$ 1.00	$ 1.00
Silicón	2	$ 0.10	$ 0.20
Madera 50*30 cm	1	$ 4.00	$ 4.00
Total			**$ 23.95**

Transmisor de Corriente			
Descripción	**Cantidad**	**Precio Unitario**	**Precio Total**
Componentes Electrónicos			
Amplificadores Operacionales LF353	3	$ 1.80	$ 5.40
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			**$ 12.55**

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**

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**

ANEXO E

TABLAS

1. Tabla de relación de temperatura con respecto al voltaje

Temperatura(ºC)	Vsalida (mV)
20	230
21	241
22	255
23	258
24	268
25	280
26	294
27	300
28	307
29	318
30	326
31	338
32	349
33	362
34	370
35	382

ANEXO F

Manual de Usuario

1. Conectar la cabina a 110 Voltios 60 Hz.

2. Conectar los ventiladores, tarjetas a una fuente de +/- 12 Voltios y el controlador a una fuente de 5 voltios.

3. Inmediatamente usted podrá visualizar en la Lcd un mensaje que dice “Control de temperatura” para salir de este menú presione la tecla numeral del teclado que forma parte del sistema.

4. A continuación se observara en la Lcd un menú que corresponde a una serie de productos en nuestro caso pinturas, se debe escoger el tipo de pintura con el cual esta pintado el vehículo o pieza que ingresa a la cabina.

5. Se debe conectar el controlador a la PC por medio de un cable USB para poder adquirir los datos temperatura y poderlos graficar posteriormente.

6. Una vez escogida la opción se mostrara en la Lcd la temperatura seteada y el valor actual de temperatura a la que se encuentra la cabina.

7. El foco con el cual se controla la temperatura en la cabina permanecerá encendido hasta que la cabina llegue a su temperatura seteada, una vez ocurrido esto se procederá a extraer el aire caliente en la cabina y a enfriarla por medio de un impulsor de aire, el mismo que se apagara cuando la cabina llegue a una temperatura estable.

8. Cuando la cabina llegue a una temperatura estable se muestra en la Lcd un mensaje que dice “Proceso terminado”, se debe presionar la tecla numeral para retornar al menú principal.

ANEXO G

GRAFICAS DE RESULTADOS OBTENIDOS EN VISUAL BASIC

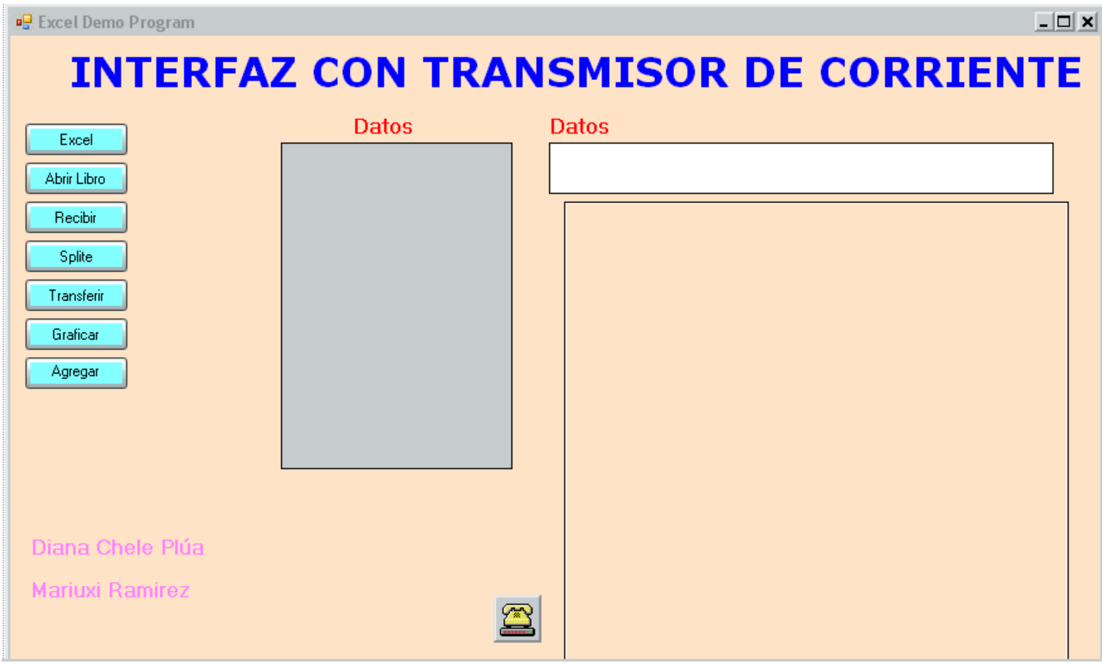


Figura G.1

BIBLIOGRAFIA

1. 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.

2. Nacional Semiconductor, Manual del Sensor LM35: Precisión Temperatura Sensors, (Edición mayo 1999) .

3. Página:
http://www.monografias.com/trabajos12/microco/
microco.shtml, Titulo: Micro-controladores.

4. Página:
www.datasheetcatalog.net/es/datasheets_pdf/L/M/3/5/LM35.shtml

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