



**UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO**

FACULTAD DE ESTUDIOS SUPERIORES ARAGÓN

**“DISEÑO Y DESARROLLO DE UN SISTEMA DE CONTROL Y
MONITOREO PARA TANQUES DE MADURACIÓN DE GELATINA
CON MICROCONTROLADORES PIC”**

T E S I S

**QUE PARA OBTENER EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN
P R E S E N T A :
RAFAEL GONZÁLEZ BETANCOURT**



ASESOR DE TESIS: M. EN I. ARCELIA BERNAL DÍAZ

MÉXICO 2011



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

JURADO ASIGNADO:

Presidente: ING. ENRIQUE GARCÍA GUZMÁN

Secretario: ING. OSCAR ESTRADA GARCÍA

Vocal: M. EN I. ARCELIA BERNAL DÍAZ

1er. Suplente: M. EN C. JESÚS HERNÁNDEZ CABRERA

2do. Suplente: ING. IGNACIO MENDOZA NUCAMENDI

ASESOR DE TESIS:

M. EN I. ARCELIA BERNAL DÍAZ

FIRMA

Dedicatoria

A mi esposa Nora, por todo su tiempo, apoyo y tolerancia, porque me ha enseñado a perseverar y a no rendirme, por estar siempre a mi lado, por dictarme, pero sobre todo por su infinito amor.

A mis hijos Nora Abigail y Eduardo, porque me permitieron robarles mucho del tiempo en el que debí estar con ellos, porque en todo momento me brindaron su apoyo y su amor, porque son el motor que me mueve.

A mi madre Hortensia Carmen, por su amor y su paciencia.

A mi hermano Gaspar, por ser mi mejor amigo, y simplemente por estar a mi lado en las buenas y en las malas.

A mis tíos, Salvador, Mauro, Gaspar, Manuel y Santiago, por ser los padres que no tuve, por formarme por el camino del bien y la honradez, porque siempre predicaron con su ejemplo, porque les debo mucho.

A mis tías, Sol, Socorruto, Mimi, Marcela y Luisi, por todo su cariño y enseñanzas, por permitirme ser parte de su familia.

Agradecimientos

A Dios por permitirme ver realizado este sueño y por todo lo que me ha brindado.

A mi asesor de tesis M. en I. Arcelia Bernal Díaz, por su guía a lo largo de este proyecto, por compartir sus conocimientos conmigo, por siempre darme ánimos.

Al Ing. Ignacio Mendoza Nucamendi, por todo el apoyo y orientación que me brindó desde que fue mi profesor y en este proyecto.

Al Ing. Oscar Estrada García, por el tiempo y dedicación que amablemente tuvo hacia mi persona.

A mis sinodales:

Ing. Enrique García Guzmán

Ing. Oscar Estrada García.

M. en I Arcelia Bernal Díaz

M. en C. Jesús Hernández Cabrera

Ing. Ignacio Mendoza Nucamendi

A la Universidad Nacional Autónoma de México, por permitirme ser parte de ella, por brindarme todos los conocimientos que me han hecho ser un orgulloso universitario.

A todos mis profesores de la FES Aragón, por su tiempo, y conocimientos.

INTRODUCCIÓN	1
CAPÍTULO 1 ANTECEDENTES.	
1.1 <i>Gelcaps Exportadora de México, S.A. de C.V. Orígenes.</i>	1
1.2 <i>Fabricación de cápsulas de gelatina blanda</i>	3
1.3 <i>¿Qué es la gelatina blanda?</i>	4
1.4 <i>¿Cómo se fabrica la gelatina blanda?</i>	5
1.5 <i>Proceso de maduración de la gelatina</i>	6
CAPÍTULO 2 TANQUES DE MADURACIÓN.	
2.1 <i>Tanque de maduración de gelatina</i>	7
2.2 <i>Componentes eléctricos del tanque</i>	8
2.2.1 <i>Resistencia de inmersión</i>	9
2.2.2 <i>Relevador de estado sólido</i>	10
2.3 <i>Componentes electrónicos del tanque</i>	10
2.3.1 <i>Control de temperatura.</i>	11
2.3.2 <i>Sensor PT-100</i>	12
2.4 <i>Funcionamiento del tanque de maduración</i>	12
2.5 <i>Conclusión capítulo 2.</i>	16
CAPÍTULO 3 LOS MICROCONTROLADORES PIC.	
3.1 <i>Arquitectura general de los microcontroladores PIC</i>	20
3.1.1 <i>Arquitectura Von Neumann</i>	21
3.1.2 <i>Arquitectura Harvard</i>	22
3.1.3 <i>Procesador Segmentado</i>	24
3.1.4 <i>Procesador RISC</i>	26
3.1.5 <i>Arquitectura Ortogonal</i>	27
3.1.6 <i>Arquitectura basada en un banco de registros</i>	28
3.2 <i>Características Especiales del Procesador</i>	28
3.2.1 <i>Alimentación Eléctrica</i>	28
3.2.2 <i>El oscilador Principal</i>	29
3.2.3 <i>Reinicio (Reset)</i>	32
3.2.4 <i>Memoria de Usuario</i>	33
3.2.5 <i>Interrupciones</i>	34
3.2.6 <i>Programación Serie en Circuito</i>	34

3.3. Tipos de Encapsulados	35
3.3.1. DIP (Dual Inline Package)	35
3.3.2. SOIC (Small Out-Line Integrated Circuit)	35
3.3.3. SSOP (Shrink Small Out-Line Package)	35
3.3.4. PLCC (Plastic Leaded Chip Carrier)	36
3.3.5. TQFP (Thin Quad Flatpack Package)	36
3.3.6. QFN (Quad Flatpack Package No Leads)	36
3.4. Los microcontroladores PIC están agrupados en familias	36
3.4.1. Familia Enana y Baja	38
3.4.2. Familia Media	39
3.4.3. Familia Alta	39
3.4.4. Familia Mejorada	40
3.5. Sistema de identificación del microcontrolador	40
3.6. Conclusión capítulo 3.	42

CAPÍTULO 4 COMUNICACIÓN.

4.1 Estándares de comunicación.	43
4.2 Comunicación Alámbrica.	44
4.2.1 RS232	44
4.2.2 USB	50
4.2.3 Modem.	58
4.3 Comunicación Inalámbrica	59
4.3.1 Radio Modem.	59
4.3.2 IrDA	60
4.3.3 Blue Tooth	61
4.3.4 WI-FI	63
4.4 Conclusión capítulo 4.	68

CAPÍTULO 5 DISEÑO Y DESARROLLO DE LA INTERFAZ DE SOFTWARE DEL SISTEMA DE CONTROL Y MONITOREO (SCM).

5.1 Interfaz	69
5.1.1 Interfaz gráfica de usuario (GUI).	69
5.1.2 Interfaz Hardware.	70
5.2 Programa seleccionado	70

5.2.1	C# (“C Sharp”).	71
5.2.2	Instalando Microsoft Visual C# Express Edition.	71
5.3	Configurando el proyecto	72
5.4	Interfaz gráfica visual.	74
5.5	Desarrollo de la Interfaz gráfica para el control de temperatura.	76

CAPÍTULO 6 DISEÑO Y DESARROLLO DE LA INTERFAZ EN HARDWARE DEL SISTEMA DE CONTROL Y MONITOREO (SCM).

6.1	Componentes del circuito.	79
6.1.1	MAX232.	79
6.1.2	PIC18F2550	80
6.2	Armado del circuito	81
6.2.1	Alambrado puerto serial.	81
6.2.2	Alambrado puerto USB.	82
6.2.3	Alambrado MAX232	83
6.2.4	Alambrado PIC18F2550.	84
6.2.5	Circuito armado.	84
6.3	Programando el PIC18F2550	85
6.3.1	MPLAB-IDE V8.63 (Ambiente de Desarrollo Integrado).	85
6.3.2	Proteus 7.4	88
6.3.2.1	Simulación con VSM (Virtual System Modelling, Modelado de Sistema Virtual)	90
6.3.3	Grabando el PIC18F2550	93
6.4	Prueba de la interfaz en hardware (Circuito alambrado)	96
6.4.1	Conexión de componentes	97
6.5	Conclusión capítulo 6	98

CONCLUSIONES	100
---------------------	------------

BIBLIOGRAFÍA	103
---------------------	------------

MESOGRAFÍA	103
-------------------	------------

INTRODUCCIÓN

Gelcaps Exportadora de México, S.A. de C.V., es una empresa dedicada a la elaboración de fármacos para el cuidado de la salud, con 40 años de experiencia en la fabricación de cápsulas de gelatina blanda, lo que la ha hecho una de las empresas líder en esta forma de medicamento.

En cada uno de los procesos que constituyen la fabricación de cápsulas de gelatina blanda, la empresa ha ido innovando y adaptando la tecnología más reciente para hacer más eficientes sus procesos, sin embargo al adaptar esta tecnología a procesos ya establecidos, provoca que no siempre se obtengan los mejores resultados o que ésta no se adapte al 100%, como sería lo ideal, es el caso que se presenta en el proceso de maduración de la gelatina previo al encapsulado, proceso en el que el control se efectúa de forma manual por parte de los operadores, esto debido que aunque los equipos cuentan con los instrumentos necesarios para llevar a cabo un control automatizado, éstos no se adaptaron de tal forma que se pueda hacer así, lo que lo hace un proceso eficaz pero no eficiente.

Con base a lo ya mencionado, la presente tesis comprende el desarrollo de un prototipo de interfaz de hardware y software, que logre la automatización de esta parte del proceso y por ende un control total y eficiente, utilizando los mismos instrumentos con que cuentan los equipos para la maduración de gelatina. Este prototipo consistirá principalmente de una interfaz de hardware, en la que en la parte electrónica emplearemos un microcontrolador PIC de microchip y un circuito MAX232, además como medios de comunicación implementaremos RS232 y USB, también una interfaz gráfica de usuario desarrollada en lenguaje C#. Cabe mencionar que se tendrán como principales parámetros de desarrollo los siguientes aspectos: tamaño, costo, facilidad de operación, y el uso de la tecnología con que ya cuentan los equipos a controlar.

Este trabajo escrito corresponde al prototipo de la interfaz de hardware o sistema de comunicación entre una computadora y el control de temperatura, así como de su interfaz gráfica y está estructurado de la siguiente forma:

Capítulo 1, proporciona una introducción de la empresa Gelcaps Exportadora de México, S.A. de C.V., dando sus antecedentes, así como la forma farmacéutica que elabora (cápsula de gelatina blanda) y el proceso para fabricar este tipo de medicamentos.

Capítulo 2, este capítulo se enfoca en el proceso de la maduración y calentamiento de la gelatina, previa al encapsulado, así como de los componentes con que se lleva a cabo, haciendo una descripción de cada uno de ellos, con la finalidad de entender el proceso y el trabajo que realiza cada uno de los componentes.

Capítulo 3, contiene los conceptos básicos de los microcontroladores PIC, así como de su arquitectura, sus características especiales, los tipos de encapsulado con que se presentan y sus diferentes familias.

Capítulo 4, aquí se mencionan los diferentes estándares de comunicación y cómo funcionan éstos, dividiéndolos en comunicación por cable y comunicación inalámbrica, enfocándonos más a detalle en la comunicación que se implementará en este trabajo, USB y RS232.

Capítulo 5, este capítulo contiene lo referente a la interfaz de software, o interfaz de usuario, explicando que es una interfaz, contiene también información del lenguaje de programación C#, que fue el que se seleccionó para realizar la interfaz, explicando porque se seleccionó, como se instaló, además de cómo se elaboró la interfaz y como funciona.

Capítulo 6, aquí se menciona todo lo referente a la electrónica del circuito, explicando cada uno de los componentes que lo forman, así como de las herramientas software y hardware necesarias para programar el PIC y hacer la simulación del circuito, con la finalidad de corroborar su funcionamiento, además de las pruebas de funcionamiento.

Conclusiones, en esta sección vierto mi opinión acerca del trabajo realizado, así como del planteamiento del mismo y como se fue desarrollando hasta ver terminando el presente trabajo.

Bibliografía y mesografía, en esta sección se enlistan los diferentes libros y páginas de internet consultados para el desarrollo de este trabajo.

CAPÍTULO 1 ANTECEDENTES.

1.1 *Gelcaps Exportadora de México, S.A. de C.V. Orígenes.*

Gelcaps Exportadora de México, S.A. de C.V., es una empresa farmacéutica que fue fundada en 1971. Gelcaps es pionera en la forma farmacéutica de cápsulas de liberación o gelatina blanda, en esta primera etapa sólo se dedicó a maquilar cápsulas de gelatina blanda para otras empresas, en 1976 se crea la línea de productos marca **gelcaps**, que en un inicio fueron: vitamina E, lecitina de soya y ajo, con lo que se ingresó al mercado naturista. Para 1984 se da el crecimiento de la línea con nuevos productos, con una estrategia de mercadotecnia formal, lo que la hace una empresa aún más competitiva.

Para 1991 **Sobel**, corporativo holandés, adquiere **Pharmacaps Inc.**, empresa estadounidense matriz de “Gelcaps Exportadora de México”, en 1992 **Sobel** adquiere **Banner**, principal competidor de **Pharmacaps, Inc.** con plantas en Canadá y en Estados Unidos, una vez que se hace oficial la fusión de estas empresas, se les da el nombre de **Banner Pharmacaps**.

En el 2005 **Gelcaps Exportadora de México** orgullosamente se consolida como parte de la red mundial de los expertos en formas farmacéuticas de liberación.

Para el 2009 Gelcaps introduce 2 nuevas tecnologías, mastigels, una innovadora forma farmacéutica masticable especialmente adecuada en poblaciones pediátricas o geriátricas y entericare, cápsulas entéricas transparentes diseñado para compuestos farmacéuticos sensibles a ácidos o que tiendan a causar malestar gástrico, también es ideal para compuestos que deban actuar en el intestino delgado.

Con más de 40 años en el mercado. Gelcaps Exportadora de México se ha consolidado como líder indiscutible en México, en la fabricación de cápsulas de gelatina blanda herméticamente sellada, también conocida como Softgels. La experiencia adquirida en este período nos ha permitido incursionar en otros mercados tales como Europa, Canadá, América Latina y el Caribe, para ofrecer

diversos medicamentos y multivitamínicos en una variedad de formas de cápsulas de gelatina blanda, (*figura 1.1*)



Fig. 1.1 Diferentes tamaños y formas de cápsulas de gelatina blanda

A lo largo de los años la tecnología utilizada por **gelcaps** en la creación de las cápsulas de gelatina blanda ha ido avanzando, incorporando en cada uno de sus procesos los elementos electrónicos, eléctricos y mecánicos disponibles en el mercado, con la finalidad de automatizar sus procesos y ser una empresa eficiente y competitiva.

Desde principios de los 80's una de las herramientas que se incorporó a los procesos químicos y bioquímicos realizados en los departamentos de Control Químico e Investigación y Desarrollo, fue la computadora, utilizando programas propios para realizar análisis químicos y microbiológicos. De igual forma en las áreas de producción se comenzaron a introducir instrumentos de control como los PLC (Control Lógico Programable), controles de temperatura, sensores, *touchscreen*, relevadores, etc., sin embargo en el proceso de maduración de la gelatina blanda se necesita llevar un control y monitoreo más exacto, porque aunque el dispositivo utilizado para este fin cuenta con algunos componentes que ayudan a controlar el proceso, no está automatizado del todo, pues sigue siendo un proceso manual. Es en este punto donde se desarrolla un sistema auxiliándonos de los microcontroladores PIC, para controlar y monitorear la temperatura de los tanques de maduración de gelatina, así como el tiempo que requiere cada tipo de gelatina para madurar, dándonos el control sobre esta parte del proceso.

1.2 Fabricación de cápsulas de gelatina blanda.

El proceso de fabricación de las cápsulas de gelatina blanda comienza con la selección de los materiales necesarios para elaborar la gelatina y la formulación o medicamento que se va a encapsular, una vez que se han elaborado se llevan en sus respectivos tanques al área de mezanine ¹ donde por medio de gravedad se harán bajar hacia la máquina encapsuladora o máquina Shoc (figura 1.2), la cual utiliza el proceso de encapsulado rotatorio.

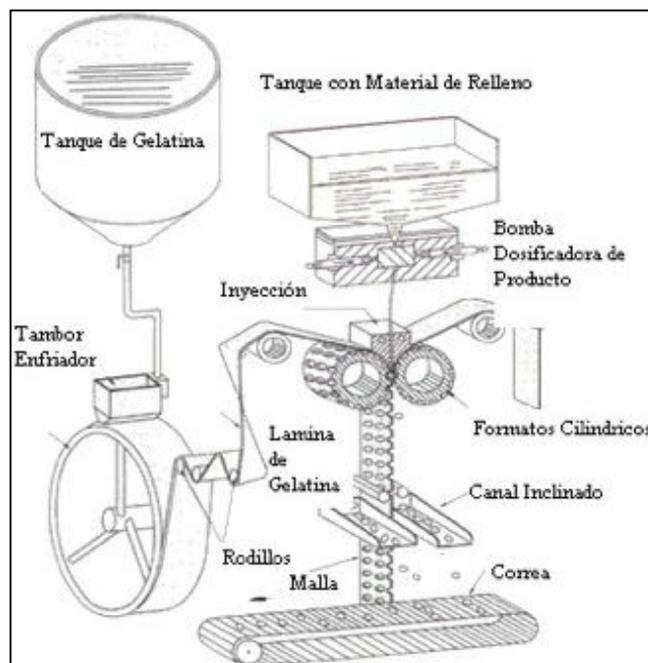


Fig. 1.2 Esquema general máquina encapsuladora

La gelatina se baja hacia los vertedores (contenedores de bronce), que dan forma a la película o lamina de gelatina blanda, es decir aquí se le da el ancho y el espesor a la película de gelatina vertiéndola sobre la superficie de un tambor de acero inoxidable, que se encarga de enfriar y transportar la película de gelatina hacia unos rodillos tensores que hacen que llegue al área de los formatos cilíndricos (moldes) y el segmento de inyección del medicamento, en esta sección de la máquina el molde le da forma a la gelatina, encargándose de cada mitad de la cápsula y el segmento inyecta la cantidad exacta de medicamento dentro de las 2 láminas de gelatina que forman la cápsula, la presión de los moldes hace que sellen y corten la cápsula de las láminas que la forman desprendiéndose y dejando solo una red o malla. La cápsula recién formada es

1.- Área destinada para el calentamiento de los tanques de maduración

transportada a los secadores rotatorios, donde se les elimina el exceso de aceite usado en la lubricación que se le agrega a las películas o láminas de gelatina para que no se peguen en los tambores, una vez que se les quito el exceso de aceite se transportan a los túneles de secado donde por medio de control de aire y humedad se deshidratan para que tomen su consistencia idónea llevando la cápsula a una humedad final del 4% al 10% y puedan ser consumidas por el público.

1.3 ¿Qué es la gelatina blanda?

La gelatina blanda es un compuesto formado por: gelatina, plastificante y agua, a continuación se explica cada uno de ellos.

Gelatina es una mezcla heterogénea de proteínas solubles en agua de alto peso molecular, extraídas de diferentes fuentes de colágeno como; huesos y pieles de animales como bovinos, cerdos, pescado e incluso de origen sintético, generalmente su presentación es seca en gránulos sólidos, listos para ser disueltos en agua.

Plastificante (glicerina) es un líquido viscoso totalmente inodoro, faltar de color, transparente de sabor dulce, soluble en agua en cualquier proporción, tiene un punto de ebullición de 290 °C y un punto de fusión de 18 °C.

Agua, potable o purificada la cual sirve como solvente para hacer moldeable la masa de gelatina.

También puede contener conservadores, colorantes, opacificantes aromatizantes y saborizantes, posiblemente azúcar para impartir características masticables a la cubierta.

1.4 ¿Cómo se fabrica la gelatina blanda?

En un homogeneizador o reactor (*figura 1.3*) se mezcla la gelatina seca granulada con agua a 60°C, para hidratarla, se agrega la glicerina (plastificante) y se somete a agitación constante, se le aplica vacío para retirar el aire acumulado hasta que la mezcla sea homogénea, una vez obtenida la mezcla la cual es totalmente transparente, se vacía y filtra a través de la válvula que tiene el reactor en su parte inferior (*figura 1.4*), vaciando el contenido en los tanques de maduración, los cuales son llevados al área de agitadores, cabe mencionar que el contenido de un reactor puede llenar hasta 5 tanques de maduración de 200 litros cada uno.

Los agitadores cuentan con una propela² (*figura 1.6*) de acero inoxidable la cual se introduce en el tanque para que agite la gelatina (*figura 1.5*) y en ella se puedan disolver otros componentes como colores, saborizantes, aromatizantes o el opacificante para darle un color mate a la gelatina



Figura 1.3 Homogeneizador con escotilla abierta



Figura 1.4 Parte inferior del homogeneizador



Figura 1.5 Gelatina en agitación y coloración



Figura 1.6 Agitador con propela de acero inoxidable

2.- Agitador de hélice de acero inoxidable, se emplea para agitar líquidos viscosos

1.5 *Proceso de maduración de la gelatina*

Una vez que se ha obtenido el tipo de gelatina que se va a usar para encapsular un determinado medicamento, es decir que tiene el color y los componentes necesarios que se agregaron durante la agitación, el tanque de maduración es transportado al área de mezanine donde es conectado a una toma de corriente de 220 V. (figura 1.7), la cual alimenta una resistencia (resistencia de inmersión), que es la encargada de mantener caliente el tanque a una temperatura de 60°C de esta forma se mantiene la gelatina a la temperatura deseada, cuando la gelatina a madurado el tiempo indicado, el operador desconecta la resistencia y traslada el tanque hacia el área del mezanine donde la gelatina es bajada por gravedad hacia la máquina encapsuladoras, en esta parte del mezanine se vuelve a conectar el tanque a la corriente eléctrica para que mantenga su temperatura de 60°C \pm 5°C en lo que se vacía el tanque, lo que dependiendo del medicamento que se esté encapsulando puede tardar de 5 a 6 horas en promedio por tanque, cabe mencionar que la maduración de una gelatina, dependiendo del medicamento a encapsular y del tipo de gelatina, puede tener un tiempo de maduración que va desde las 5 horas como mínimo, hasta 30 horas.



Figura 1.7 Tanque en proceso de maduración de gelatina

CAPÍTULO 2 TANQUES DE MADURACIÓN.

2.1 Tanque de maduración de gelatina

El tanque de maduración de gelatina, también llamado tanque enchaquetado, es un contenedor de forma cilíndrica (*figuras 2.1, 2.2 y 2.3*), fabricado en acero Inoxidable grado 316, lo que le da una gran resistencia a la oxidación y/o corrosión, esta característica lo hace ideal para la industria alimenticia y farmacéutica, se le conoce también como tanque enchaquetado por presentar un doble fondo, es decir que al tanque lo recubre otra pared de acero inoxidable a modo de chaqueta, estas 2 paredes, están separadas una de la otra por una distancia de 1.5 pulgadas, es en este hueco que se forma entre las paredes, en donde se agrega agua, la cual se calienta utilizando una resistencia de inmersión (*ver figura 2.4*), para a su vez calentar la gelatina en una especie de baño maría.³ En la parte baja del tanque en su cara externa, se encuentra una abertura cilíndrica con rosca para instalar la resistencia, que se encargará de calentar el agua, en la parte alta y colocada a un costado en la pared externa se encuentra un pivote (tubo) en forma de L, también de acero inoxidable, por donde se agrega o extrae el agua. En un costado del tanque se encuentra la caja de control donde se alojan los instrumentos que se encargan de controlar la temperatura, también cuenta con una válvula de bola de 1.5" de diámetro, fabricada en acero inoxidable, que es por donde se descarga la gelatina hacia las máquinas encapsuladoras

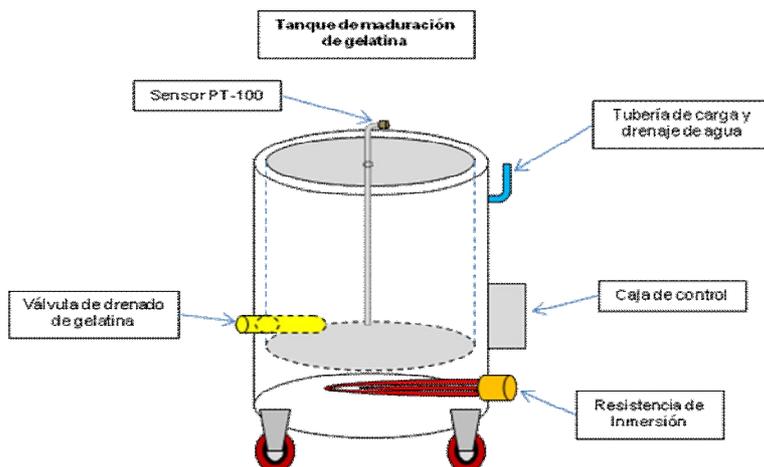


Figura 2.1 Esquema de los componentes del tanque

3.- Método empleado en las industrias (farmacéutica, cosmética, y de alimentos) y en laboratorio de química, para conferir temperatura uniforme a una sustancia líquida o sólida o para calentarla lentamente

El tanque cuenta con una tapa redonda en acero inoxidable, con agarraderas y un orificio diseñado para introducir el sensor de temperatura, la tapa del tanque embona a presión.



Figura 2.2 Frente del tanque

Figura 2.3 Reverso del tanque

2.2 Componentes eléctricos del tanque

La caja de control del tanque de maduración de gelatina cuenta con una serie de componentes eléctricos que trabajan con un voltaje de 220 V., estos son la resistencia de inmersión (figura 2.4), un relevador de estado sólido un apagador de un tiro y un fusible tipo americano de 220V 10 Amps. (Figura 2.5)



Figura 2.4 Resistencia de inmersión

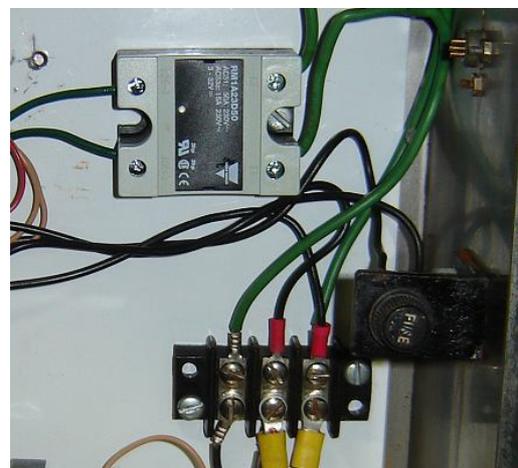


Figura 2.5 Relevador, apagador y fusible

2.2.1 Resistencia de inmersión

Las resistencias de inmersión convierten energía eléctrica en calor. Procedimiento descubierto por James Prescott Joule cuando en 1841 al hacer circular corriente eléctrica a través de un conductor se liberó calor por encontrar resistencia. La resistencia está diseñada para el calentamiento en contacto directo de un fluido por lo general agua, que será la encargada de calentar cualquier líquido o sólido contenido en un tanque o recipiente, que a su vez es calentado por ésta agua, la resistencia nunca está en contacto con el contenido del tanque.

En la actualidad las resistencias de inmersión se utilizan para infinidad de aplicaciones, por ejemplo en las industrias (farmacéutica, cosmética, de alimentos y conservas) y en laboratorios de químicos entre otros.

La gran mayoría de las resistencias son fabricadas con un alambre de una aleación de níquel (80%) y cromo (20%). Esta aleación soporta temperaturas muy altas (1000°C), es resistivo (condición necesaria para generar calor), es muy resistente a los impactos y es inoxidable.

La resistencia eléctrica de inmersión es fabricada en acero inoxidable 316, con una baja densidad de carga para evitar la calcificación prematura, por las sales minerales contenidas en el agua.

Las resistencias de inmersión ofrecen buenas características térmicas, eléctricas y mecánicas. Todos los materiales utilizados ofrecen máxima protección contra la corrosión.

Este formato de resistencia incorpora un tapón roscado NPT de 1 ½" que le permite fijar de forma adecuada y segura la resistencia al dispositivo a calentar.

Las características Técnicas de esta resistencia son:

- Tapón roscado 1 1/2" en latón
- Elementos de acero inoxidable 316
- Potencia: 3000 watts.
- Longitud: 50 cm (Sin incluir el tapón)
- Alimentación: 220 V

2.2.2 Relevador de estado sólido

Se llama relevador de estado sólido (*figura 2.6*), a un circuito híbrido, normalmente compuesto por un optoacoplador que aísla la entrada, un circuito de disparo, que detecta el paso por cero de la corriente de línea y un triac o dispositivo similar que actúa de interruptor de potencia. Su nombre se debe a la similitud que presenta con un relevador electromecánico; este dispositivo es usado generalmente para aplicaciones donde se presenta un uso continuo de los contactos del relevador que en comparación con un relevador convencional generaría un serio desgaste mecánico, además de poder conmutar altos amperajes que en el caso del relevador electromecánico destruirían en poco tiempo los contactos. Estos relevadores permiten una velocidad de conmutación muy superior a la de los relevadores electromecánicos, la función del relevador de estado sólido en este arreglo es la de apagar y encender la resistencia de inmersión cuando reciba la señal del control de temperatura



Figura 2.6 Relevador de estado sólido

2.3 Componentes electrónicos del tanque

Los componentes electrónicos que complementan la caja de control de temperatura son un control de temperatura y un sensor PT-100

2.3.1 Control de temperatura.

El control de temperatura (*figura 2.7*) es un dispositivo electrónico mediante el cual se puede regular la temperatura de algún sistema físico tal como una pieza de algún material o un recinto. Los controles de temperatura se pueden ubicar en dos grandes grupos de sistemas, sistemas de lazo abierto y sistema de lazo cerrado



Figura 2.7 Control de temperatura FUJI

Sistema de lazo abierto

En un sistema de lazo abierto, se entrega una cierta cantidad de energía constante o variable para lograr una temperatura prefijada, o una variación de temperatura. Cualquier variación de las condiciones del elemento a controlar, no será corregida, por no disponer de una retroalimentación directa de la temperatura a controlar.

Sistema de lazo cerrado.

En el sistema de lazo cerrado, se entrega una cierta cantidad de energía que será dependiente de la diferencia de temperatura real del elemento a controlar y de la temperatura prefijada o sea que existe una realimentación.

Una forma segura y sencilla de lograr la temperatura deseada sobre el elemento, es aplicar un **sensor** sobre éste y con los datos obtenidos, retroalimentar el sistema a fin de aplicar la energía calórica necesaria para lograr el resultado requerido.

2.3.2 Sensor PT-100

El sensor PT-100 (*figura 2.8*), es un sensor de temperatura que basa su funcionamiento en la variación de resistencia a cambios de temperatura del medio. El elemento consiste en un arrollamiento muy fino de platino bobinado entre capas de material aislante y protegido por un revestimiento cerámico. El sensor PT-100 es llamado así por estar armado con el elemento platino Pt y por tener una resistencia de 100 ohms para 0°C, podemos decir que el sensor PT-100 es simplemente una resistencia que varía linealmente su resistencia de acuerdo con la temperatura, son ideales en procesos de producción de alimentos, medición ambiental, etc.



Figura 2.8 Sensor PT-100

2.4 Funcionamiento del tanque de maduración

Una vez que hemos conocido los componentes eléctricos y electrónicos que conforman al tanque de maduración de gelatina, podemos analizar el funcionamiento en conjunto de éstos.

El objetivo del tanque de maduración es mantener la gelatina a 60°C durante el tiempo necesario para que ésta madure, lo cual debe ser en un intervalo que va de 6 horas mínimo a 30 horas como máximo, ya que antes de 6 horas la gelatina presenta problemas de falta de viscosidad adecuada, lo que impide un buen sellado al momento de formarse la cápsula, y mantenerla por más de 30 horas, hará que se degrade haciéndola extremadamente chiclosa, de igual forma si en el

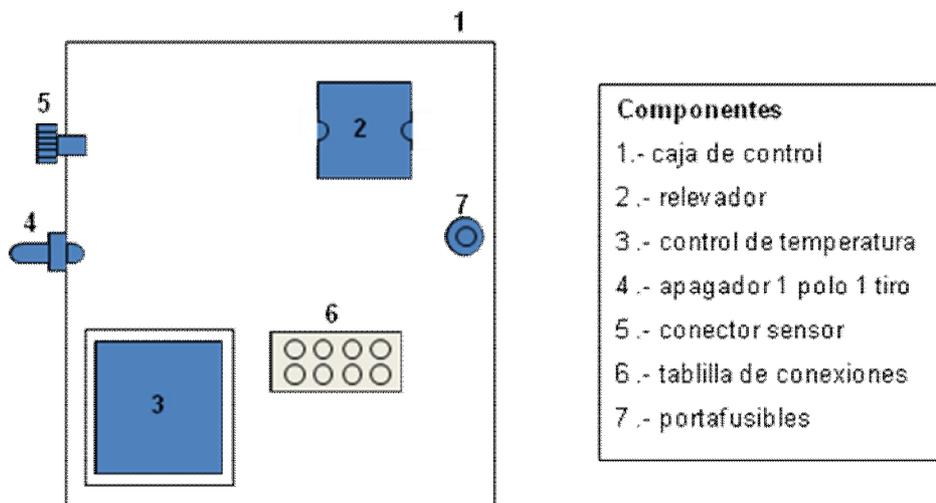
transcurso de la maduración la cápsula esta por fuera de los rangos de temperatura adecuados ($60^{\circ}\text{C} \pm 5^{\circ}\text{C}$), presentará los mismos problemas. He aquí la importancia de tener un buen control tanto en la temperatura como en el tiempo que se mantendrá en calentamiento la gelatina.

A continuación describiremos el circuito eléctrico con el que se controla la temperatura de los tanques (*figura 2.9*)



Figura 2.9 elementos de la caja de control

La caja de control está compuesta por los elementos mostrados en el siguiente cuadro:



Como ya sabemos el elemento encargado de calentar el tanque es la resistencia de inmersión, ésta es alimentada por 220 V y controlada por el relevador de

estado sólido como se muestra en el esquema (figura 2.10), también el control de temperatura es alimentado por este mismo voltaje.

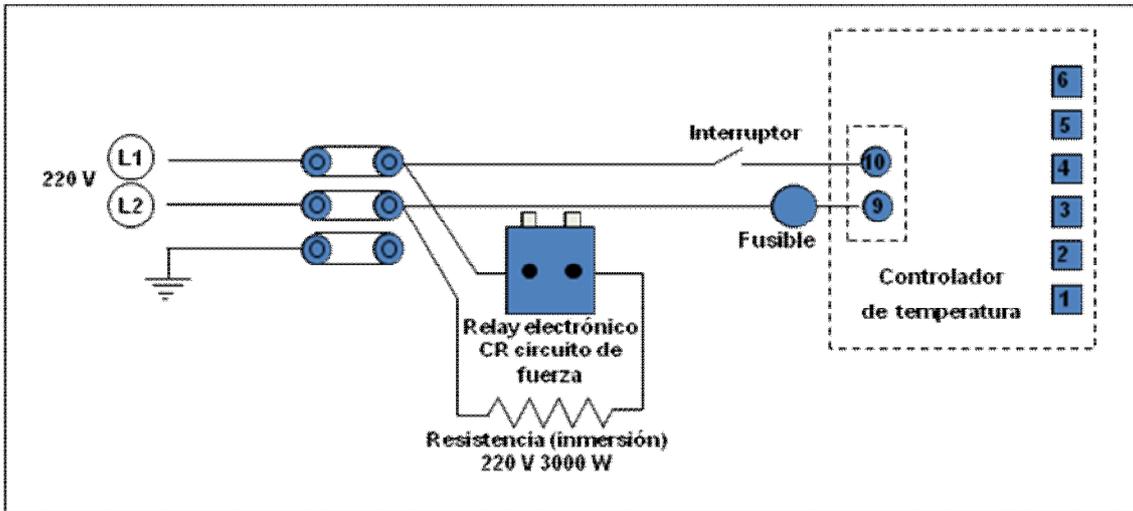


Figura 2.10 Esquema eléctrico

El sensor PT-100 es el encargado de tomar la temperatura de la gelatina y envía pulsos al control de temperatura indicándole la temperatura, si la temperatura no coincide con la temperatura con que se programó el control, este envía una señal al relevador de estado sólido para apagar o encender la resistencia de inmersión (figura 2.11), hasta que la temperatura de la gelatina coincida con la temperatura programada.

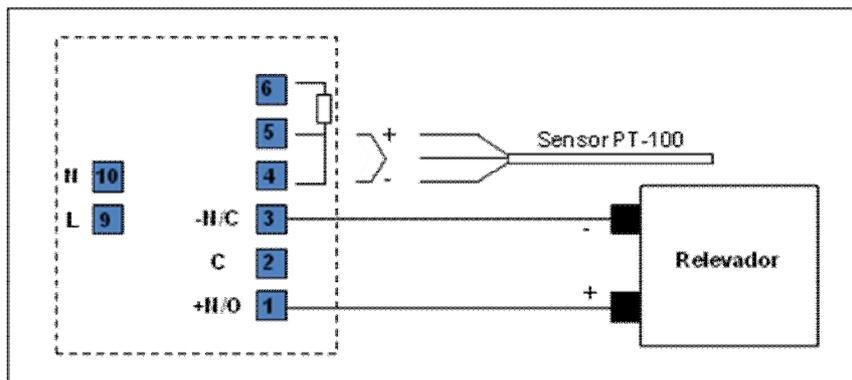


Figura 2.11 Esquema componentes electrónicos

El proceso que sigue el operador de los tanques de gelatina se describe a continuación.

Una vez que se ha elaborado la gelatina en los reactores, ésta se vacía a los tanque de maduración y se le agrega color, una vez hecho esto, se sube el tanque al área conocida como mezanine,⁴ en donde el tanque es conectado a una toma de corriente de 220 V., la cual alimenta la resistencia de inmersión, que es la encargada de calentar el agua de la chaqueta del tanque, el operador programa manualmente una temperatura de 60°C en el control de temperatura del tanque, un sensor de temperatura (PT-100) sumergido en la gelatina, envía constantemente su señal en forma de pulsos al control de temperatura, cuando la gelatina ha alcanzado la temperatura programada, el control de temperatura envía una señal a un relevador de estado sólido, el cual interrumpe la energía que alimenta la resistencia, la cual deja de calentar, en cuanto la gelatina disminuye su temperatura, el control envía la orden al relevador de estado sólido para que nuevamente encienda la resistencia, caliente y evite que la temperatura de la gelatina disminuya por debajo de la tolerancia marcada hacia abajo para 60°C la cual es de 5°C debajo de 60° es decir que nunca debe estar la temperatura debajo de 55°C, de esta forma se mantiene la gelatina a la temperatura deseada, cuando la gelatina a madurado el tiempo necesario, el operador desconecta la resistencia y traslada el tanque hacia el área de bajada de gelatina a máquinas encapsuladoras, en esta parte del mezanine se vuelve a conectar el tanque a la corriente eléctrica para que mantenga su temperatura de 60°C + 5°C durante el tiempo que tarda en vaciarse el tanque, lo que dependiendo del medicamento que se esté encapsulando puede tardar de 5 a 6 horas en promedio.

En esta área de calentamiento podemos tener hasta 17 tanques de maduración conectados simultáneamente a la corriente, incluyendo 5 que estarán en proceso de encapsulado. Como lo hemos comentado el tiempo de maduración de cada gelatina es diferente, si a esto agregamos que el operador no sólo vigila los tanques, sus temperaturas y tiempos, casi cada hora, sino que también tiene que preparar los reactores, para elaborar más gelatina, hacer los cálculos de la materia prima que va a usar según el programa de producción, preparar los colores,

4.- Mezanine es un piso intermedio donde se madura y descarga la gelatina hacia las máquinas encapsuladoras

colorear la gelatina, lavar los tanques que se han dejado de usar y elaborar reportes, todo esto durante 3 turnos 6 días de la semana, ya que el séptimo día el personal de guardia asiste únicamente para estar monitoreando y corrigiendo cualquier error que se pueda dar en los tanques de maduración o calentamiento, podemos deducir la ventaja que representa automatizar el proceso de maduración de gelatina, es decir que la tecnología haga esta parte del proceso, de esta forma los operadores de este departamento podrán hacer todas sus demás actividades, dejando el control y monitoreo de los tanques al supervisor del departamento de gelatinas, quien será el encargado de manejar la P.C. donde estará instalada la interfaz y la base de datos necesarias para junto con el circuito electrónico automatizar esta parte del proceso, con lo cual se obtendrá un ahorro de horas hombre, se evitarán errores humanos, se podrán monitorear y controlar de forma más fácil los tanques de maduración, además de obtener un historial confiable del trabajo de cada tanque.

2.5 Conclusión capítulo 2.

Lo comentado en estos dos primeros capítulos, es fundamental para entender el proceso de fabricación de las cápsulas de gelatina blanda, así como la parte del proceso que nos interesa, el calentamiento y maduración de la gelatina, para aplicar la tecnología de los microcontroladores y dar una solución al control de los tanques de maduración de gelatina, punto crítico y fundamental para el proceso de encapsulado.

CAPÍTULO 3 LOS MICROCONTROLADORES PIC.

Una **computadora digital** es un dispositivo electrónico capaz de realizar millones de cálculos por segundo para procesar grandes volúmenes de datos ejecutando una lista de instrucciones conocida como **programa**.

De manera general, para que un dispositivo pueda ser considerado una computadora debe contar con las siguientes **unidades funcionales**:

- **Unidad Central de Procesamiento** o CPU (*Central Process Unit*) por sus siglas en inglés, está encargada del control de los elementos de la computadora y de la ejecución del programa. El **microprocesador** es un ejemplo de CPU en las computadoras modernas.
- **Unidades de Almacenamiento**. También llamadas **dispositivos de memoria**, son usados para resguardar datos e instrucciones ya sea de manera permanente o temporal.
- **Unidades de Entrada y Salida**. En estos elementos son conectados **dispositivos periféricos** que permiten la comunicación con otros sistemas de cómputo o el usuario. Teclados, pantallas y bocinas son ejemplos de dispositivos periféricos que se conectan a las unidades de entrada y salida para la transmisión de datos.

Históricamente, todas las computadoras diseñadas desde los años 50 han tenido estas unidades funcionales, pero al paso del tiempo se han empleado diferentes tecnologías para implementarlas, siendo la tecnología principal en los años 50 la válvulas de vacío, en los 60 los transistores y desde los años 70 los circuitos integrados.

Dentro del mundo de la electrónica digital existen básicamente dos maneras de diseñar o construir un circuito: mediante componentes discretos y circuitos integrados, o bien utilizando microcontroladores, explicado mediante términos sencillos, podemos definir a un microcontrolador como un circuito integrado programable capaz de realizar cálculos con gran rapidez obedeciendo a un

conjunto de instrucciones (chip) que incluye en su interior las tres unidades funcionales de una computadora, es decir, procesador central, memoria y dispositivos de entrada-salida y opcionalmente algunos dispositivos periféricos que auxilian al microcontrolador a cumplir la tarea para la que ha sido programado

Para definir un microcontrolador, partiremos del concepto de microcomputador, la *figura 3.1* muestra el esquema general básico de un microcomputador, éste es un circuito integrado o chip que está compuesto en su interior por las tres unidades funcionales de una computadora: CPU (Unidad Central de Procesamiento, **C**entral **P**rocessing **U**nit), memoria y unidades de E/S (entrada/salida), estas unidades se conectan entre sí mediante grupos de líneas eléctricas denominadas buses, los buses pueden ser de: direcciones (si transportan direcciones de memoria o de entrada y salida), datos (si transportan datos o instrucciones) y de control (si transportan señales de control diversas).

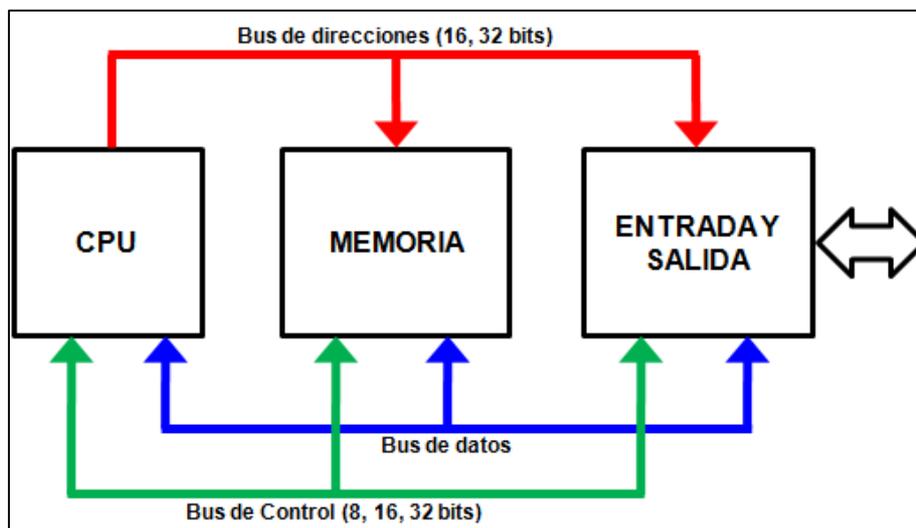


Figura 3.1 Esquema básico general de un microcomputador. La CPU es el microprocesador

La CPU es el “cerebro” del microcomputador compuesto por un **microprocesador** y actúa bajo el control del programa almacenado en la memoria. La CPU se encarga básicamente de traer las instrucciones del programa desde la memoria, interpretarlas y hacer que se ejecuten, la CPU también incluye la ALU (Unidad Aritmética Lógica, **A**rithmetic and **L**ogic **U**nit), que es la encargada de realizar las

operaciones aritméticas y lógicas elementales con los datos binarios. Un microcomputador es una computadora de **propósito general**, un microcontrolador es un circuito integrado programable de **uso específico** que al igual que el microcomputador incluye en su interior las tres unidades funcionales de una computadora: CPU, memoria y unidades de E/S, componentes necesarios para controlar el funcionamiento de una tarea determinada, como un teclado, una impresora, un sistema de alarma, el control de una lavadora, hornos de microondas, etc.

Un **microcontrolador** es un circuito integrado programable capaz de realizar cálculos con gran rapidez obedeciendo a un conjunto de instrucciones. En su interior posee los mismos elementos que una computadora digital, es decir, procesador central, memoria y dispositivos de entrada-salida y opcionalmente algunos dispositivos periféricos que auxilian al microcontrolador a cumplir la tarea para la que ha sido programado.

Un microcontrolador tiene una **arquitectura cerrada** porque todos los elementos que necesita para funcionar están dentro del mismo chip. Los microcontroladores son de **propósito específico** pues una vez programados y configurados residen en el sistema que van a controlar y sólo sirven para una tarea en particular, a diferencia de los microprocesadores que son de **propósito universal**.

Los microcontroladores son usados en diversas áreas como comunicaciones digitales, medicina, instrumentación científica, aparatos electrodomésticos, sistemas industriales, agricultura y un gran número de aplicaciones más.

Una de las empresas líderes en el desarrollo de microcontroladores es *Microchip Technology Inc.*, quien ofrece un gran número microcontroladores de aplicación general (MCU PICmicro) y controladores para el **procesamiento digital de señales** (DSPic). Dentro de estos productos destacan los microcontroladores PIC quienes han tenido gran aceptación debido a que *Microchip* ofrece abundante información sobre sus productos, además de herramientas en hardware y

software de bajo costo para el desarrollo de aplicaciones con sus circuitos integrados.

Los microcontroladores PIC tienen su origen en los años 70. Fueron diseñados por una empresa llamada GI Microelectronics Division que más tarde se convertiría en Microchip Technology Inc. Microchip construyó los microcontroladores PIC como parte de un proyecto para auxiliar al microprocesador CP1600 en el manejo de entrada y salida de datos: el *Peripheral Interface Controller* o PIC por sus siglas en inglés.

Resumiendo podemos decir que las principales características que diferencian a un microcontrolador de un microprocesador son:

1.- Los microcontroladores son sistemas cerrados, ya que contienen todos los elementos de una computadora en un solo chip, frente a los microprocesadores que son sistemas abiertos, ya que sacan las líneas a los buses de datos, direcciones y control exterior, para la conexión de memorias, interfaces de E/S, etc.

2.- Los microcontroladores son de propósito específico, es decir, son programados para realizar una única tarea, mientras que los microprocesadores son de propósito general.

3.1 Arquitectura general de los microcontroladores PIC

La **arquitectura de una computadora** se refiere a las características de los elementos que la conforman, la manera en que interactúan y la forma en que se interconectan para que la información viaje entre sus unidades funcionales.

Las principales arquitecturas que podemos encontrar en el diseño de microcontroladores y microprocesadores son dos: la **arquitectura Von Neumann** y la **arquitectura Harvard**.

3.1.1. Arquitectura Von Neumann

John Von Neumann (28 de diciembre de 1903 - 8 de febrero de 1957) propuso el modelo que se aplica a la mayoría de las computadoras digitales y en honor a él lleva su nombre. Para que una computadora pueda realizar una tarea debe seguir una serie de instrucciones a las que conocemos como **programa**. El programa se guarda en una memoria no volátil de sólo lectura, es decir, que no pierde la información aunque se apague la computadora.

De manera general, la arquitectura Von Neumann *figura 3.2* posee sólo una memoria principal donde se almacenan datos e instrucciones. En un microcontrolador esto no es del todo cierto ya que en un microcontrolador sólo hay un programa activo y no es necesario copiarlo a la memoria principal. El programa suele ser almacenado en una memoria de tipo no volátil o ROM que comparte el espacio de direcciones y por tanto las líneas de control y dirección con la memoria de datos RAM para formar un sólo espacio de direcciones como si se tratara de un bloque de memoria único.

ARQUITECTURA VON NEUMANN

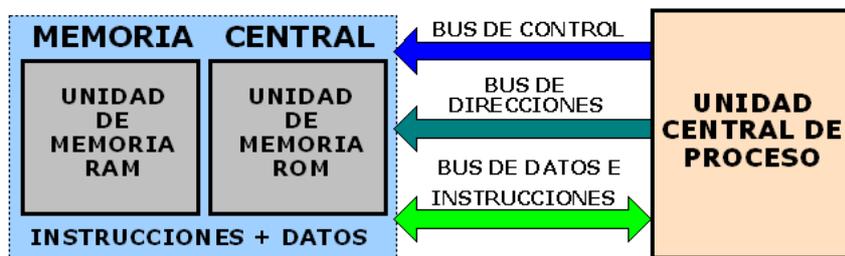


Figura 3.2 Esquema arquitectura Von Neumann

La Unidad Central de Proceso accede a la memoria de instrucciones y a la memoria de datos a través de un mismo conjunto de alambres llamado **bus de datos e instrucciones**. Compartir el bus para datos e instrucciones tiene ventajas y desventajas, entre ellas podemos comentar:

- No puede accederse a una instrucción y a un dato al mismo tiempo ya que comparten la misma línea o bus. Esto trae en consecuencia que la ejecución de un programa se realice en mayor tiempo.

- La longitud de la palabra para representar una instrucción o un dato, así como la longitud del bus por el que viajan es del mismo tamaño. Si los códigos de operación necesitan más bits para representarse que los datos, entonces deben ser almacenados en dos o más partes, lo que implica que habrán que hacerse más ciclos de lectura sobre la memoria de instrucciones haciendo la ejecución del programa más lenta.

En el momento histórico en que fue presentado se adoptó sobre otras arquitecturas al grado de convertirse en la arquitectura tradicional ya que era más fácil y económica de construir comparada con una arquitectura Harvard.

3.1.2. Arquitectura Harvard

Para optimizar el trabajo de un microprocesador podemos recurrir a diversas técnicas de diseño. Una de ellas es aumentar la frecuencia de reloj para así ejecutar más instrucciones por unidad de tiempo, pero muchas de las operaciones que una computadora realiza son de lectura y escritura a unidades de memoria que suelen ser mucho más lentas que el microprocesador, por lo que la CPU estará ociosa durante mucho tiempo esperando a que el dato o la instrucción estén listos para ser procesados. Una solución sería hacer unidades de memoria que fueran tan rápidas como el microprocesador pero el costo del sistema tendería a aumentar.

Otra solución considera usar más de una CPU en diferentes arreglos para aumentar el rendimiento, pero esta solución sigue influyendo en el precio final del sistema por lo que el uso de alguna de estas opciones tiene que justificarse según las necesidades de la aplicación donde serán montados.

Una tercera alternativa está en modificar el cómo viajan las instrucciones y los datos de las unidades de memorias a la CPU. Si podemos hacer que viajen por caminos separados es posible, con el diseño de hardware apropiado, acceder a la memoria de instrucciones y a la memoria de datos al mismo tiempo. Esta propuesta se conoce como arquitectura Harvard. *Figura 3.3.*

ARQUITECTURA HARVARD

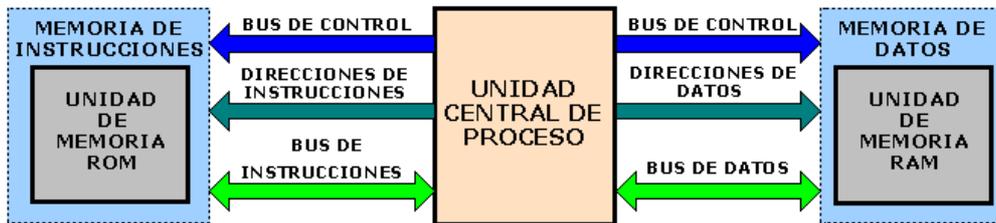


Figura 3.3 Esquema arquitectura Harvard

La arquitectura Harvard fue desarrollada en la Universidad de Harvard en 1944 por Howard Aiken. Fue implementada en la IBM *Automatic Sequence Controlled Calculator* (ASCC) también llamada Mark I, una computadora electromecánica. La Mark I era programada a través de un panel de interruptores y los datos eran introducidos en un módulo lector de tarjetas perforadas.

Tener buses independientes para la memoria de datos y la memoria de instrucciones tiene ventajas sobre la arquitectura Von Neumann:

- El acceso a cada memoria puede ser simultáneo y a través de buses independientes por lo que se reduce el tiempo que la CPU emplea en acceder a las unidades de memoria.
- El número de bits que forman los buses de datos e instrucciones pueden ser de diferente tamaño. Lo que implica que los códigos de operación pueden ser optimizados para que ocupen sólo una localidad de memoria, reduciendo los ciclos de lectura para obtener una instrucción o un dato.

Una versión diferente de esta arquitectura es llamada **arquitectura Harvard modificada** y consiste en colocar una pequeña cantidad de memoria muy rápida entre la memoria de datos y la CPU, donde se colocarán los datos que son más frecuentemente utilizados. Con esto se logra disminuir el tiempo que la CPU tiene que esperar por un ciclo de lectura o escritura a la memoria. Esta pequeña memoria se llama **CACHE** y lo mismo se hace entre la memoria de instrucciones y la CPU.

Los microcontroladores PIC y los nuevos modelos de microcontroladores como los **DsPIC**¹ de Microchip incorporan la arquitectura Harvard modificada.

3.1.3. Procesador Segmentado

Otra técnica usada en el diseño de microprocesadores es la **segmentación encausada**, en inglés *pipeline*. Esta técnica consiste en dividir la ejecución de una instrucción en un número finito etapas, cada una con la misma duración. De la misma manera, el procesador es diseñado de forma modular para permitir la ejecución de múltiples instrucciones en paralelo, pero en etapas diferentes, pasando el resultado de una etapa a la unidad siguiente que la atenderá.

Para ejecutar una instrucción en un microcontrolador PIC, es necesario invertir 8 ciclos de reloj. 4 ciclos de reloj son empleados en la primera etapa que consiste en la búsqueda y preparación de la instrucción. Los siguientes 4 pulsos de reloj son ocupados en la segunda etapa correspondiente a la ejecución y almacenamiento de los resultados.

Al implementar en un microcontrolador PIC la segmentación encausada es posible ejecutar la instrucción actual y de manera simultánea preparar la siguiente, reduciendo el tiempo que se emplea a sólo 4 ciclos de reloj. Estos 4 ciclos de reloj son llamados **ciclo de instrucción** y es el tiempo que tarda el procesador del PIC en buscar, decodificar y ejecutar una instrucción.

La mayoría de las instrucciones en un microcontrolador PIC tardan un ciclo de instrucción en ser ejecutadas, con excepción de las instrucciones que involucran saltos en el orden del programa, las cuales duran dos ciclos debido a que no se conoce el lugar en la memoria dónde se encuentra la siguiente instrucción a ejecutarse hasta que no haya sido procesada la instrucción de salto.

Como ejemplo podemos analizar el siguiente programa dónde la instrucción 3 es una instrucción de salto. *Figura 3.4.*

¹ DsPIC ®. Digital Signal PIC ®. Es un microcontrolador diseñado para el procesamiento digital de señales, usado en aplicaciones de audio, video y comunicaciones, entre otras.

```

Instrucción No 0  MOVLW  0x05    ; Carga el registro de trabajo con 0x05
Instrucción No 1  ADDWF   0x20, 1 ; Suma el registro de trabajo con el contenido
                               ; del registro 0x20
Instrucción No 2  RRLF    0x02, 1 ; Rota a la derecha el contenido del registro
                               ; 0x20
Instrucción No 3  GOTO    0x00    ; Salta a la instrucción 1
Instrucción No 4  CLRF    0x06    ; Esta instrucción no es ejecutada

Instrucción No 5  RRLF    0x02, 1 ; Esta instrucción no es ejecutada
    
```

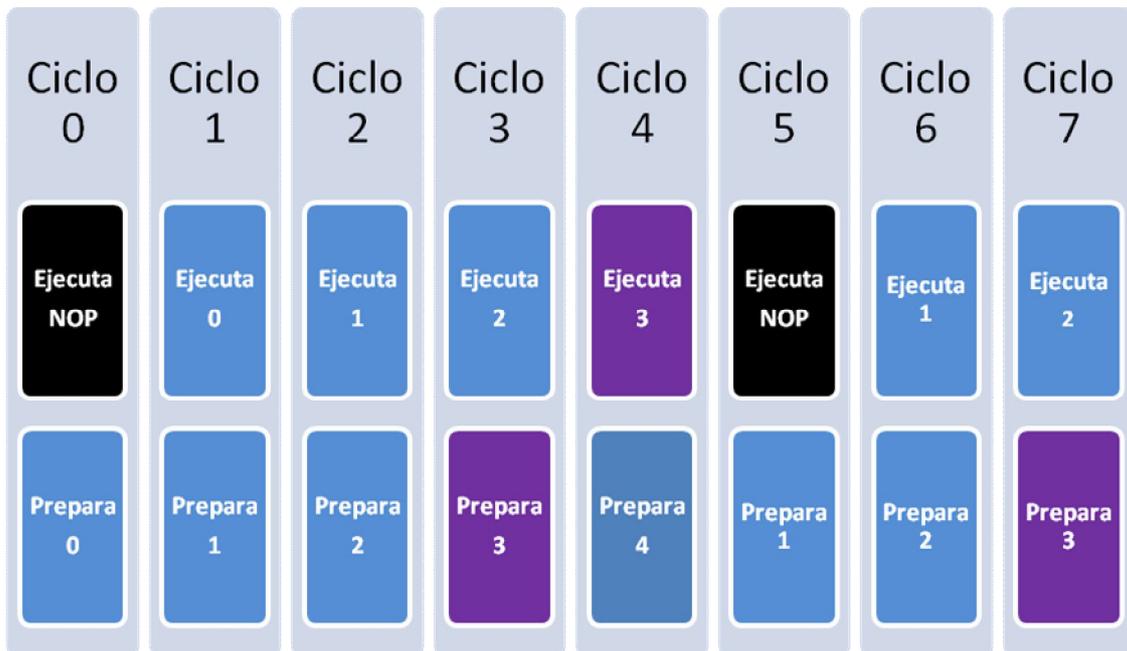


Figura 3.4 Diagrama de programación segmentada o pipe-line

Cada 4 pulsos de reloj se genera un ciclo de instrucción. En el ciclo 0, cuando el sistema se reinicia, no existe una instrucción lista para ser procesada por lo que se ejecuta una instrucción que no modifica el estado de ninguna de las unidades del procesador mientras que la instrucción 0 es preparada, la instrucción NOP.

Durante el siguiente ciclo, el ciclo 1, la instrucción 0 está lista para ejecutarse mientras que de manera paralela la instrucción 1 es preparada para su ejecución. Así seguirá la secuencia hasta el ciclo 4.

Ahora analicemos que sucede cuando se ejecuta la instrucción 3 (GOTO 0x05) una instrucción de salto. La instrucción 3 es preparada durante el tercer ciclo y está lista para ejecutarse en el ciclo 4. De manera paralela, en el ciclo 4 es

preparada la instrucción 4, pero al llegar al quinto ciclo la instrucción 4 que está lista para ejecutarse es sustituida por una instrucción NOP.

¿Por qué no fue ejecutada la instrucción 4? La razón es que en el programa la instrucción 3 obliga al procesador a brincar a la instrucción 1, lo que rompe con la ejecución normal del programa. La **fila de procesamiento** debe ser llenada nuevamente, iniciando con la instrucción 1 que es la siguiente en ser ejecutada.

Si analizamos la gráfica, podemos ver que la primera instrucción y todas las instrucciones que involucren saltos en la ejecución del programa tomarán 2 ciclos en ejecutarse ya que la fila de procesamiento debe ser llenada nuevamente.

3.1.4. Procesador RISC

Para los años 80 las técnicas de construcción de microprocesadores estaban orientadas a dotarlos con una gran cantidad de instrucciones, como consecuencia de buscar mantener la compatibilidad con antiguas versiones y otorgarles nuevas funcionalidades. Algunas de estas instrucciones tardaban varios ciclos de máquina en ejecutarse y ocupaban más de una localidad de memoria en ser representadas junto con sus operandos por lo que se consideraba que eran complejas y poderosas.

En 1980 un grupo de investigadores de la Universidad de California Berkeley, dirigido por David A Patterson y Carlo Séquin, comenzó a diseñar chips de CPU a los que dotaron de un juego de instrucciones más reducido que se ejecutaban con rapidez. También demostraron matemáticamente que a partir de la combinación de instrucciones más sencillas que se ejecutaran rápidamente podían construirse instrucciones igual de poderosas y complejas como las que tenían los demás procesadores de la época, con un diseño en hardware más reducido y un costo menor.

El acrónimo RISC (*Reduced Instruction Set Computer*) significa **computadora de juego de instrucciones reducido** y fue utilizado para describir el diseño de microprocesadores que contaban con un número pequeño de instrucciones,

alrededor de 50, que se ejecutan por lo regular en un **ciclo de máquina** y que son ortogonales, es decir, que guardan un mismo formato².

Un elemento del procesador conocido como **decodificador de instrucción** se encarga de indicarle a la CPU de que instrucción se trata y así habilitar los demás elementos que necesita para ejecutar la instrucción. Ya que las instrucciones son ortogonales, es más fácil construir hardware que las decodifique y ejecute.

Para guardar la compatibilidad con arquitecturas ya definidas se siguieron construyendo procesadores que fueron conocidos como de **arquitectura CISC** (*Complex Instruction Set Computer*) o **Computadoras de Juego de Instrucciones Complejo** que reunían un conjunto de 200 a 300 instrucciones y ocupaban varios ciclos de máquina en ejecutarse.

Existe otra tendencia en el desarrollo de microprocesadores, los de **arquitectura SISC** (*Specific Instruction Set Computer*) que hacen referencia a procesadores con un **Juego de Instrucciones Específico** como los usados para la codificación y decodificación de formatos tanto de audio como de videodigital y para el procesamiento de archivos de música, video, fotografías, entre otros.

3.1.5. Arquitectura Ortogonal

En una **arquitectura tradicional** o no ortogonal *figura 3.5* los resultados de una operación que realice la Unidad Aritmética-Lógica (ALU) se guardan en un registro de propósito específico llamado **Acumulador**. El resultado puede ser utilizado entonces como operando de una nueva instrucción o ser movido a un registro de la memoria de datos.

En una **arquitectura ortogonal**, el resultado de la ejecución de una instrucción puede guardarse no necesariamente en el acumulador, también puede ser movido directamente a un registro de la memoria de datos sin haber pasado antes por el acumulador. En los microcontroladores PIC, el registro que funciona como acumulador se llama **registro de trabajo** o **registro w** del inglés *work*.

² El formato de la instrucción se refiere a la cantidad de bits que son usados para representar la instrucción, y sus operandos, generalmente en una sola localidad de memoria.

Un microprocesador con arquitectura ortogonal tiene la ventaja de que ahorra el uso de ciclos de instrucción, ya que cualquier resultado puede colocarse en el registro de trabajo o en el mismo registro de donde se tomo el dato ahorrando una instrucción de movimiento.

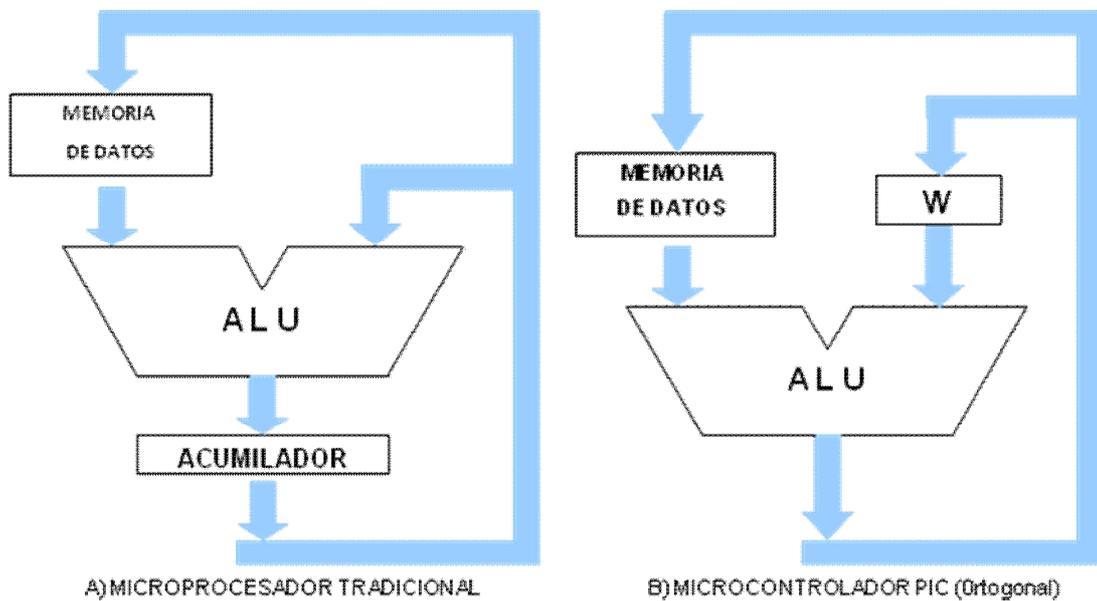


Figura 3.5 Arquitectura tradicional y ortogonal

3.1.6. Arquitectura basada en un banco de registros

La definición de **Arquitectura Basada en Banco de Registros** nos dice que cualquier elemento de la arquitectura puede ser implementado físicamente como un registro en la memoria de datos. En un microcontrolador PIC se reserva una sección de la memoria de datos llamada **Registros con Función Específica** o SFR (*Special Function Registers*) que es usada para configurar o conocer el estado tanto de los elementos del procesador como de los recursos periféricos.

3.2. Características Especiales del Procesador

3.2.1. Alimentación Eléctrica

Un microcontrolador funciona con una alimentación eléctrica de 5V, como todos los circuitos integrados de la familia TTL³. Los microcontroladores PIC tienen

³ TTL. Lógica Transistor-Transistor. La familia de circuitos integrados TTL está diseñada con arreglos de transistores.

dispuestos al menos dos terminales para la alimentación eléctrica, una que se conecta al voltaje positivo (5V) y otra tierra.

De manera típica se considera que el voltaje mínimo de alimentación para que el microcontrolador funcione adecuadamente es de 4.5V y el máximo de 5.5V antes de que el microcontrolador sufra un daño irreparable.

Es necesario considerar que a mayor frecuencia de trabajo el microcontrolador PIC consumirá mayor corriente, lo que implica que disipará mayor cantidad de calor. Existen modelos que soportan diferentes rangos de temperatura, humedad y frecuencia de trabajo lo que permite elegir el microcontrolador adecuado a la aplicación que estemos diseñando.

En aplicaciones donde es necesario un consumo menor de energía eléctrica se usan los microcontroladores de bajo consumo de potencia. Estos microcontroladores se pueden alimentar desde 2.5V hasta 5.5V lo que permite que funcionen en sistemas donde el ahorro de energía y disipación de calor son un factor importante.

Estos microcontroladores funcionan bajo el mismo principio de la frecuencia de trabajo, a mayor frecuencia se consume mayor corriente, por lo que el nivel mínimo de alimentación se incrementa proporcionalmente entre los 2.5V y los 4.5V para garantizar su correcto funcionamiento.

3.2.2. El oscilador Principal

Los circuitos digitales emplean una señal que permite la sincronización de los procesos dentro del sistema. Esta señal es conocida comúnmente como señal reloj. La **señal de reloj** es una señal uniforme como una señal senoidal o una señal de cuadrada. Dentro del microcontrolador existe un circuito llamado **oscilador principal** quien es el encargado de generar la señal de reloj y distribuirla a través de los elementos del microcontrolador que la requieran.

Por lo general, para generar la señal de reloj el oscilador principal necesita de elementos externos que permitirán fijar la velocidad de trabajo. Estos elementos se conectan a dos terminales nombradas como OCS1/CLKIN y OSC2/CLKOUT.

Algunos modos de operación para el oscilador principal son:

- Oscilador LP: Cristal de baja velocidad y bajo consumo de potencia.
- Oscilador XT: Cristal de cuarzo o resonador cerámico.
- Oscilador HS: Cristal de cuarzo o resonador cerámico de alta velocidad.
- Oscilador RC: Oscilador hecho con una arreglo de resistencia y capacitor.
- Oscilador IntRC: Oscilador interno hecho con una arreglo de resistencia y capacitor.
- Oscilador ER: Oscilador que utiliza una resistencia externa para hacer funcionar un arreglo
- RC Interno en el microcontrolador.

Las configuraciones para el oscilador principal son las siguientes:

Oscilador LP

Un oscilador LP es un cristal de cuarzo de baja velocidad y bajo consumo de potencia. Su frecuencia de trabajo está comprendida entre 32 y 200kHz. Para funcionar, deben conectarse las entradas del cristal a las terminales OSC1/CLKIN y OSC2/CLKOUT respectivamente y un capacitor cerámico entre cada terminal y referencia a tierra del circuito.

Oscilador XT

Un oscilador XT puede ser un resonador cerámico o un cristal de cuarzo de mediana velocidad. Los cristales XT son los más usados y comprenden una frecuencia entre 100kHz y 4MHz.

Oscilador HS

El oscilador HS es un cristal de cuarzo o resonador de alta velocidad que trabaja a una frecuencia comprendida entre 4 y 20MHz.

El diagrama de conexión es el mismo para los osciladores LP, XT y HS. En ocasiones es necesario colocar una resistencia (R_s), entre el oscilador y la terminal OSC2/CLKOUT dependiendo del modelo de microcontrolador. Los valores típicos de los capacitores cerámicos usados para la conexión de un oscilador LP, XT y HS en un microcontrolador PIC16F84A están en la Tabla 3.1.

Tipo de Oscilador	Frecuencia del Cristal	Valor del Cap. 1	Valor del Cap. 2
LP	32 KHz	68 - 100 pF	68 - 100 pF
	200 KHz	15 - 33 pF	15 - 33 pF
XT	100 KHz	100 - 150 pF	100 - 150 pF
	2 MHz	15 - 33 pF	15 - 33 pF
HS	4 MHz	15 - 33 pF	15 - 33 pF
	4 MHz	15 - 33 pF	15 - 33 pF
	20MHz	15 - 33 pF	15 - 33 pF

Tabla 3.1. Valor de capacitores para cristales de cuarzo para el microcontrolador PIC® 16F84A.

Otra opción de configuración del oscilador principal es el oscilador RC.

Oscilador RC

Un **oscilador RC** está formado por una resistencia y un capacitor. Es útil en aplicaciones donde no se requiere exactitud en los tiempos (como aplicaciones que no usan los puertos de comunicación serie) ya que su mayor inconveniente es que no se trata de un oscilador muy exacto pero si muy económico. La figura 10 muestra el arreglo de este oscilador y la tabla 3.2 algunos valores para el arreglo RC.

Los valores recomendados para Rext y Cext están dados por la relación: $5K\Omega \leq R_{ext} \leq 100 K\Omega$ y $C_{ext} > 20 pF$. A través de la terminal OSC2/CLKOUT sale la cuarta parte de la frecuencia de oscilación total, esta señal puede ser aprovechada para ingresarse por la terminal OSC1/CLKIN de otro microcontrolador y así hacer funcionar varios microcontroladores en cascada partir de una sola señal de reloj.

Cext	Rext	Frecuencia
20 pF	5 KΩ	4.61 MHz
	10 KΩ	2.66 MHz
	100 KΩ	311 KHz
100 pF	5 KΩ	1.34 MHz
	10 KΩ	756 KHz
	100 KΩ	82.8 KHz
300 pF	5 KΩ	428 KHz
	10 KΩ	243 KHz
	100 KΩ	26.2 KHz

Tabla 3.2. Valores para un oscilador RC

3.2.3. Reinicio (Reset)

El **sistema de Puesta a Cero** o **Reinicio del Sistema (RESET)** se utiliza para poner al procesador y sus periféricos en un estado conocido antes de iniciar la ejecución del programa. Saber que origina el reinicio del microcontrolador es fácil, sólo es necesario checar dos bits del registro de estado del microcontrolador (*STATUS*) o el registro PCON (Power Control, Dirección 87H).

La mayoría de los microcontroladores PIC responden las siguientes fuentes de reinicio:

- *Power-on Reset (POR)*. Reinicio al conectar la alimentación eléctrica.
- Reinicio a través de la terminal *MCLR* durante la operación normal del microcontrolador.
- Reinicio a través de la terminal *MCLR* durante el estado de bajo consumo (*SLEEP*) del microcontrolador.
- Reinicio por el desbordamiento del Perro Guardián (*WTD*) durante la operación normal del microcontrolador.
- *Brown-on Reset (BOR)*. Reinicio por falla en la alimentación durante la operación normal del microcontrolador.
- Reinicio por error de paridad (*PER*).

El contenido de la mayoría de los registros del microcontrolador no son afectados al entrar en el estado de reinicio. Para conocer como es afectado cada uno de los registros de un modelo en particular es necesario revisar el manual de operación.

Para que un microcontrolador funcione en modo normal la terminal debe estar conectada en un nivel lógico alto (5V), mientras que para entrar en estado de reinicio es necesario un nivel lógico bajo (0v). Para conmutar entre estos dos estados puede conectarse un interruptor a una resistencia *pull-up*, o *resistencia de polarización*.

Si el interruptor está abierto, el microcontrolador recibirá un nivel de voltaje alto por la terminal *MCLR*, lo que supone que el microcontrolador está en un estado de operación normal. Cuando el interruptor es presionado, la terminal *MCLR* recibirá un nivel de voltaje bajo, por lo que entrará en el estado de reinicio. Mientras el

microcontrolador se encuentre en estado de reinicio no trabajará. Cuando pase al estado normal el microcontrolador ejecutará el programa desde el inicio.

3.2.4. Memoria de Usuario

La memoria de usuario es una zona especial en la memoria de instrucciones que sólo puede ser accedida durante la programación del microcontrolador. Esta zona está mapeada desde la dirección 2000h hasta 2007h.

No todos los microcontroladores tienen implementados todos estos registros, pero la mayoría hace uso de estos registros para tres propósitos: **Localidades de Identificación**, **Identificador de Dispositivo** y **Bits de Configuración**.

Los bits de configuración se encuentran en la localidad 2007. Un bit no programado se lee como 1 y un bit programado se lee como 0, por lo que un microcontrolador sin programar leerá la palabra de configuración con el valor 3FFFh. Entre las opciones que se pueden configurar en los bits de configuración están:

- Selección del Modo de Oscilador.
- Protección del código de la memoria de programa.
- Protección del código de la memoria EEPROM de datos.
- Activación de la escritura de la memoria de programa.
- Activación del modo de depuración en circuito.
- Activación del sistema de programación de bajo voltaje.
- Activación del sistema *Brown-on Reset*.
- Activación del sistema *Power-up Timer*.
- Activación del Perro Guardián.

Las localidades que están mapeados de la dirección 2000h a 20003h de la memoria de instrucciones están diseñadas para que el usuario pueda almacenar una **suma de comprobación** (*checksum*) o un código de identificación (*ID Locations*). Microchip recomienda usar sólo los 4 bits menos significativos de cada localidad, para tener 4 dígitos de 4 bits cada uno.

En la dirección 2006 de algunos microcontroladores es posible encontrar un registro con un código único que sirve para identificar el modelo de

microcontrolador. Este identificador de dispositivo es usado por los dispositivos programadores para asegurarse que está bien configurado para programar un microcontrolador. Este registro es de sólo lectura.

3.2.5. Interrupciones

La mayoría de los microcontroladores permiten fuentes de interrupción. Una interrupción suspende la ejecución normal de programa para dirigir el flujo a una sección especial de código llama **Rutina de Servicio a la Interrupción**.

Una vez que la interrupción es atendida, se ejecuta una instrucción especial que obliga al microcontrolador a regresar a la última instrucción que quedo pendiente de ser cuando se entro en la Rutina de Servicio a la Interrupción.

Diferentes periféricos pueden generar una interrupción. Para permitir que estos periféricos generen interrupciones o averiguar que periférico genero una interrupción se hace uso de diferentes banderas y bits de configuración contenidos en registros de con función específica.

Las interrupciones permiten atender eventos en tiempo real, justo cuando suceden, en lugar generar rutinas que ocupen al microcontrolador en preguntar si el evento a pasado.

3.2.6. Programación Serie en Circuito

Los microcontroladores PIC se programan mediante un protocolo serie maestro-esclavo llamado *In-Circuit Serial Programming* o ICSP. El protocolo hace uso de 5 líneas. Dos de ellas son para transmisión bidireccional de datos y envío de la señal de reloj respectivamente, mientras que las restantes son usadas para la alimentación del circuito y aplicar el voltaje de programación.

El protocolo ICSP permite la programación en campo, es decir, programar el microcontrolador después de que ha sido montado en la aplicación. Así, el fabricante puede construir tarjetas con dispositivos no programados para posteriormente programar el microcontrolador con el firmware más reciente o un firmware a la medida.

3.3. Tipos de Encapsulados

Una de las etapas del diseño de soluciones tecnológicas con microcontroladores es elegir el encapsulado que usaremos en el prototipo final. Elegir el encapsulado más adecuado permite aprovechar el espacio en la placa de circuitos y una mejor distribución de los componentes.

Existen gran cantidad de encapsulados y seguramente emplearemos el encapsulado de Doble Línea o DIP (*Dual Inline Package*) durante la etapa de desarrollo y prueba, pero en el prototipo final un encapsulado de montaje superficial o uno de PLCC (*Plastic Leaded Chip Carrier*) pueda mejorar la distribución del espacio.

3.3.1. DIP (*Dual Inline Package*)

Este encapsulado es quizá el más conocido. Presenta dos líneas de terminales en lados opuestos. Sus dimensiones varían dependiendo del número de terminales que pueden llegar a ser de 20 por lado. Las terminales se cuentan en sentido anti horario, comenzando por la número 1 que es indicada por una marca o hendidura.

3.3.2. SOIC (*Small Out-Line Integrated Circuit*)

Puede decirse que este tipo de encapsulado es el equivalente a un encapsulado DIP pero de montaje superficial, es decir, se suelda a una cara de la placa por lo que no hay necesidad de perforarla. Sus terminales simulan la forma de un ala de gaviota.

3.3.3. SSOP (*Shrink Small Out-Line Package*)

Este tipo de encapsulado es similar al SOIC, con la diferencia de que el espacio entre terminales es más pequeño. Poseen dos líneas de terminales a lados opuestos del encapsulado en forma de ala de gaviota. El número de terminales varía de 6 a 28.

3.3.4. PLCC (*Plastic Leaded Chip Carrier*)

Este tipo de encapsulados de circuito integrado necesita de una base para ser colocado sobre la placa del circuito que controlará por lo que no puede ser soldado directamente a la placa. Las terminales están dispuestas en los cuatro lados del encapsulado y son numeradas a partir de la terminal uno que se indica con una marca en sentido anti horario.

Para retirarlo de la base es necesario auxiliarse de una pinza especial. Sus terminales tienen una forma como de *J*, lo que permite que se sujeten a la base y hagan contacto con las terminales de la misma base.

3.3.5. TQFP (*Thin Quad Flatpack Package*)

Este tipo de encapsulados es similar al PLCC ya que tiene terminales a los cuatro lados del encapsulado. Las terminales son en forma de ala de gaviota y es una variante del QFP que tiene el espacio entre terminales más pequeño. Estos circuitos integrados no se sueldan a la placa, sino que se adhieren a ella con una pasta especial termo adherible.

3.3.6. QFN (*Quad Flatpack Package No Leads*)

Un encapsulado de circuito integrado de este tipo es similar al QFP, pero sus terminales no están extendidas hacia afuera, sino hacia dentro. Es unido a la placa con una pasta termo adherible por lo que no necesita ser perforada la placa de circuito impreso.

3.4. Los microcontroladores PIC están agrupados en familias

Con todo lo comentado en las secciones anteriores podemos resumir que los microcontroladores PIC son microcontroladores con un procesador basado en una arquitectura Harvard RISC de 8 bits, por lo que las instrucciones, dependiendo del modelo, pueden representarse con 12, 14 o 16 bits, mientras que los datos sólo con 8 bits.

Una de las características más interesantes es el diseño segmentado del procesador (pipeline) que permite la ejecución de las instrucciones en un menor tiempo. También implementan una arquitectura basada en bancos de registros.

Los encapsulados varían en diferentes diseños como DIP⁴, SOIC⁵ y PLCC⁶ formados desde 6 hasta 80 terminales. Lo que permite elegir el encapsulado más adecuado a la aplicación que estemos diseñando para aprovechar de manera óptima el espacio.

Es relativamente fácil migrar un programa entre diferentes modelos debido a que poseen la misma arquitectura y sólo varían sus características respecto al número y cantidad de recursos que implementan, para así encontrar el mejor modelo que se adapte a nuestro sistema.

Cada microcontrolador posee diferentes periféricos dentro del mismo chip, entre los que se encuentran módulos de captura y comparación, circuitos generadores de modulación por ancho de pulsos, contadores-temporizadores, temporizadores de perro guardián, convertidores analógicos digitales de 10 y 12 bits, circuitos de protección contra fallo en la alimentación, circuitos para referencia de voltaje, un estado de bajo consumo y circuitos para la comunicación serie, entre otros controladores que soporten los protocolos de comunicación más comunes como RS232/RS485, SPI, I²C, CAN, USB, LIN, Radio RF, TCP/IP.

Una forma de Microchip para organizar sus microcontroladores PIC de 8 bits es en familias. Esta división se realiza dependiendo de las características que cada microcontrolador incorpora, como el tamaño de palabra, de memoria de programa, el número de instrucciones para programarlo, número de terminales de su encapsulado y dispositivos periféricos que incorpora cada chip.

Al pasar el tiempo Microchip ha desarrollado nuevos modelos, mientras que otros ya han salido del mercado, por lo que hablar de una división rigurosa de qué modelo pertenece a que familia es un poco difícil. Al momento de escribir este texto Microchip habla de 4 familias, la familia 10F, 12F, 16F y 18F. Otros autores

⁴ DIP (*Dual In line Package*) Encapsulado de doble línea

⁵ SOIC (*Small Outline Integrated Circuit*) Circuitos Integrados de Líneas de Salidas Pequeñas

⁶ PLCC (*Plastic Leaded Chip Carrier*) Circuito Integrado Portador de Plástico con Plomo

clasifican a los microcontroladores PIC en 5 gamas conocidas como enana, baja, media, alta y mejorada.

3.4.1. Familia Enana y Baja

La familia enana es conocida así porque sus encapsulados son de 6 u 8 terminales. Está formado por los microcontroladores PIC 10F y 12F. Son los modelos más básicos de microcontroladores PIC, son usados en aplicaciones que requieren bajo consumo de potencia, pocas terminales y un tamaño reducido, por ejemplo en un control remoto para la alarma del auto.

El tamaño de palabra con que se representa una instrucción o **Código de Operación** en las gamas baja y enana es de 12 bits con lo que se forma un repertorio de 33 instrucciones. El tamaño de la memoria de instrucciones varía dependiendo del modelo desde 256 hasta 512 palabras para la familia 10F, mientras que en la familia 12F los modelos se componen de 512, 1024 o 2048 palabras.

Dependiendo del modelo, la frecuencia máxima de operación es de 4 y 8 MHz para la familia 10F y de 4, 8 y 20 MHz para la familia 12F. La pila sólo tiene dos niveles y no admiten fuentes de interrupción. La familia 10F no incorpora memoria de datos EEPROM mientras que en la familia 12F sólo 4 modelos la implementan con capacidades de 128 a 256 bytes.

Entre los recursos periféricos que implementa podemos encontrar:

- Sistema *Power On Reset* (POR).
- Sistema de Protección de Código (CP).
- Perro Guardián (WDT). Es un temporizador⁷ que de no ser refrescado antes terminar su cuenta reinicia el microcontrolador. Es usado para prevenir errores en la programación.
- Líneas de entrada y salida de alta corriente. Usadas para excitar ciertos periféricos como LEDs⁸. Proveen entre 20 y 25 mA.
- Modo de Reposo o Bajo Consumo (SLEEP).

⁷ Un temporizador es un registro que se incrementa o decrementa y funciona como un cronómetro.

⁸ LED. *Light Emitting Diode*. Diodo Emisor de Luz.

- Temporizador/Contador de 8 bits (TMR0).

Los modelos elegidos para comenzar a trabajar en el capítulo 5 son el PIC12F508 y PIC12F509 por ser los más completos de esta gama.

3.4.2. Familia Media

Existen varios modelos en la familia media, pero en particular son cuatro los más usados en la enseñanza de la programación de microcontroladores PIC, los modelos PIC16F84A, PIC16F628, PIC16F877A y PIC16F887. Esta es una de las gamas más amplias en cuanto a modelos, por lo que los recursos que implementan son muy bastos. En particular nos centraremos en comentar las principales características de la familia media más que en las características de algún modelo en particular.

Los códigos de operación tienen una longitud de 14 bits para formar un juego 35 instrucciones. El tamaño de la memoria de instrucciones varía dependiendo del modelo en 1, 2, 4 u 8 KWords. La pila tiene 8 niveles y aceptan diferentes fuentes de interrupción. La memoria de datos EEPROM va desde 64 a 128 y 256 Bytes.

Todos los modelos de la familia media pueden alcanzar una frecuencia máxima de operación de 20MHz.

Implementan los mismos recursos que la familia baja y enana, pero dependiendo del modelo pueden variar sus prestaciones y periféricos. Algunos de ellos son:

- Convertidor Analógico Digital (CAD).
- Temporizadores de 8 y 16 bits (TMRx).
- Puertos de Comunicación Serie.
- Puerto Paralelo Esclavo.
- Comparadores Analógicos.
- Módulo de Captura, Comparación y Modulación de Pulsos (CCP).

3.4.3. Familia Alta

La gama alta envuelve a los miembros de la familia 17F. Los códigos de operación tienen una longitud de 16 bits para formar un juego 58 instrucciones. Todos los

modelos de la familia alta pueden alcanzar una frecuencia máxima de operación de 20MHz.

La pila tiene 16 niveles y aceptan diferentes fuentes de interrupción que para su manejo hacen uso de 4 vectores de interrupción. Pueden ser configurados para trabajar en una arquitectura cerrada o abierta en donde las señales internas de dirección, control y estado salen a través de las terminales para conectarse a memorias lo que permite extender los recursos del microcontrolador.

3.4.4. Familia Mejorada

Con la aparición de nuevos modelos de microcontroladores se han aumentado las prestaciones y recursos que se incluyen en cada modelo. La gama mejorada incluye microcontroladores de la familia 18F que son los microcontroladores de 8 bits más potentes de Microchip.

Cuentan con un repertorio de 77 instrucciones de 16 bits, una pila de 32 niveles y 4 vectores de interrupción.

Además incorporan nuevos periféricos como módulos de comunicación serie que soportan los protocolos CAN, USB y Ethernet, controladores de pantallas de cristal líquido (LCD) y motores o varios temporizadores de 8 y 16 bits.

Su principal característica es que pueden ser configurados para trabajar en aplicaciones que requieren mayores cantidades de memoria de datos, permitiendo así la salida de las señales de control, datos y direcciones para la conexión de dispositivos de memoria, como si se tratará de un sistema de arquitectura abierta.

3.5. Sistema de identificación del microcontrolador

Microchip usa un sistema para identificar cada uno de sus modelos de microcontroladores y conocer sus características. Este sistema consta de colocar una serie de números y letras en una etiqueta sobre el microcontrolador. Una etiqueta como PIC16F84A-4I/P indica que:

- El modelo de microcontrolador pertenece a la familia PIC16.
- El modelo del microcontrolador es 16F84A.

- Una letra F indica que el tipo de memoria de programa es FLASH, lo que permite que sea borrada eléctricamente. Una letra C indica que el tipo de memoria no puede ser borrada (OTP) y una R que se trata de una memoria tipo ROM.
- El número 4 indica la máxima frecuencia de operación que en este caso es de 4MHz.
- Una letra I (industrial) indica que el rango de temperatura es de -40°C a +85°C. una letra E (Extended) indica que se trata de un rango extendido de -40°C a 125 °C.
- La última letra o letras indican el tipo de encapsulado que en este caso es de tipo PDIP.
 - ML = QFN (Metal Lead Frame)
 - PT = TQFP (Thin Quad Flatpack)
 - SO = SOIC
 - SP = Skinny Plastic DIP
 - P = PDIP
 - L = PLCC
 - S = SSOP

Ejemplos de este sistema de identificación son:

1. PIC16LF876A-I/SO =Microcontrolador PIC16F84A de bajo consumo y memoria flash. Rango de temperatura industrial y encapsulado SOIC.
2. PIC16F877A-I/P = Microcontrolador PIC16F877a de consumo normal y memoria flash. Rango de temperatura industrial y encapsulado PDIP.
3. PIC16C627 -E/PT = Microcontrolador PIC 16F627 de consumo normal y memoria OTP. Rango extendido de temperatura y encapsulado QTFP
4. PIC16LF627A - I/SO = Microcontrolador PIC16F627A de bajo consumo y memoria Flash. Rango de temperatura industrial y encapsulado SOIC

3.6. Conclusión capítulo 3.

Desarrollar soluciones con microcontroladores, supone tener conocimiento de varias materias del área de la computación. Este capítulo puede parecer muy teórico, pero es necesario ya que una vez comprendidos los conceptos que aquí se manejaron, podremos desarrollar el circuito idóneo y su programa, para dar solución al problema que se nos plantea.

Para desarrollar soluciones con microcontroladores es bien sabido que la mejor forma de entender la teoría es poniéndola en práctica, pero la práctica no sustituye el esfuerzo y dedicación que pongamos en el estudio, una debe ir acompañada de la otra.

CAPÍTULO 4 COMUNICACIÓN.

4.1 Estándares de comunicación.

En la industria de las comunicaciones, desde hace tiempo, se aceptó la necesidad de los estándares para definir las características físicas, mecánicas, eléctricas y de procedimiento de los equipos de comunicación. Los fabricantes de equipos de cómputo comprendieron que sus equipos deberían en general, interconectarse y comunicarse con equipos desarrollados por terceros. Los diferentes equipos utilizados en las comunicaciones, de diferentes fabricantes, deben comunicarse entre sí y es más, dada la evolución actual en la normalización de los protocolos, los clientes no admiten ya tener que desarrollar o adquirir software para adaptar protocolos de uso específico. Como consecuencia, en la actualidad la normalización se está imponiendo en todas las áreas tecnológicas.

Un protocolo de comunicación describe los siguientes aspectos:

1. El tiempo relativo al intercambio de mensajes entre dos sistemas de comunicación.
2. El formato que el mensaje debe contener, para que la comunicación entre dos computadoras, que emplean protocolos diferentes, se pueda establecer.
3. Las acciones que deben realizar en el caso de producirse error en la comunicación.
4. Las suposiciones acerca del medio ambiente en el cual se ejecutará el protocolo

Para el caso de este trabajo describiremos las características principales de los protocolos RS232, USB y mencionaremos los protocolos para modem, radio modem, IrDA, Bluetooth y WIFI.

4.2 Comunicación Alámbrica.

Comunicación alámbrica, también llamada comunicación por cable, pues tiene lugar a través de líneas o cables (tradicionalmente de cobre) que unen al emisor y al receptor. La información se transmite mediante impulsos eléctricos.

4.2.1 RS232

El puerto serie RS-232 (*Recommended Standard 232*), presente en las PC's actuales, es la forma más comúnmente usada para realizar transmisiones de datos entre PC's. El RS-232 es un estándar que constituye la tercera revisión de la antigua norma RS-232, propuesta por la EIA (*Asociación de Industrias Electrónicas, Electronic Industries Alliance*), realizándose posteriormente una versión internacional por el CCITT (*Comité Consultivo Internacional Telegráfico y Telefónico, Consultative Committee for International Telegraphy and Telephony*), conocida como V.24, las diferencias entre ambas son mínimas, por lo que a veces se habla indistintamente de V.24 y de RS-232C (incluido el sufijo "C"), refiriéndose siempre al mismo estándar. El estándar RS-232, además de especificar los niveles de voltaje y la distancia máxima de transmisión de señales digitales de datos, define otras características requeridas para conformar una interfaz de datos, que deberán ser usados cuando una comunicación serial está siendo enviada entre un equipo terminal de datos (DTE), como una computadora, instrumentos digitales o una terminal y un equipo de comunicación de datos (DCE), como un modem, por ejemplo. Los niveles de voltaje considerados en el protocolo RS-232C están definidos de la siguiente manera:

El protocolo de comunicación RS-232C maneja niveles de voltaje de -15 voltios a -3 voltios para representar al uno lógico y de +3 a +15 V para representar el cero lógico. Dependiendo de la funcionalidad del circuito de intercambio de datos, los valores eléctricos se interpretarán como datos binarios o como señales de control. Esta normalización especifica que, respecto a una referencia de tierra común, una tensión más negativa que -3 voltios se interpretará un uno binario, mientras que una tensión mayor de +3 voltios se interpretará como un cero binario

(cualquier valor entre -3 voltios y +3 voltios tiene un nivel lógico no definido). De acuerdo a esta especificación, la codificación de las señales corresponde a NZR-L (*No Retorno a Nivel Cero, Nonreturn to Zero-Level*), en donde el nivel del “0” lógico le corresponde un nivel de voltaje alto, mientras que al “1” lógico uno bajo. Típicamente, un sistema RS-232C usa voltajes nominales de -12 V para el uno lógico y +12 V para el cero lógico. Hay que notar que el voltaje más negativo corresponde al uno lógico. Esto es conocido como lógica negativa. En la *figura 4.1* se muestra lo antes mencionado. Cabe destacar que, de acuerdo a la norma, para la transmisión de datos el cable de transmisión no debe sobrepasar los 15 metros de longitud.

Para las señales de control se aplican los mismos niveles de tensión: una tensión menor de -3 voltios se interpreta como OFF y una tensión mayor de +3 voltios se interpreta como ON.

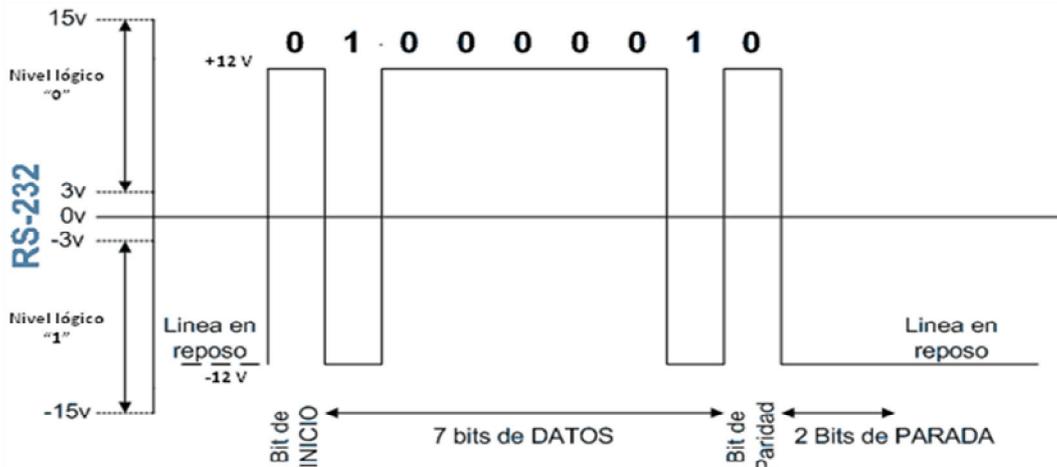


Figura 4.1 Niveles de tensión lógicos RS232

En cuanto a la especificación del número de terminales, se tienen conectores de 25 y 9 terminales, conocidos como DB-25 y DB-9. En general, en las comunicaciones bajo el estándar RS-232, se utilizan tres terminales para transmitir información: dos son canales portadores de datos (TxD y RxD) y el tercero es un conductor de tierra de señal, que sirve como camino de retorno de la corriente de

la señal. Los 22 terminales restantes sirven como terminales de control entre DTE y el DCE.

En la tabla 4.1 se muestra la distribución de las terminales de acuerdo a la señal y su descripción correspondiente. La figura 4.2 muestra como es físicamente el conector.

Número	de Pin	Señal	Descripción	E/S
En DB-25	En DB-9			
1	1	-	Masa chasis	-
2	3	TxD	Transmit Data	S
3	2	RxD	Receive Data	E
4	7	RTS	Request To Send	S
5	8	CTS	Clear To Send	E
6	6	DSR	Data Set Ready	E
7	5	SG	Signal Ground	-
8	1	CD/DCD	(Data) Carrier Detect	E
15	-	TxC(*)	Transmit Clock	S
17	-	RxC(*)	Receive Clock	E
20	4	DTR	Data Terminal Ready	S
22	9	RI	Ring Indicator	E
24	-	RTxC(*)	Transmit/Receive Clock	S

Tabla 4.1 Distribución de terminales de los conectores DB25 y DB9
 (*) = Normalmente no conectados en el DB25

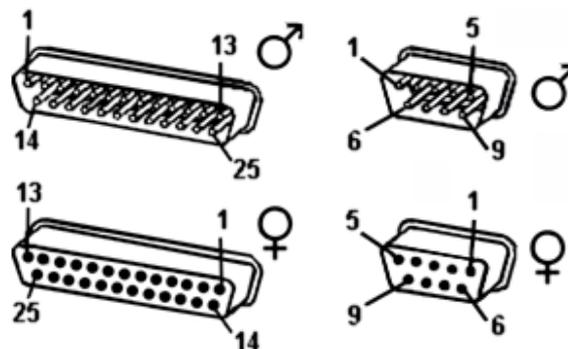


Figura 4.2 Conectores DB25 y DB9

En sistemas convencionales, los niveles de voltaje lógicos estándar están de acuerdo a los voltajes TTL estándar de 0 V y 5 V. esos voltajes son empleados también por los dispositivos MOS y CMOS del sistema. Estos niveles TTL no son

compatibles con equipo periférico que utilice la interfaz RS232 y consecuentemente algún tipo de circuito de interfaz es necesario para convertir entre voltajes TTL y RS-232C.

Los diferentes equipos que cuentan con un puerto de comunicación serial, contienen un circuito integrado denominado UART (*Transmisor Receptor Asíncrono Universal, Universal Asynchronous Receiver Transmitter*) o USART (*Transmisor Receptor Síncrono Asíncrono Universal, Universal Synchronous Asynchronous Receiver Transmitter*). La función principal de estos circuitos integrados es transformar el flujo de información en paralelo a un flujo de información en serie. Normalmente se utilizan los siguientes modelos del circuito integrado UART: 8250 (bastante antiguo, con fallos, sólo llega a 9600 baudios), 16450 (versión corregida del 8250, llega hasta 115,200 baudios) y 16550A (con buffer de E/S). A partir de la gama Pentium, en las computadoras, la circuitería UART de la placa base son todas de alta velocidad, es decir UART 16550A. De hecho, la mayoría de los módems que son conectados al puerto serie necesitan dicho tipo de UART, incluso algunos juegos para red a través del puerto serie necesitan de este tipo de puerto serie. Las computadoras portátiles suelen llevar circuitos integrados 82510 (*con buffer especial, emula al 16450*) o el 8251 que es un tipo de circuito USART, compatible con los microprocesadores Intel.

Para controlar el puerto serie, la CPU emplea direcciones de puertos de E/S y líneas denominadas de interrupción, IRQ (*Petición de Interrupción, Interrupt ReQuest*); las direcciones 3F8h (o 0x3F8) e IRQ 4 se utilizan para el puerto de comunicaciones definido como COM1, y 2F8h e IRQ 3 para el COM2. Al añadir otros puertos serie, se eligieron las direcciones 3E8 y 2E8 para COM3-COM4, pero las IRQ no están especificadas, es decir, la IRQ para estos COM son asignadas por el sistema operativo, dentro del rango de hasta IRQ 15.

Mediante los puertos de E/S se pueden intercambiar datos, controlando este intercambio a través de las IRQ, que producen una interrupción para indicar a la CPU que ha ocurrido un evento, por ejemplo, que ha llegado un dato, o que ha

cambiado el estado de algunas señales d entrada. La CPU debe responder a estas interrupciones lo más rápido posible, para que de tiempo a recoger el dato antes de que el siguiente lo sobrescriba. En el caso de la UART 16550A, ésta contiene unos *buffers* de datos de tipo FIFO (*primero en entrar primero en salir, First Input First Output*), dos de 16 bytes, donde se pueden guardar varios datos antes de que la CPU los recoja. Esto también disminuye el número de interrupciones por segundo generadas por el puerto serie.

El RS-232 puede transmitir los datos en grupos de 5, 6, 7 u 8 bits, a velocidades determinadas (normalmente, 9600 bits por segundo o más). Para la transmisión de los datos, se inicia con un bit de comienzo, que corresponde al valor binario 0, le sigue la información a transmitir, después un bit opcional de paridad (indica si el número de bits transmitidos es par o impar, para detectar fallos), y por último 1 o 2 bits de parada (stop). Normalmente, el protocolo utilizado es 8N1 (8 bits de datos, sin paridad y con 1 bit de stop). En la *figura 4.3* se presenta el diagrama de transmisión de un dato con formato 8N1. El receptor indica al emisor que puede enviarle datos activando la salida RTS. El emisor envía un bit de STAR (nivel alto) antes de los datos y un bit de STOP (nivel bajo) al final de éstos.

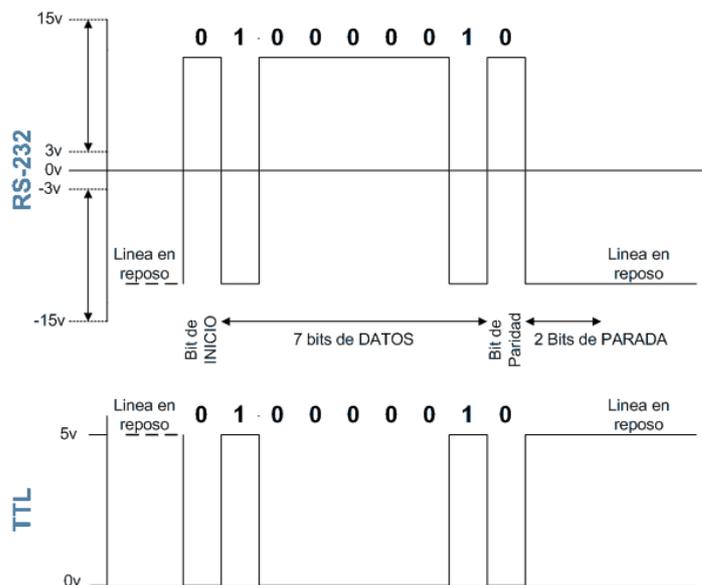


Figura 4.3 Transmisión de un dato con formato 8N1

Una vez que ha comenzado la transmisión de un dato. Los bits tienen que llegar uno detrás de otro a una velocidad constante y en determinados instantes. Por eso se dice que el RS-232 es asíncrono por carácter y síncrono por bit. Las terminales que portan los datos RXD y TXD. Las demás terminales se encargan de otros trabajos, por ejemplo: DTR indica que la computadora está encendida, DSR que el aparato conectado a dicho puerto está encendido, RTS que la computadora puede recibir datos (porque no está ocupado), CTS que el aparato conectado puede recibir datos, y DCD detecta que existe una comunicación (presencia de datos).

Tanto el dispositivo a conectar como la computadora (o el programa terminal) tienen que usar el mismo protocolo serie para comunicarse entre sí. Puesto que el estándar RS-232 no permite indicar en qué modo se está trabajando, es el usuario quien tiene que decidirlo y configurar ambas partes. Los parámetros configurables son: protocolo serie (8N1), velocidad del puerto serie y protocolo de control de flujo (*handshaking*). Este último puede ser por hardware (RTS/CTS) o bien por software (XON/XOFF), que son señales que tienen la capacidad de detener o reanudar el flujo de datos. Por ejemplo, para el caso XON/XOFF, si una computadora envía datos a una impresora, más rápido de lo que la impresora puede procesar. La impresora contiene un buffer donde los datos son almacenados para poder recibir los mismos a la velocidad de envío de la computadora. Si dicho buffer se llena, antes de que la impresora haya procesado todos los datos, un pequeño microprocesador en la misma envía una señal XOFF para que pare de enviar datos y evitar perderlos. Cuando el buffer se ha vaciado lo suficiente, entonces la impresora vuelve a enviar una señal XON para que continúe el envío. Cuando lo que se envía son datos binarios, no es recomendable utilizar el tipo de *handshaking* XON/XOFF, ya que es posible que las señales no se reconozcan, puesto que se contienen caracteres codificados.

La opción de velocidad del puerto especifica la velocidad máxima con la que los programas pueden transmitir datos, aunque, cuando es iniciada la transmisión de datos entre dispositivos éstos deben compartir dicha velocidad, para evitar errores

de transmisión. La mayoría de los programas que realizan comunicación de datos a través del puerto serial, establecen su propia velocidad, con lo que se limita la velocidad a la que pueden enviar los datos, un ejemplo es una hiperterminal de datos de Windows.

En el caso de la comunicación entre equipos PC, en la figura 2.4 se muestra como se deben conectar las diferentes terminales.

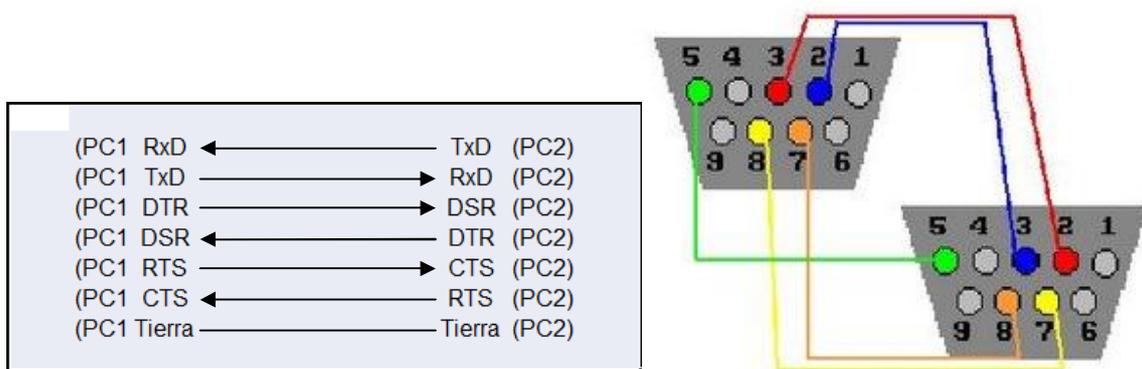


Figura 4.4 Conexión de terminales

4.2.2 USB

En general, para el manejo de dispositivos externos a una PC, por ejemplo, las tarjetas de adquisición de datos (TAD), se hace uso de las ranuras internas de ésta. Dichas ranuras se conocen como bus ISA (*Arquitectura Estándar de la Industria, Industry Estándar Architecture*) y PCI (*Interconexión de Componentes Periféricos, Peripheral Component interconnect*).

El puerto ISA fue creado por IBM en 1980 para ser empleado en las computadoras AT (Alta Tecnología) de IBM. La versión original del bus ISA se desarrolló para trabajar en 8 bits.

Posteriormente, se creó una extensión de 16 bits aumentando su velocidad (8MHz). Este extensión es compatible de forma descendente con el puerto ISA de 8 bits. El ancho de banda máximo del puerto ISA de 16 bits es de 16 MBytes/segundo. Este ancho de banda es insuficiente para las necesidades actuales, como tarjetas de video de alta resolución, por lo que el puerto ISA no se

emplea en los equipos de cómputo actuales. Este puerto ha sido sustituido por el puerto PCI.

El puerto PCI, es un puerto de la computadora que se utiliza para conectar dispositivos (tarjetas) periféricos a la tarjeta “madre” de la computadora. A diferencia de los puertos ISA el puerto PCI permite la configuración dinámica de un dispositivo periférico (Plug and Play). En el encendido de la computadora las tarjetas PCI y el sistema BIOS interactúan y negocian los recursos que son pedidos por las primeras. Esto permite la asignación de interrupciones y direcciones del puerto para las tarjetas PCI.

La conexión de las tarjetas de datos de E/S en los puertos ISA y PCI requiere que el usuario abra el gabinete de la computadora, inserte la tarjeta, ajuste los interruptores y puenteadores se asegure de que los cambios no estén en conflicto con otras tarjetas, cierre el gabinete y reinicie el equipo de cómputo. Este proceso es difícil para la mayoría de los usuarios que no están familiarizados con el hardware del sistema, ya que existe una probabilidad alta de cometer algún error. Además, el número de ranuras ISA y PCI es muy limitado (2 o 3 por lo regular), por ejemplo, para usuarios que requieren de varias aplicaciones en su equipo de cómputo, como puede ser tarjetas de: adquisición de datos, video de alta resolución, sonido modem, entre otros. Las tarjetas plug and play eliminan los cambios de interruptores y puenteadores, pero, aun así, el usuario tiene la necesidad de abrir la computadora para instalar la tarjeta, además, que el número de ranuras sigue siendo limitado.

Para resolver el problema mencionado, a mediados de la década de los noventa se reunieron representantes de siete compañías (Compaq, DEC, IBM, Intel, Microsoft, NEC y Northern Telecom), para diseñar una mejor manera de conectar dispositivos de E/S de baja velocidad a una computadora. Desde entonces, cientos de compañías más se le han unido. El estándar resultante se llama USB (Bus Serial Universal, Universal Serial Bus).

Algunos de los objetivos de las compañías que originalmente concibieron el USB e iniciaron el proyecto son los siguientes:

- Los usuarios no deben tener que ajustar interruptores ni puenteadores en las tarjetas o dispositivos.
- Los usuarios no deben tener que abrir el gabinete para instalar dispositivos de E/S nuevos.
- Sólo debe haber un tipo de cable que sirva para conectar todos los dispositivos.
- Los dispositivos de E/S deberán obtener su energía del cable.
- Se deberán poder conectar hasta 127 dispositivos a una sola computadora
- El sistema deberá apoyar dispositivos de tiempo real (sonido, teléfono, entre otros).
- Los dispositivos deberán poder instalarse mientras la computadora está funcionando.
- No se deberá requerir un reinicio después de instalar un nuevo dispositivo.
- El nuevo bus y sus dispositivos de E/S deberán ser de bajo costo

La primera especificación comercial de USB 1.1 fue liberada el 23 de septiembre de 1998. Un año después, USB era ya una interfaz común en la mayoría de los equipos de cómputo personal. El objetivo se cumplió, permitir que dispositivos de diversos fabricantes pudieran comunicarse entre sí en una arquitectura abierta, se diseñó para dispositivos de baja velocidad como teclados, ratones, cámaras de foto fija, digitalizadores de imágenes, teléfonos digitales, entre otros. Las velocidades de transferencia de este dispositivo es de 1.5 MBytes/s para dispositivos lentos y velocidad completa de 12 MBytes/s, para los dispositivos que necesiten mayor ancho de banda. En abril del 2009 se presentó el USB de alta velocidad o USB 2.0 que llega a 480 Mbytes/s. un dispositivo de alta velocidad USB 2.0 puede conectarse a un controlador USB 1.1, pero no operará en toda su capacidad. Por su parte, un dispositivo USB 1.1 puede conectarse a un controlador USB 2.0 a un máximo de 12 Mbytes/s. si la computadora tiene USB

1.1 y se desea migrar a USB 2.0, será necesario adquirir e instalar un nueva tarjeta controladora.

Un sistema USB consiste en una estructura jerárquica en el cual el modo principal se denomina eje raíz, donde éste es conectado a la línea de control y datos principal, es decir como si se conectará a un puerto PCI de la tarjeta madre. Este eje tiene zócalos para cables que se pueden conectar a dispositivos de E/S o ejes de expansión, a fin de tener más zócalos.

Cuando se conecta un nuevo dispositivo de E/S, el eje raíz detecta este suceso e interrumpe el sistema operativo. Luego este consulta al dispositivo para averiguar qué es y qué ancho de banda de USB necesita. Si el sistema operativo decide que hay suficiente ancho de banda para el dispositivo, asigna a este una dirección única (1-127) y coloca esta dirección y otra información en los registros de configuración dentro del dispositivo. De este modo es posible añadir dispositivos nuevos sobre la marcha, sin que el usuario tenga que configurarlos y sin tener que instalar nuevas tarjetas ISA o PCI. El sistema USB puede verse como un conjunto de canales bit que van del eje raíz a los dispositivos de E/S. Cada dispositivo puede dividir su canal de bits en, cuando más, 16 subcanales para diferentes tipos de datos (audio y video).

Dentro de cada canal o subcanal, fluyen datos del eje raíz al dispositivo o en la otra dirección.

No hay tráfico entre dos dispositivos de E/S.

Cada 1.00 ± 0.05 ms, el eje raíz transmite una nueva trama para mantener a todos los dispositivos sincronizados en el tiempo. Una trama se asocia a un canal de bits y consiste en paquetes, el primero de los cuales va del eje raíz al dispositivo. Los paquetes subsecuentes de la trama también podrían enviarse en esa dirección, pero también podrían regresar del dispositivo al eje raíz. En la figura 4.5 se presenta un ejemplo de lo que se acaba de comentar.

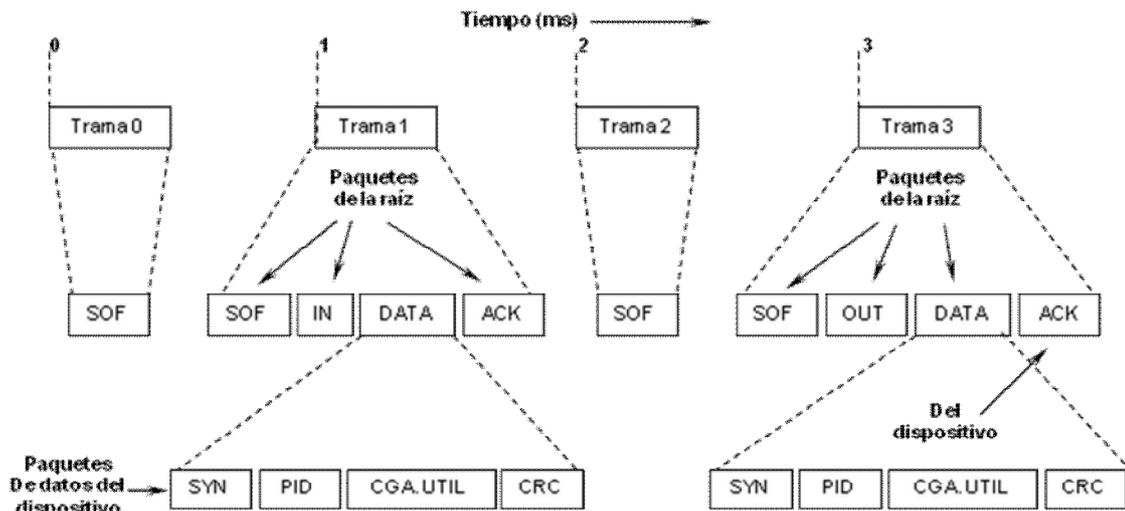


Figura 4.5 Flujo de datos en USB.

En la figura anterior, se puede observar que no existe trabajo que realizar en las tramas 0 y 2, así que todo lo que se necesita es un paquete SOF (Inicio de Trama, Start of Frame). Este paquete siempre se difunde a todos los dispositivos. La trama 1 es un escrutinio, por ejemplo, una solicitud a un scanner para que devuelva los bits que ha encontrado en la imagen que está digitalizando. La trama 3 consiste en datos que se envían a algún dispositivo, digamos una impresora. La interfaz USB reconoce cuatro clases de tramas: control, isocrónicas, volumen e interrupción.

Las tramas de control sirven para configurar dispositivos, para dar órdenes y preguntar por su estado. Las tramas isocrónicas son para dispositivos de tiempo real, como son: micrófonos, altavoces y teléfonos que necesitan enviar o aceptar datos a intervalos de tiempo precisos. Las tramas de control e isocrónicas tienen un retraso altamente predecible pero no ofrecen retransmisión en caso de ocurrir errores. Las tramas de volumen son para transferencias grandes de datos, hacia y desde dispositivos que no requieren tiempo real como las impresoras.

Por último, las tramas de interrupción son necesarias, porque USB no reconoce interrupciones.

Por ejemplo, en lugar de hacer que un teclado cause interrupción cada vez que se pulsa una tecla, el sistema operativo puede escrutarlo cada 50 ms para recolectar cualquier digitalización que esté pendiente.

Una trama consiste en uno o más paquetes de información. Existen cuatro tipos de paquetes de información: testigo, datos, saludo y especiales. Los paquetes de testigo que van de la raíz a un dispositivo periférico y sirven para el control del sistema. Los paquetes SOF, IN y OUT son paquetes del tipo testigo. El paquete SOF es el primero de cada trama y marca su principio. Si no hay trabajo que efectuar, el paquete SOF es el único de la trama. El paquete testigo IN es un escrutinio, que pide al dispositivo periférico devolver ciertos datos. Los campos del paquete IN indican cual es el canal de bits que se está escrutando para que el dispositivo sepa que datos debe devolver. El paquete de testigo OUT anuncia que a continuación vienen datos para el dispositivo periférico. Los paquetes de datos, DATA, sirven para transmitir hasta 64 bytes de información en cualquier dirección. En la figura 4.4 se muestra el formato de un paquete de datos (trama 1), como respuesta a un paquete testigo (trama SOF); el paquete de datos consiste en un campo de sincronización de 8 bits, un de paquete denominado PID (Campo de Identificador de Paquete, Packet Identifier Field) de 8 bits, la carga útil y en un código para detección de errores, CRC (Código de Redundancia Cíclica). En el caso del paquete de saludo se han definido tres clases de paquetes: ACK (el paquete de datos se recibió correctamente, Acknowledgement), NAK (se detecto un error de CRC) y STALL (ocupado).

Para el paquete especial se define el paquete PRE (para el tráfico en la línea de control y datos).

Una descripción a detalle de la figura 4.5, nos indicaría que cada 1.00 ms debe enviarse a una trama del eje raíz, incluso si no hay trabajo. Las tramas 0 y 2 consisten en un solo paquete SOF, lo que indica no hubo trabajo. La trama 1 es un escrutinio, de modo que principia con paquetes SOF e IN de la computadora al periférico de E/S, seguidos de un paquete DATA del periférico de E/S a la

computadora. El paquete ACK indica al dispositivo que los datos se recibieron correctamente. En caso de haber un error, se devolvería un NAK al dispositivo y el paquete se retransmitiría en el caso de datos de volumen. La trama 3 tiene una estructura similar a la 1, excepto que ahora los datos fluyen de la computadora al dispositivo de E/S.

Desde la perspectiva del usuario, los puertos e interfaces USB son muy sencillos de emplear, empezando por los cables, cuyos conectores sólo son de dos tipos e imposibles de colocar de manera errónea. Cuando un dispositivo nuevo USB se asocia a una computadora el sistema operativo detecta su presencia e instala el controlador a dicho dispositivo, o bien, puede solicitar al usuario el disco de instalación de ese periférico. Después de trabajar con un dispositivo, el usuario puede desconectarlo directamente del puerto USB, sin riesgo de perder la configuración o dañar el dispositivo.

El cable USB consta de cuatro hilos (figura 4.6), los cuales son: dos para datos, uno para alimentación (+5 V) y uno para tierra, dichos hilos están rodeados de una capa de blindaje para evitar interferencias. El sistema de señalización transmite un 0 lógico como una transmisión de voltaje y 1 lógico como la ausencia de transmisión de voltaje, de modo que las series largas de ceros generan una serie de pulsaciones regular.

Cuando se deben conectar más dispositivos de los que ya ocupan los puertos USB, es indispensable usar un concentrador USB (Hub). El concentrador amplía la cantidad de puertos disponibles para otros dispositivos. Una sola computadora, combinando cables de no más de 5 metros de longitud cada uno y concentradores, pueden tener asociados hasta 127 dispositivos USB.

El USB utiliza un nuevo tipo de cable pequeño, sencillo, económico y fácil de conectar. Los cables tienen diferentes conectores en el extremo del eje y en el extremo del dispositivo, para evitar que las personas conecten accidentalmente un zócalo del eje a otro; únicamente existe un tipo de cable, pero con dos tipos de

conectores diferentes: A y B. Los primeros presentan las cuatro terminales correspondientes a los cuatro hilos alineados en un plano, mientras que los conectores B presentan los contactos distribuidos en dos planos paralelos, dos en cada plano.

La longitud del cable no debe exceder de 5 metros, aunque es posible encadenar 5 cables periféricos USB y cuatros Hubs donde se puede obtener de 25 metros.

Terminal	Nombre	Descripción
1	VBUS	+5 VDC
2	D-	Data -
3	D+	Data +
4	GND	Ground

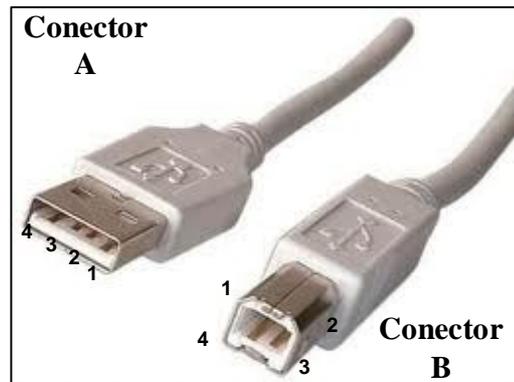


Figura 4.6 Cable USB y conectores (A y B)

El USB utiliza la decodificación de datos NRZI (Non Return to Zero, Invert on ones), se mantiene constante el nivel de tensión durante la duración de un bit. Los datos se decodifican mediante la presencia o ausencia de una transición de la señal al principio del intervalo de duración de un bit. Un uno se codifica la transición (bajo a alto o alto bajo) al principio del intervalo de señalización, mientras que un cero se representa por la ausencia de transición.

NRZI es un ejemplo de codificación diferencial. En la codificación diferencial, en lugar de determinar el valor absoluto, la señal se codifica en función de los cambios entre los elementos de señales adyacentes. En términos generales, la codificación de cada bit se hace de la siguiente manera: si se trata del valor binario cero, se codifica que la misma señal que el bit anterior; si se trata de un valor binario uno, entonces se codifica con una señal diferente que la utilizada que el bit precedente, como se muestra en la figura 4.7. Una ventaja de este esquema es

que en presencia de ruido puede ser más seguro detectar una transición en lugar de comparar un valor con un umbral.

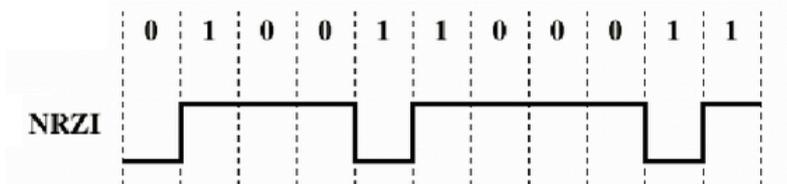


Figura 4.7 Formato de codificación NRZI

Los voltajes de operación del USB son: 4.75 a 5.25 voltios en el puerto, considerando un nivel bajo menor de 0.8 voltios y un nivel alto mayor de 2.0 voltios.

A continuación mencionaremos otros protocolos de comunicación de forma general, sin profundizar mucho en ellos, ya que para el presente trabajo no son de utilidad, sin embargo es justo mencionarlos dada su importancia en lo que a comunicaciones alámbricas e inalámbricas se refiere.

4.2.3 Modem.

Con el fin de lograr que cualquier computadora se pueda comunicar con otras computadoras, la EIA creó una interfaz denominada RS232. Las características principales de esta norma son: que cualquier computadora que reconozca la interfaz RS232 se pueda comunicar con alguna otra computadora que de igual forma reconozca dicha interfaz; también son definidas las características tales como: el tamaño del cable, la forma física del conector, los niveles de voltajes, así como las terminales correspondientes de conector.

Una limitante de la interfaz RS232 es la distancia de conexión, de acuerdo a la norma es de 15m. Una manera de resolver este problema es emplear una conexión a través del sistema telefónico. Desafortunadamente, el sistema

telefónico no es capaz de transmitir señales requeridas por la norma RS232 y es necesario insertar un dispositivo denominado modem (modulador – demodulador) entre la computadora y el teléfono y también entre la terminal y el teléfono, con el fin de efectuar la conversión de señales, es decir, la modulación transforma la señal digital a binaria en analógica. La demodulación transforma la señal analógica en digital

4.3 Comunicación Inalámbrica

La comunicación inalámbrica o sin cables es aquella en la que los extremos de la comunicación (emisor/receptor) no se encuentran unidos por un medio de propagación físico, sino que se utiliza la modulación de ondas electromagnéticas a través del espacio. En este sentido, los dispositivos físicos sólo están presentes en los emisores y receptores de la señal, entre los cuales encontramos: antenas, computadoras portátiles, teléfonos móviles, etc.

4.3.1 Radio Modem.

Como se mencionó anteriormente, un modem se encarga de convertir un flujo de datos digitales en banda base a una señal analógica apropiada para ser transmitida sobre el medio y viceversa. La principal diferencia entre un radio modem y un modem de cable se refiere a la aplicación a la que se destina. De este modo, los modems de cable están preparados para conectarse a redes de cable como pueden ser la red telefónica conmutada o una red híbrida de fibra óptica y coaxial. Los radio modems están destinados a aplicaciones en las cuales es necesario transmitir la señal vía radio, por ejemplo interconexión de ordenadores a través de LAN (red de área local, Local Area Network) o MAN (red de área metropolitana, Metropolitan Area Network) inalámbricas, envío y recepción de mensajes, telemetría, localización automática de vehículos etc.

Los radio modems están preparados para transmitir sobre un entorno más hostil que el cable, a menudo sujeto a desvanecimientos, propagación multicambio (multipath) o interferencias.

Para la transmisión de información, los radio modems disponibles comercialmente suelen utilizar las bandas ISM (Industrial Scientific and Medical) de 900 MHz (902-928 MHz), 2.4 GHz (2400-2483.5 MHz) y 5.8 GHz (5725-5850 MHz).

Los radio modems actuales están mejorados para funcionar correctamente, en aspectos como: bandas de frecuencia; tolerancia de frecuencia y seguimiento de la portadora; potencia transmitida y margen dinámico; ecualización; efecto multicambio y esquemas de modulación.

4.3.2 IrDA

Los infrarrojos fueron descubiertos en 1800 por William Herschel, un astrónomo inglés de origen alemán. Herschel colocó un termómetro de mercurio en el espectro obtenido por un prisma de cristal con el fin de medir el calor emitido por cada color. Descubrió que el calor era más fuerte al lado del rojo del espectro y observó que allí no había luz. Esta es la primera experiencia que muestra que el calor puede transmitirse por una forma invisible de luz. Herschel denominó a esta radiación "rayos calóricos", denominación bastante popular a lo largo del siglo XIX que, finalmente, fue dando paso al más moderno de radiación infrarroja.

Los primeros detectores de radiación infrarroja eran bolómetros, instrumentos que captan la radiación por el aumento de temperatura producido en un detector absorbente.

Los infrarrojos se utilizan en los equipos de visión nocturna cuando la cantidad de luz visible es insuficiente para ver los objetos. La radiación se recibe y después se refleja en una pantalla. Los objetos más calientes se convierten en los más luminosos.

La radiación térmica o infrarroja es un tipo de radiación electromagnética de mayor longitud de onda que la luz visible, pero menor que las microondas, tienen longitudes de onda que van aproximadamente de 1mm a 7×10^6 m, ésta última corresponde a la longitud de onda más grande de la luz visible. Los infrarrojos se subdividen en infrarrojos cortos (0.7-5 μm), infrarrojos medios (5-30 μm) e infrarrojos largos (30-1000 μm).

Un uso muy común es el que hacen los comandos a distancia (telecomandos o mando a distancia) que generalmente utilizan los infrarrojos en vez de ondas de radio ya que no interfieren con otras señales como las señales de televisión. Los infrarrojos también se utilizan para comunicar a corta distancia las computadoras con sus periféricos. Los aparatos que utilizan este tipo de comunicación cumplen generalmente con el estándar IrDA (Infrared Data Association.)

4.3.3 Blue Tooth

Bluetooth es una especificación industrial para Redes Inalámbricas de Área Personal (WPANs) que posibilita la transmisión de voz y datos entre diferentes dispositivos mediante un enlace por radiofrecuencia en la banda ISM de los 2,4 GHz. Los principales objetivos que se pretenden conseguir con esta norma son:

- Facilitar las comunicaciones entre equipos móviles y fijos.
- Eliminar cables y conectores entre éstos.
- Ofrecer la posibilidad de crear pequeñas redes inalámbricas y facilitar la sincronización de datos entre equipos personales.

Los dispositivos que con mayor frecuencia utilizan esta tecnología pertenecen a sectores de las telecomunicaciones y la informática personal, como PDA, teléfonos móviles, computadoras portátiles, ordenadores personales, impresoras o cámaras digitales.

Se denomina Bluetooth al protocolo de comunicaciones diseñado especialmente para dispositivos de bajo consumo, con una cobertura baja y basados en transceptores de bajo costo.

Gracias a este protocolo, los dispositivos que lo implementan pueden comunicarse entre ellos cuando se encuentran dentro de su alcance. Las comunicaciones se realizan por radiofrecuencia de forma que los dispositivos no tienen que estar alineados y pueden incluso estar en habitaciones separadas si la potencia de transmisión lo permite. Estos dispositivos se clasifican como "Clase 1", "Clase 2" o

"Clase 3" (tabla 4.1) en referencia a su potencia de transmisión, siendo totalmente compatibles los dispositivos de una clase con los de las otras.

Clase	Potencia máxima permitida (mW)	Potencia máxima permitida (dBm)	Rango (aproximado)
Clase 1	100 mW	20 dBm	~100 metros
Clase 2	2.5 mW	4 dBm	~10 metros
Clase 3	1 mW	0 dBm	~1 metro

Tabla 4.1 Clasificación de dispositivos

En la mayoría de los casos, la cobertura efectiva de un dispositivo de clase 2 se extiende cuando se conecta a un transceptor de clase 1. Esto es así gracias a la mayor sensibilidad y potencia de transmisión del dispositivo de clase 1, es decir, la mayor potencia de transmisión del dispositivo de clase 1 permite que la señal llegue con energía suficiente hasta el de clase 2. Por otra parte la mayor sensibilidad del dispositivo de clase 1 permite recibir la señal del otro pese a ser más débil.

Los dispositivos con Bluetooth también pueden clasificarse según su ancho de banda como se muestra en la tabla 4.2

Versión	Ancho de banda
Versión 1.2	1 Mbit/s
Versión 2.0 + EDR	3 Mbit/s
Versión 3.0 + HS	24 Mbit/s

Tabla 4.2 Clasificación por ancho de banda

Bluetooth SIG (Special Interest Group) está formado por un grupo de compañías de diferentes áreas de la industria (e.g. telecomunicaciones, redes, computación) entre las que se encuentran 3Com, Ericsson, IBM, Intel, Agere, Microsoft, Motorola, Nokia, Toshiba, así como más de 100 asociados y otras compañías promotoras de la tecnología. El nuevo estándar permitirá una mayor validez y soporte en el mercado de las especificaciones de Bluetooth, además es un

recurso adicional para aquellos que implementen dispositivos basados en esta tecnología.

4.3.4 WI-FI

Wi-Fi (pronunciado en español /wifi/ y en inglés /waifai/) es una marca de la *Wi-Fi Alliance* (anteriormente la *WECA: Wireless Ethernet Compatibility Alliance*), la organización comercial que adopta, prueba y certifica que los equipos cumplen los estándares 802.11 relacionados a redes inalámbricas de área local.

Nokia y Symbol Technologies crearon en 1999 una asociación conocida como WECA (Wireless Ethernet Compatibility Alliance, Alianza de Compatibilidad Ethernet Inalámbrica). Esta asociación pasó a denominarse Wi-Fi Alliance en 2003. El objetivo de la misma fue crear una marca que permitiese fomentar más fácilmente la tecnología inalámbrica y asegurar la compatibilidad de equipos.

De esta forma, en abril de 2000 WECA certifica la interoperabilidad de equipos según la norma IEEE 802.11b, bajo la marca Wi-Fi. Esto quiere decir que el usuario tiene la garantía de que todos los equipos que tengan el sello Wi-Fi pueden trabajar juntos sin problemas, independientemente del fabricante de cada uno de ellos. Se puede obtener un listado completo de equipos que tienen la certificación Wi-Fi en Alliance - Certified Products.

En el año 2002 la asociación WECA estaba formada ya por casi 150 miembros en su totalidad.

La norma IEEE 802.11 fue diseñada para sustituir el equivalente a las capas físicas y MAC de la norma 802.3 (Ethernet). Esto quiere decir que en lo único que se diferencia una red Wi-Fi de una red Ethernet es en cómo se transmiten las tramas o paquetes de datos; el resto es idéntico. Por tanto, una red local inalámbrica 802.11 es completamente compatible con todos los servicios de las redes locales (LAN) de cable 802.3 (Ethernet).

Uno de los problemas a los cuales se enfrenta actualmente la tecnología Wi-Fi es la progresiva saturación del espectro radioeléctrico, debido a la masificación de usuarios, esto afecta especialmente en las conexiones de larga distancia (mayor de 100 metros). En realidad Wi-Fi está diseñado para conectar ordenadores a la

red a distancias reducidas, cualquier uso de mayor alcance está expuesto a un excesivo riesgo de interferencias.

Un muy elevado porcentaje de redes son instalados sin tener en consideración la seguridad convirtiendo así sus redes en redes abiertas (o completamente vulnerables a los crackers), sin proteger la información que por ellas circulan.

Existen varias alternativas para garantizar la seguridad de estas redes. Las más comunes son la utilización de protocolos de cifrado de datos para los estándares Wi-Fi como el WEP, el WPA, o el WPA2 que se encargan de codificar la información transmitida para proteger su confidencialidad, proporcionados por los propios dispositivos inalámbricos. La mayoría de las formas son las siguientes:

- WEP, cifra los datos en su red de forma que sólo el destinatario deseado pueda acceder a ellos. Los cifrados de 64 y 128 bits son dos niveles de seguridad WEP. WEP codifica los datos mediante una “clave” de cifrado antes de enviarlo al aire. Este tipo de cifrado no está muy recomendado, debido a las grandes vulnerabilidades que presenta, ya que cualquier cracker puede conseguir sacar la clave.
- WPA: presenta mejoras como generación dinámica de la clave de acceso. Las claves se insertan como de dígitos alfanuméricos, sin restricción de longitud
- IPSEC (túneles IP) en el caso de las VPN y el conjunto de estándares IEEE 802.1X, que permite la autenticación y autorización de usuarios.
- Filtrado de MAC, de manera que sólo se permite acceso a la red a aquellos dispositivos autorizados. Es lo más recomendable si solo se va a usar con los mismos equipos, y si son pocos.
- Ocultación del punto de acceso: se puede ocultar el punto de acceso (Router) de manera que sea invisible a otros usuarios.
- El protocolo de seguridad llamado *WPA2* (estándar 802.11i), que es una mejora relativa a WPA. En principio es el protocolo de seguridad más seguro para Wi-Fi en este momento. Sin embargo requieren hardware y software compatibles, ya que los antiguos no lo son.

Sin embargo, no existe ninguna alternativa totalmente fiable, ya que todas ellas son susceptibles de ser vulneradas.

Existen varios dispositivos Wi-Fi, los cuales se pueden dividir en dos grupos: **Dispositivos de Distribución o Red**, entre los que destacan los routers, puntos de acceso y Repetidores; y **Dispositivos Terminales** que en general son las tarjetas receptoras para conectar a la computadora personal, ya sean internas (tarjetas PCI) o bien USB.



Figura 4.8 Router WI-FI

- Dispositivos de Distribución o Red:
 - Los puntos de acceso son dispositivos que generan un "set de servicio", que podría definirse como una "Red Wi-Fi" a la que se pueden conectar otros dispositivos. Los puntos de acceso permiten, en resumen, conectar dispositivos en forma inalámbrica a una red existente. Pueden agregarse más puntos de acceso a una red para generar redes de cobertura más amplia, o conectar antenas más grandes que amplifiquen la señal.
 - Los router inalámbricos son dispositivos compuestos, especialmente diseñados para redes pequeñas (hogar o pequeña oficina). Estos dispositivos incluyen, un Router (encargado de interconectar redes, por ejemplo, nuestra red del hogar con internet), Un punto de acceso (explicado más arriba) y generalmente un switch que permite conectar algunos equipos vía cable. Su tarea es tomar la conexión a internet, y brindar a través de ella acceso a todos los equipos que conectemos, sea por cable o en forma inalámbrica.
 - Los Repetidores Inalámbricos son equipos que se utilizan para extender la cobertura de una red inalámbrica, éstos se conectan a una red existente que

tiene señal más débil y crean una señal limpia a la que se pueden conectar los equipos dentro de su alcance.

- Los dispositivos terminales abarcan tres tipos mayoritarios: tarjetas PCI, tarjetas PCMCIA y tarjetas USB:



Figura 4.9 Tarjeta USB para WI-FI

- Las tarjetas PCI para Wi-Fi se agregan (o vienen de fábrica) a los ordenadores de sobremesa. Hoy en día están perdiendo terreno debido a las tarjetas USB. Dentro de este grupo también pueden agregarse las tarjetas MiniPCI que vienen integradas en casi cualquier computador portátil disponible hoy en el mercado.
- Las tarjetas PCMCIA son un modelo que se utilizó mucho en los primeros ordenadores portátiles, aunque están cayendo en desuso, debido a la integración de tarjeta inalámbricas internas en estos ordenadores. La mayor parte de estas tarjetas solo son capaces de llegar hasta la tecnología B de Wi-Fi, no permitiendo por tanto disfrutar de una velocidad de transmisión demasiado elevada
- Las tarjetas USB para Wi-Fi son el tipo de tarjeta más común que existe en las tiendas y más sencillo de conectar a un pc, ya sea de sobremesa o portátil, haciendo uso de todas las ventajas que tiene la tecnología USB. Hoy en día puede encontrarse incluso tarjetas USB con el estándar 802.11N (Wireless-N) que es el último estándar liberado para redes inalámbricas.

- También existen impresoras, cámaras Web y otros periféricos que funcionan con la tecnología Wi-Fi, permitiendo un ahorro de mucho cableado en las instalaciones de redes y especialmente, gran movilidad.

Las redes Wi-Fi poseen una serie de ventajas, entre las cuales podemos destacar:

- Al ser redes inalámbricas, la comodidad que ofrecen es muy superior a las redes cableadas porque cualquiera que tenga acceso a la red puede conectarse desde distintos puntos dentro de un rango suficientemente amplio de espacio.
- Una vez configuradas, las redes Wi-Fi permiten el acceso de múltiples ordenadores sin ningún problema ni gasto en infraestructura, no así en la tecnología por cable.
- La Wi-Fi Alliance asegura que la compatibilidad entre dispositivos con la marca *Wi-Fi* es total, con lo que en cualquier parte del mundo podremos utilizar la tecnología Wi-Fi con una compatibilidad total.

Pero como red inalámbrica, la tecnología Wi-Fi presenta los problemas intrínsecos de cualquier tecnología inalámbrica. Algunos de ellos son:

- Una de las desventajas que tiene el sistema Wi-Fi es una menor velocidad en comparación a una conexión con cables, debido a las interferencias y pérdidas de señal que el ambiente puede acarrear.
- La desventaja fundamental de estas redes existe en el campo de la seguridad. Existen algunos programas capaces de capturar paquetes, trabajando con su tarjeta Wi-Fi en modo promiscuo, de forma que puedan calcular la contraseña de la red y de esta forma acceder a ella. Las claves de tipo WEP son relativamente *fáciles de conseguir* con este sistema. La alianza Wi-Fi arregló estos problemas sacando el estándar WPA y posteriormente WPA2, basados en el grupo de trabajo 802.11i. Las redes protegidas con WPA2 se consideran robustas dado que proporcionan muy buena seguridad. Este problema se agrava si consideramos que no se puede controlar el área de cobertura de una conexión, de manera que un

receptor se puede conectar desde fuera de la zona de recepción prevista (desde fuera de una oficina, desde una vivienda colindante).

- Hay que señalar que esta tecnología no es compatible con otros tipos de conexiones sin cables como Bluetooth, Infrarrojo, radio modem, etc.

4.4 Conclusión capítulo 4.

Como se vio a lo largo del capítulo 4, hoy en día contamos con una variedad de formas de transmisión de datos, las hay por medio de cable o inalámbricas, decidir cual usar dependerá de varios factores como lo son: el costo, la distancia, la velocidad de transmisión, el tipo de información a transmitir, la tecnología a la que tengamos acceso, entre otros. Decidir que tecnología es la más adecuada para el problema que se nos planteé, requerirá del estudio de los factores ya mencionados, sin perder de vista el objetivo y utilizar el que de la solución más eficiente.

CAPÍTULO 5 DISEÑO Y DESARROLLO DE LA INTERFAZ DE SOFTWARE DEL SISTEMA DE CONTROL Y MONITOREO (SCM).

5.1 Interfaz

Ya que tanto en este capítulo, como en el capítulo 6 hablamos de interfaz de software y de hardware respectivamente, conviene dar una pequeña definición de este término muy usado en lo que a computación se refiere, con la finalidad de establecer a que nos referimos cuando mencionamos la interfaz.

Interfaz es la conexión entre dos computadoras o máquinas de cualquier tipo dando una comunicación entre distintos niveles. Interfaz también hace referencia al conjunto de métodos para lograr interactividad entre un usuario y una computadora.

Una interfaz puede ser del tipo GUI (Graphical User Interface, interface gráfica de usuario), también puede ser a partir de un hardware. Es decir toda conexión y/o interacción entre hardware, software y usuario es una interfaz.

Así pues podemos decir que la interfaz de usuario, es el vínculo entre el usuario y el programa de computadora, pero también hablamos de una interfaz cuando nos referimos a la comunicación que se establece entre dos computadoras o máquinas electrónicas a través de algún dispositivo.

5.1.1 Interfaz gráfica de usuario (GUI).

Tal vez cuando oímos la palabra interfaz, de inmediato pensamos en una interfaz grafica de usuario, ya que generalmente es con la que todos estamos familiarizados, sin la necesidad de saber absolutamente nada de computación, ya que hoy en día la modernidad de nuestras vidas nos hace interactuar con un sinfín de máquinas electrónicas a través de una interfaz grafica de usuario, ejemplos de estos sobran, como manejar un cajero automático, los iconos, ventanas y menús que vemos en pantalla son la interfaz grafica de usuario que permite comunicarnos con la máquina, para retirar dinero o hacer una consulta.

Cuando queremos hacer una llamada, escuchar música, tomar un video, o acceder a internet desde un teléfono celular con touch screen, tenemos que interactuar con la electrónica del teléfono a través de su interfaz gráfica de usuario.

Las interfaces gráficas de usuario, se han convertido en estándar en los materiales computacionales, el hardware y el software se convierten en simples herramientas sobre los cuales fue construida la interfaz.

5.1.2 Interfaz Hardware.

Aunque el usuario no la ve, debido a que no interactúa directamente con ella y pocos saben cómo funciona, la interfaz de hardware, es tan, o más importante que la interfaz de usuario, en pocas palabras podemos definir la interfaz de hardware como los dispositivos electrónicos que nos ayudan a interconectarnos con otros dispositivos.

Sin esta interfaz de hardware, la conexión y/o comunicación entre equipos sería literalmente imposible, cabe mencionar que la conexión y/o comunicación entre dispositivos puede ser por cable o inalámbrica.

Dentro de las interfaces que utilizan cable, tenemos las que nos ayudan a comunicarnos a través de puerto serie, paralelo, USB y Ethernet.

Las interfaces inalámbricas son aquellas en las que se utilizan dispositivos que logran la conexión y/o comunicación a distancia sin necesidad de cables, utilizando ondas electromagnéticas y tecnologías como: radiofrecuencia, infrarrojo, bluetooth, y Wi-Fi.

5.2 Programa seleccionado

Para el desarrollo de la interfaz de usuario, de este proyecto, decidí usar C# (“C Sharp”).

5.2.1 C# (“C Sharp”).

C# es un lenguaje de programación orientado a objetos de escritura segura, que es simple pero poderoso, permitiendo a los programadores construir una amplia gama de aplicaciones.

Visual C# forma parte de Visual Studio, el cual es un Ambiente de Desarrollo Integrado (IDE por sus siglas en inglés) que asiste a los desarrolladores en la creación de programas en varios lenguajes, poniendo a su disposición un gran número de herramientas invaluableles en la programación.

La sintaxis y estructuración de C# es muy similar a la de C++, ya que la intención de Microsoft con C# es facilitar la migración de códigos escritos en estos lenguajes a C# y facilitar su aprendizaje a los desarrolladores habituados a ellos. Sin embargo, su sencillez y el alto nivel de productividad son equiparables a los de Visual Basic.

Las Express Editions (ediciones expresas) de Visual Studio son una extensión de su línea de productos que incluyen herramientas simples, divertidas y fáciles de aprender, enfocadas a desarrolladores no profesionales como estudiantes o aficionados a la programación que deseen construir aplicaciones Windows dinámicas.

El desarrollo de Software en Microsoft Visual C# Express Edition es completamente gratuito sin necesidades de licencia o contratos.

En resumen, C# es un lenguaje de programación que toma las mejores características de lenguajes preexistentes como Visual Basic, Java o C++ y las combina en uno solo. El hecho de ser relativamente reciente no implica que sea inmaduro, pues su compilador es el más depurado y optimizado de los incluidos en el *.NET Framework*⁹ *SDK*¹⁰.

5.2.2 Instalando Microsoft Visual C# Express Edition.

Para instalar Visual C# Express Edition, hice una descarga gratuita desde el sitio web de Microsoft <<http://www.microsoft.com/express/Windows/>>. Existen varias versiones para diferentes sistemas operativos e idiomas. En este caso seleccione

⁹ NET Framework, es un proyecto de Microsoft para crear una nueva plataforma de desarrollo de software con énfasis en transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones

¹⁰ SDK kit de desarrollo de software o SDK (siglas en inglés de *Software Development Kit*)

la versión 2010 para Windows XP, pero se puede seleccionar la que más convenga, según las características del equipo con que contemos.

Una vez descargado el instalador, simplemente lo ejecuté y seguí las instrucciones en pantalla. Una instalación rápida por defecto fue más que suficiente para una instalación exitosa.

Ya finalizada la instalación, procederemos a ejecutar Visual C# Express por primera vez. Se pedirá un registro gratuito en línea, el cual es indispensable para el uso del software. Finalmente, el programa requerirá realizar configuraciones adicionales antes de poder ser utilizado.

5.3 Configurando el proyecto

Para poder obtener todos los beneficios de Visual C# Express, es indispensable que todos nuestros programas formen parte de un proyecto.

Los proyectos son un conjunto de parámetros y configuraciones que definen a nuestros programas y facilitan el paso de parámetros al compilador, el cual suele ser un proceso tedioso utilizando métodos convencionales, sobre todo si se trata de programas muy grandes.

Lo único que debemos hacer para definir un nuevo proyecto es:

a) Desde la pantalla de bienvenida al inicio del programa, dar clic en el vínculo “New Project...” en el área superior izquierda, figura 5.1

b) Alternativamente, podemos encontrar la misma opción en el menú “File -> New Project...” en la esquina superior izquierda de la pantalla o utilizar la tecla rápida Ctrl + Shift + N.

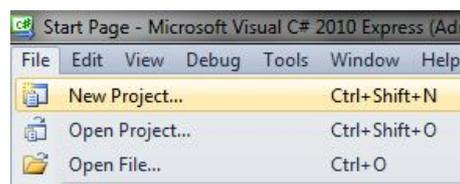
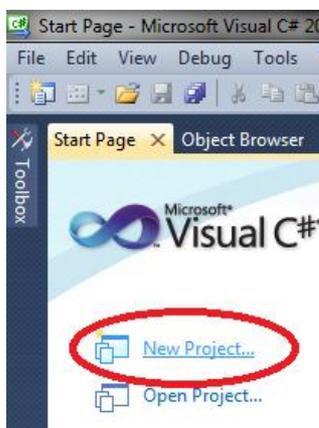


Figura 5.1 Nuevo proyecto

A continuación aparecerá la pantalla de configuración de

nuevos proyectos, en la cual haremos las siguientes selecciones, para seguir con la configuración de nuestro proyecto, figura 5.2

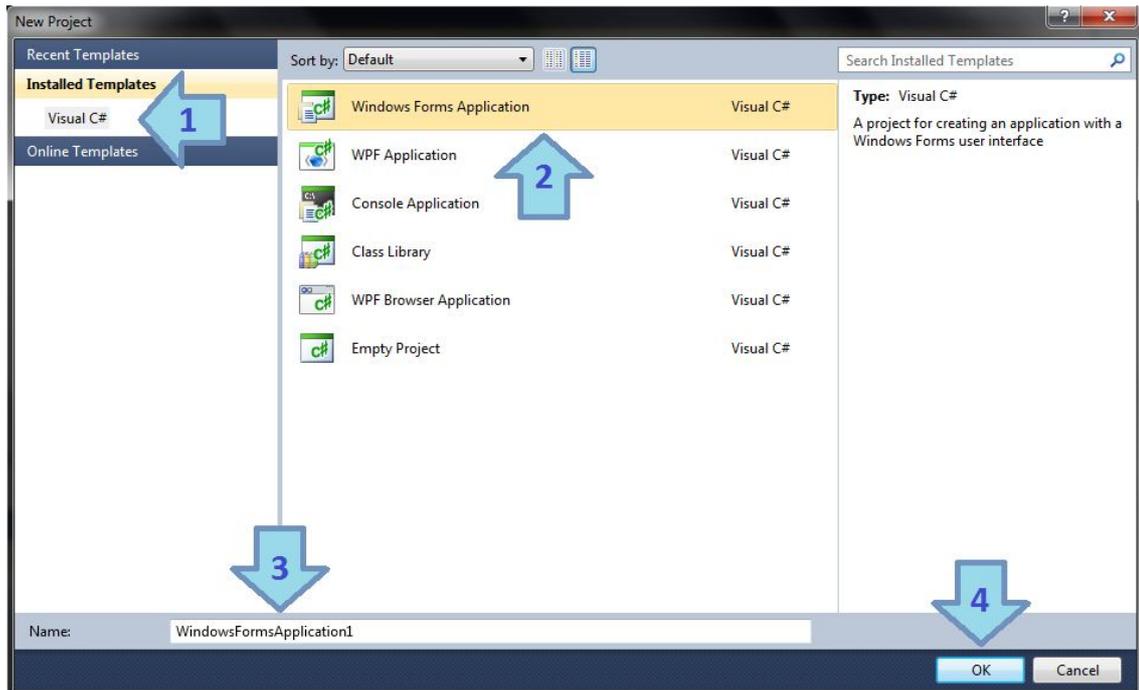


Figura 5.2 Configuración

1. Seleccionar una colección de plantillas, en este caso “Visual C#”.
2. Seleccionar tipo de proyecto, para este proyecto “Windows Forms Application”.
3. Especificar un nombre para el proyecto, en este caso lo nombramos comunicación.
4. Dar clic en el botón “OK” para finalizar.

Visual C# configurará automáticamente el proyecto con los parámetros necesarios para compilar programas de Windows Forms correctamente.

Finalmente, para construir y probar nuestro proyecto, basta con hacer clic en el ícono *Start Debugging* (Iniciar Depuración, tecla de acceso rápido “F5”) figura 5.3, localizado en el área superior de la interfaz. Realizar esta operación será necesario cada vez que modifiquemos nuestro proyecto para poder ver los cambios reflejados en nuestro archivo ejecutable, el cual, como en la mayoría de los compiladores, podemos encontrar en la subcarpeta bin\Release (o Debug,

dependiendo de los parámetros de compilación) dentro de la carpeta de nuestro proyecto.

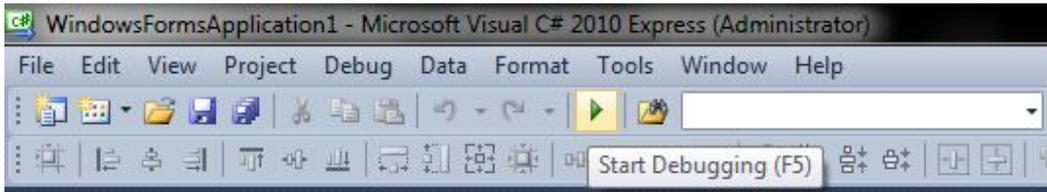


Figura 5.3 Start debugging (F5)

5.4 Interfaz gráfica visual.

Ya que hemos creado un nuestro proyecto de Windows Forms, se nos presentará la interfaz de diseño, figura 5.4, que cuenta con varias ventanas, cada una con diferente función.

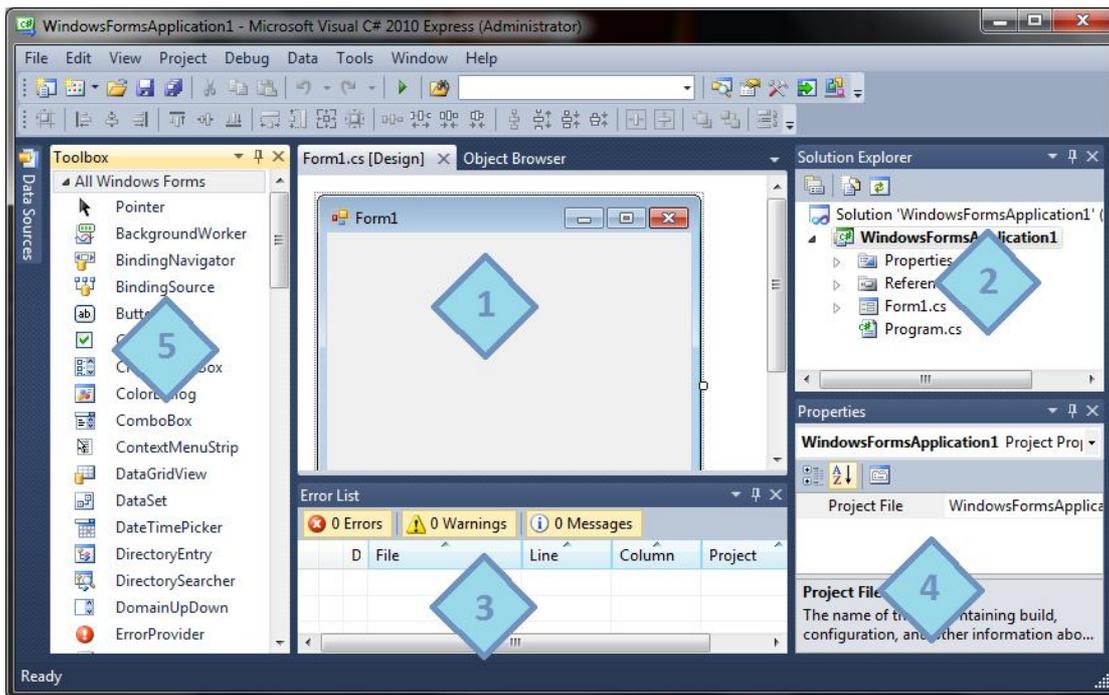


Figura 5.4 Interfaz de diseño

Para poder trabajar de forma óptima y correcta nuestro proyecto al menos tenemos que tener a la vista las siguientes ventanas.

1. Design: Área de Diseño. En esta área es donde podemos posicionar los controles sobre nuestra ventana. Si lo deseamos, también podemos consultar el código fuente generado desde aquí.

2. Solution Explorer: *Explorador de Soluciones.* Aquí podemos visualizar todos los archivos que forman parte de nuestro proyecto, así como agregar nuevos o eliminar los ya existentes.

3. Error List: *Lista de Errores.* Esta sección nos mostrará cualquier problema con nuestro programa al momento de compilar, desde simples sugerencias hasta errores fatales durante la ejecución del programa.

4. Properties: *Ventana de Propiedades.* Nos muestra las propiedades de los controles que tengamos seleccionados en el Área de Diseño, de tal forma que podamos modificarlos fácilmente.

5. Toolbox: *Caja de Herramientas.* Desde esta ventana podemos seleccionar los controles que queramos incluir en nuestro formulario, simplemente arrastrándolo hacia el Área de Diseño

Finalmente, como en la mayoría de las aplicaciones de desarrollo en Windows, en el área superior de la interfaz, podemos encontrar diversos menús y botones de acceso rápido para el manejo de nuestros proyectos, ya sea Guardar / Abrir proyectos o modificar su configuración actual, entre otras cosas, figura 5.5

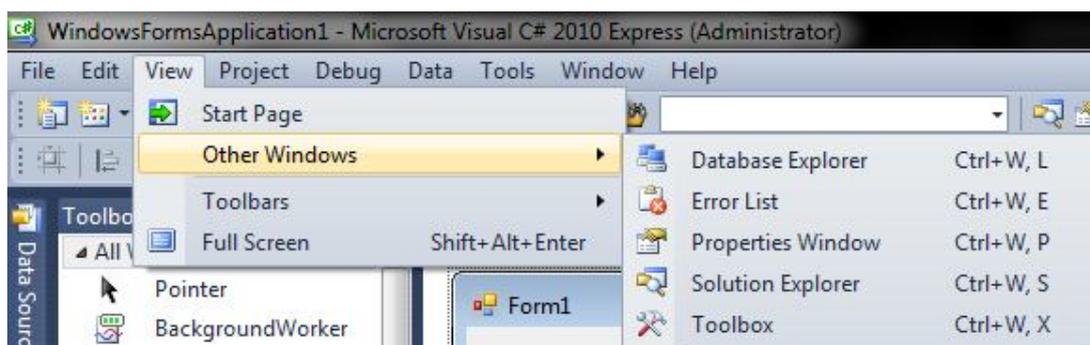


Figura 5.5 Menus y botones

Si no se muestran algunos de los elementos anteriores de la interfaz gráfica, es posible que se encuentren ocultos. Para visualizarlos nuevamente, basta con dirigirse al menú “View -> Other Windows” y seleccionar la ventana deseada

5.5 Desarrollo de la Interfaz gráfica para el control de temperatura.

La interfaz grafica se diseño siguiendo los siguientes pasos:

1. Creamos un Nuevo Proyecto de Windows Forms en C#.
2. Añadimos los Componentes necesarios a nuestro Form.
 - a) Para este caso particular, necesitaremos cuatro *TextBox* una para cada campo de información acerca de los parámetros necesarios para controlar los tanques de maduración. Las *TextBox* son *Componentes* que nos permiten introducir cadenas de texto directamente, modificando la propiedad *Text* de las mismas en el proceso, con la cual podremos trabajar directamente utilizando código C# en un futuro.

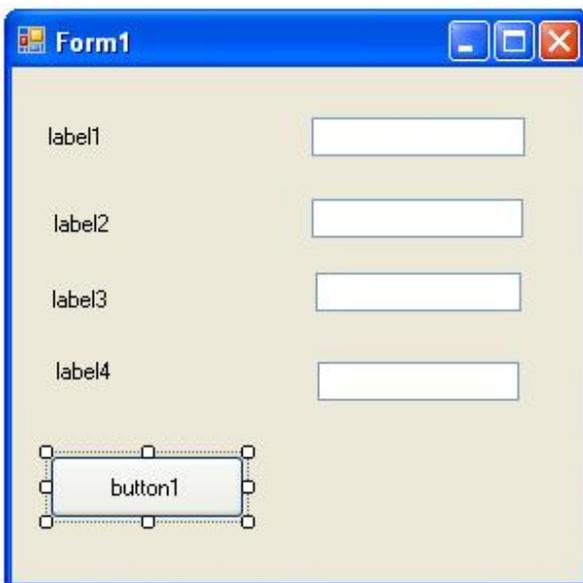


Figura 5.6 Diseñando la interfaz

- b) También será necesario incluir un *Button*, el cual cumpliría con la función de conectarnos al puerto serial y de enviar los datos capturados en los *TextBox* al PIC y de aquí al control de temperatura. Los *Buttons* son *Componentes* que nos indican que es posible realizar una interacción con el programa dando clic izquierdo sobre los mismos.

- c) Finalmente, agregaremos cuatro *Labels*, con el objetivo de ilustrar el funcionamiento de nuestras *TextBox*. y un *SerialPort* para poder establecer la comunicación con el circuito o alambrado, figura 5.6.

3. Ahora modificamos los Componentes según sea necesario

Una vez posicionados los *Componentes* a nuestro agrado, es necesario modificar sus propiedades para asegurarnos de que cumplen correctamente con su función, figura 5.7.

- a) Establecer la propiedad *Text* de *label1* como “No Tanque”.
 - b) Establecer la propiedad *Text* de *label2* como “Producto”.
 - c) Establecer la propiedad *Text* de *label3* como “Temperatura °C”.
 - d) Establecer la propiedad *Text* de *label4* como “Tiempo Hrs”.
 - e) Establecer la propiedad *Text* de *button1* como “Encendido”.
- Establecer la propiedad *Text* de *Form1* como “Control de Tanques”.

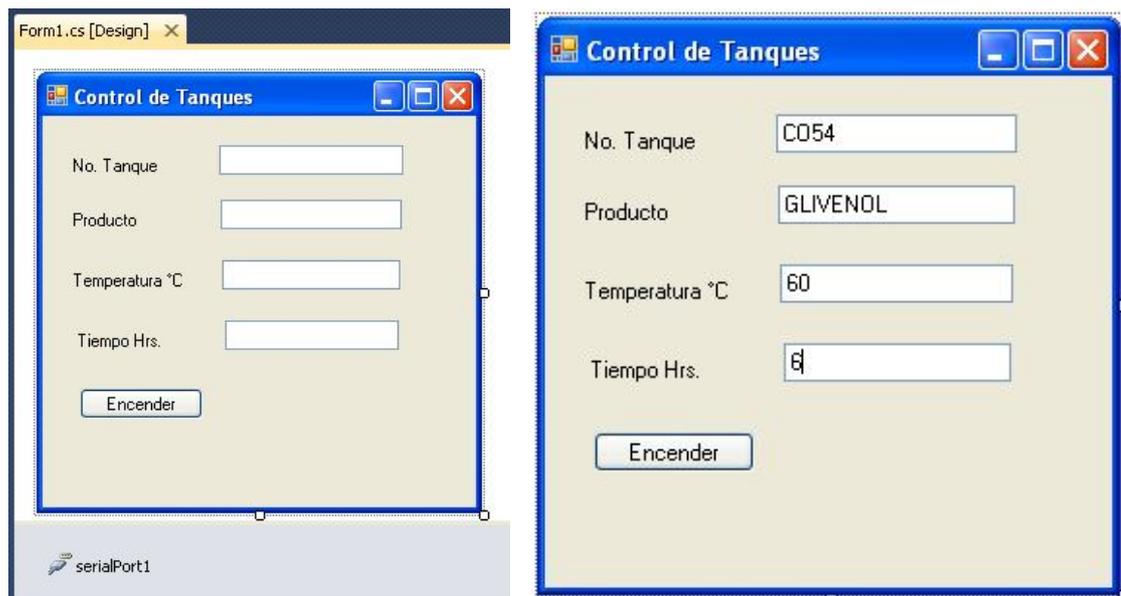


Figura 5.7 Modificando propiedades.

Una vez que hemos ejemplificado como se hizo la pantalla, donde capturaremos los datos para el control del tanque, diseñaremos una pantalla de bienvenida al sistema de control y una pantalla de monitoreo, figura 5.8

Una vez que tenemos todos los componentes de nuestra interfaz gráfica y hemos modificado sus propiedades, comenzaremos a añadirle funcionalidad, es decir a programar los componentes que sean necesarios.

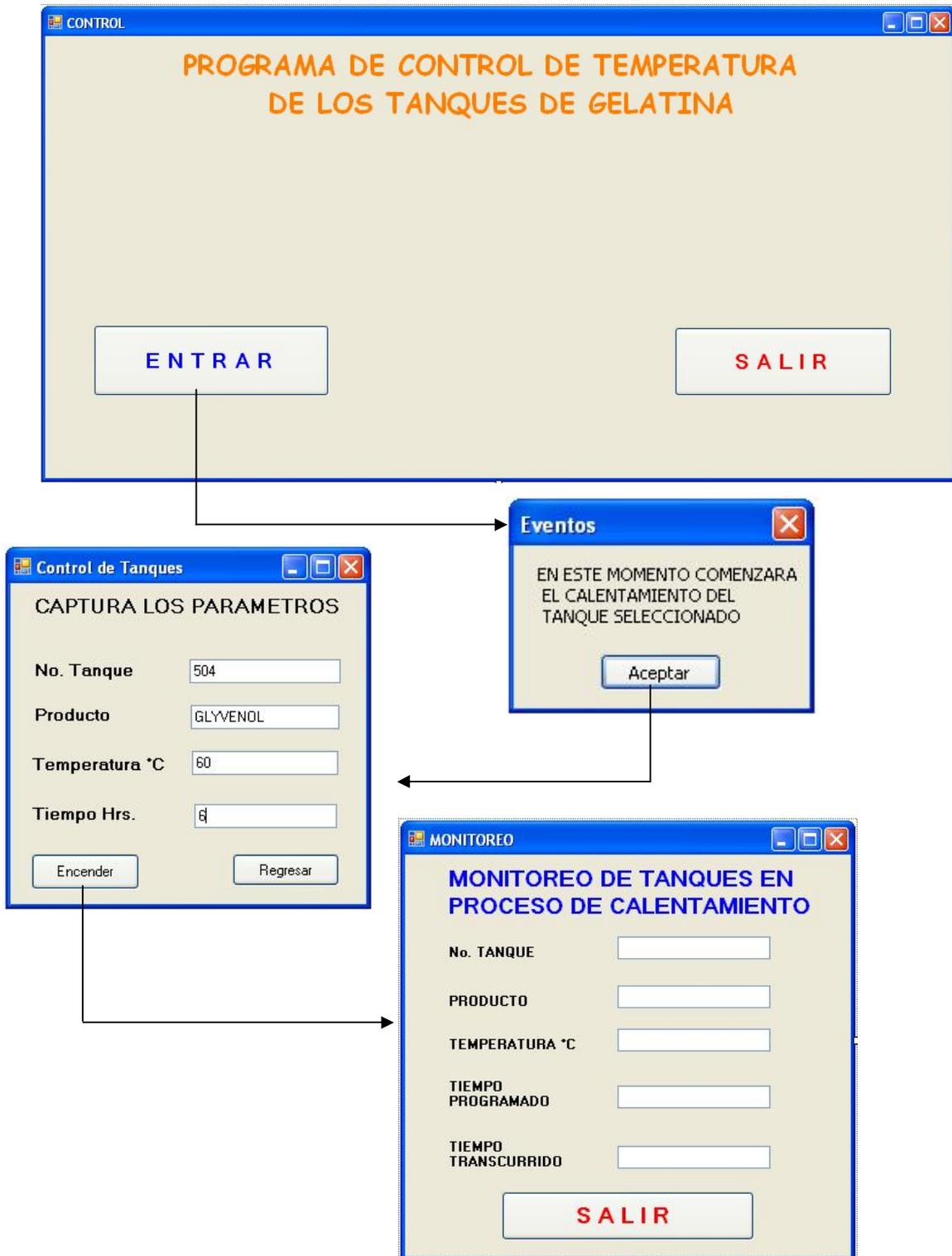


Figura 5.8 Pantallas de inicio, captura y monitoreo

CAPÍTULO 6 DISEÑO Y DESARROLLO DE LA INTERFAZ EN HARDWARE DEL SISTEMA DE CONTROL Y MONITOREO (SCM).

6.1 Componentes del circuito.

Para diseñar el circuito con el que comunicamos la PC con el control de temperatura, y a su vez con la computadora, se escogió el PIC18F2550, debido a que presenta las características ideales para trabajar con el MAX232, y para la comunicación por medio de USB, con la PC.

La comunicación entre el control de temperatura y el circuito que se diseñó, se realizó a través del conector serial DB-9 (con el protocolo de comunicación RS-232), utilizando el circuito MAX232, esto debido a que el control de temperatura sólo cuenta con este tipo de interfaz para llevar a cabo la comunicación. En el mercado no se encuentran disponibles controles de temperatura con interfaz USB, esto debido a que la única información que maneja un control de temperatura es únicamente la temperatura, por lo que la comunicación serial RS232 es la adecuada, por precio y por el número de datos que puede transmitir un control de temperatura, utilizar una interfaz USB en este tipo de instrumentos los encarecería considerablemente y no tendría ningún beneficio o mejora hacerlo, pues se estarían desperdiciando todos los recursos que ofrece la interfaz USB. Haciendo una analogía, sería como tener un auto formula 1 para recorrer las calles del centro de la ciudad.

Para comunicar el circuito con la PC utilizamos una interfaz USB a partir del PIC18F2550, que entre sus características cuenta con Interfaz USB 2.0 de alta velocidad 12Mbit/s, voltaje de operación: 2 V a 5.5 V, y también una interfaz para comunicación serial RS232.

6.1.1 MAX232.

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.

El MAX232 soluciona la conexión necesaria para lograr comunicación entre el puerto serie y cualquier otro circuito con funcionamiento en base a señales de nivel TTL/CMOS figura 6.1. El circuito integrado posee dos convertidores de nivel TTL a RS232 y otros dos que, a la inversa, convierten de RS232 a TTL. Estos convertidores son suficientes para manejar las cuatro señales más utilizadas de cualquier puerto serie, que son TX, RX, RTS y CTS.

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

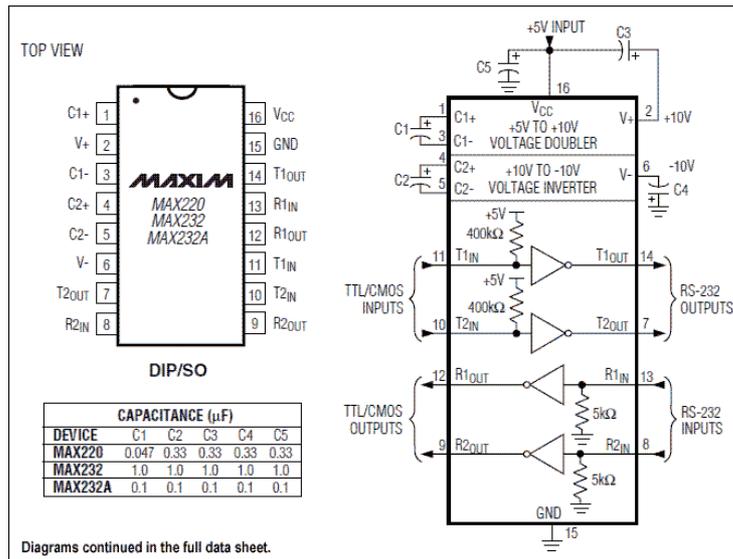


Figura 6.1 Esquema del circuito MAX232

6.1.2 PIC18F2550

El microcontrolador PIC18F2550 de Microchip pertenece a una familia de microcontroladores de prestaciones medias/altas de 8 bits (bus de datos) que tienen las siguientes características generales:

- Arquitectura Harvard
- Tecnología RISC
- Tecnología CMOS

Estas características se conjugan para lograr un dispositivo altamente eficiente en el uso de la memoria de datos y programa y por lo tanto en la velocidad de ejecución. Este PIC cuenta con dos pines para la interfaz o comunicación serial, RS232, figura 6.2, el pin 17 RC6/TX/CK, y el pin 18 RC7/RX/DT/SDO, también cuenta con 2 puertos o pines especiales para la interfaz USB, el pin 15 RC4/D-/VM y el pin 16 RC/D+/VP, lo cual lo hace ideal para la comunicación USB con cualquier otro dispositivo

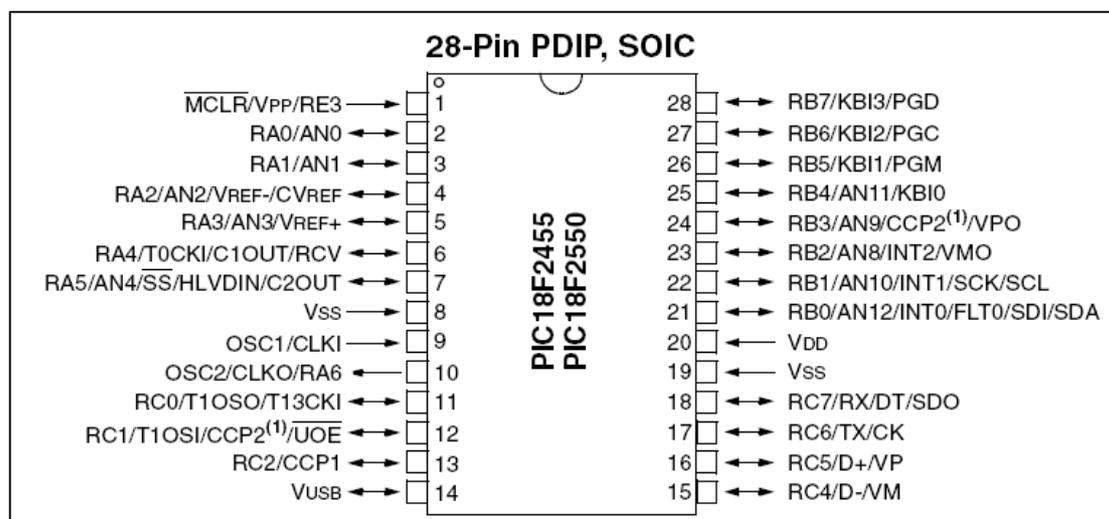


Figura 6.2 Distribución de puertos del PIC18F2550

6.2 Armado del circuito

Básicamente el armado del circuito se dividió en 2 partes, la primera parte fue la comunicación entre el PIC18F2550 y el control de temperatura utilizando los pines 17(TX) y 18(RX) con el circuito MAX232, la segunda parte fue la comunicación entre la computadora y el PIC18F2550 por medio de la interfaz USB utilizando los pines 15(D-) y 16(D+).

6.2.1 Alambrado puerto serial.

Para lograr la comunicación entre el control de temperatura y el PIC18F2550, se alambró la primera parte del circuito en una protoboard usando el MAX232 y un conector DB9 haciendo la conexión entre los pines 17 (TX) y 18 (RX) del PIC18F2550 y los pines 11 (T1in) y 12 (R1out) del circuito MAX232, como se

muestra en la figura 6.3., a su vez del MAX232 se conectaron los pines 13 (R1in) y 14 (T1out) a las terminales 2 (RxD) y 3 (TxD) respectivamente, del puerto serial DB9

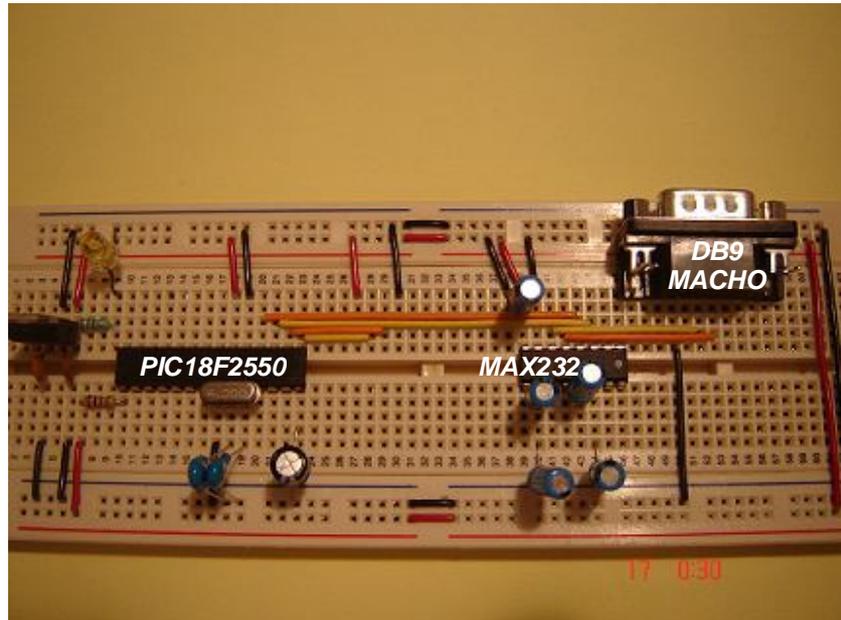


Figura 6.3 Protoboard conexión serial

6.2.2 Alambrado puerto USB.

En lo que respecta a la comunicación entre la computadora y el PIC18F2550, usamos la interfaz USB, esto lo logramos conectando los pines 15(D-) y 16(D+) del PIC, con las respectivas terminales 2(D-) y 3(D+) del cable USB, figura 6.4, las terminales 1 y 4 se conectan a voltaje y tierra respectivamente.

Terminal	Nombre	Descripción
1	VBUS	+5 VDC
2	D-	Data -
3	D+	Data +
4	GND	Ground



Figura 6.4 Terminales del conector USB

El alambrado en la protoboard de la conexión USB queda como se muestra en la figura 6.5, donde podemos apreciar las conexiones del PIC18F2550, D- y D+ con las terminales 2 y 3 del cable USB que corresponden al conector USB.

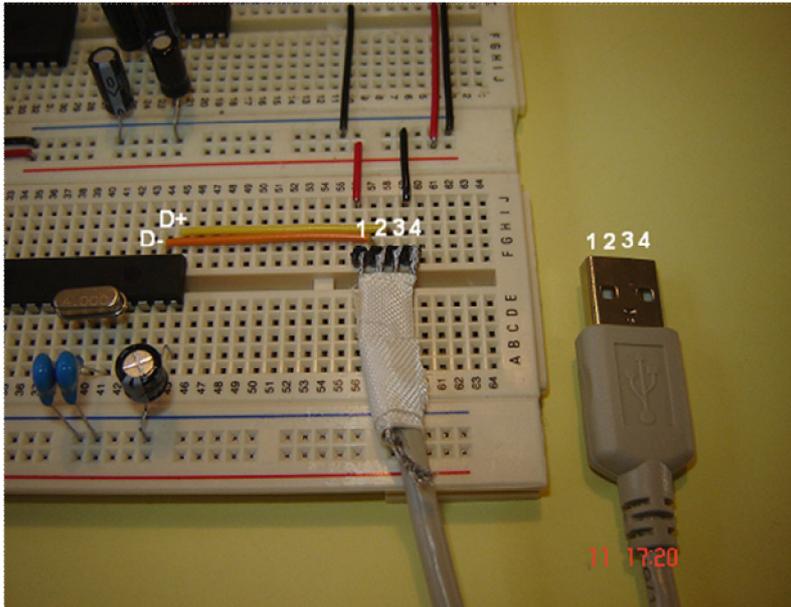


Figura 6.5 Protoboard conexión USB

Una vez realizadas las interfaces serial y USB, procedimos a agregar los demás componentes necesarios para el funcionamiento del PIC18F2550 y del MAX232

6.2.3 Alambrado MAX232

Para alambrear el MAX232 y que funcione de forma correcta, se necesitaron 5 capacitores dieléctricos de $1\mu\text{F}$, distribuidos como se muestra en la figura 6.6, además de sus conexiones a tierra y voltaje.

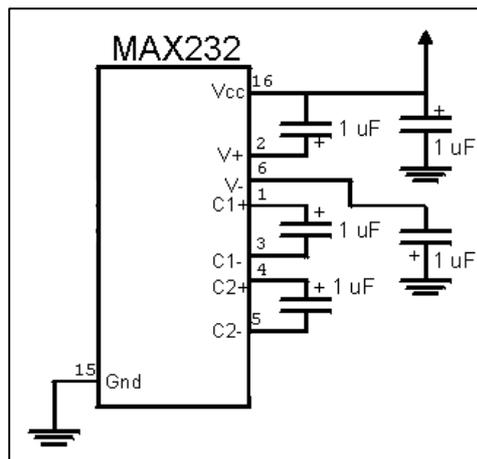


Figura 6.6 distribución de componentes del MAX232

6.2.4 Alambrado PIC18F2550.

Para alambrear el PIC18F2550 se necesitan los siguientes componentes:

1 oscilador de 4MHz, 1 resistencia de 10K Ω , 2 capacitores cerámicos de 22pF y 1 capacitor dieléctrico de 47 μ F, distribuidos como se muestra en la figura 6.7.

La resistencia de 10 K Ω se conecta al pin 1 (MCLR), el oscilador de 4 MHz se conecta a los pines 9(OSC1/CLK) y 10(OSC2/CLKOUT) y cada pin del oscilador se conecta a su vez a los capacitores cerámicos de 22pF y estos a tierra, el capacitor dieléctrico de 47 μ F se conecta al pin 14(VUSB) y a tierra

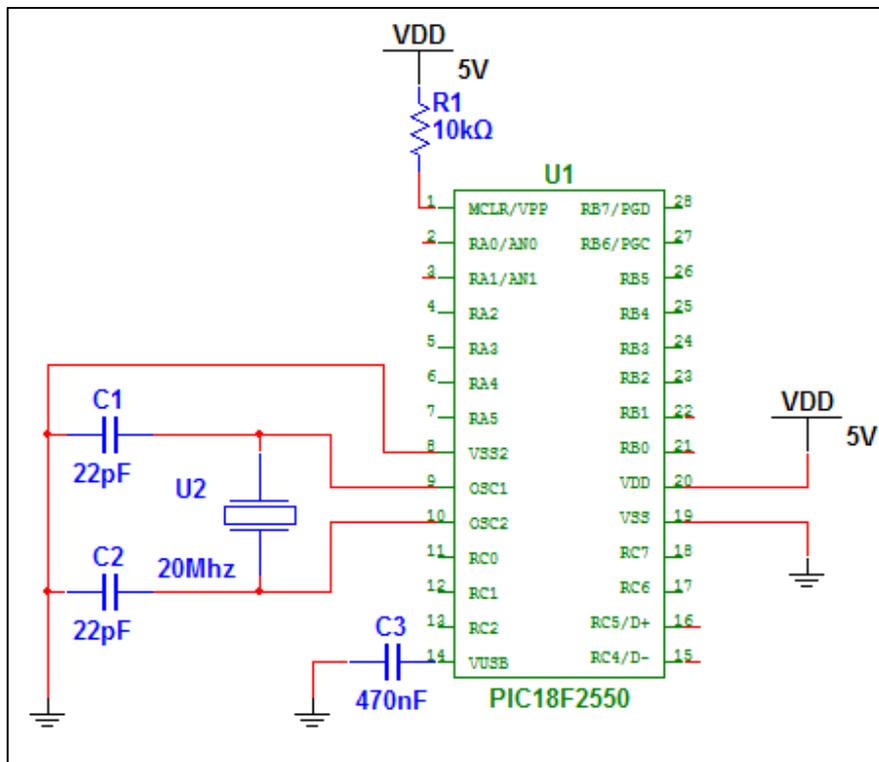


Figura 6.7 Alambrado PIC18F2550

6.2.5 Circuito armado.

Con la parte eléctrica y la de señales de transmisión conectadas tanto al MAX232, como al PIC18F2550 y a sus respectivas interfaces USB y puerto serie DB-9, figura 6.8, el circuito está casi terminado, ahora sólo falta programar el PIC18F2550.

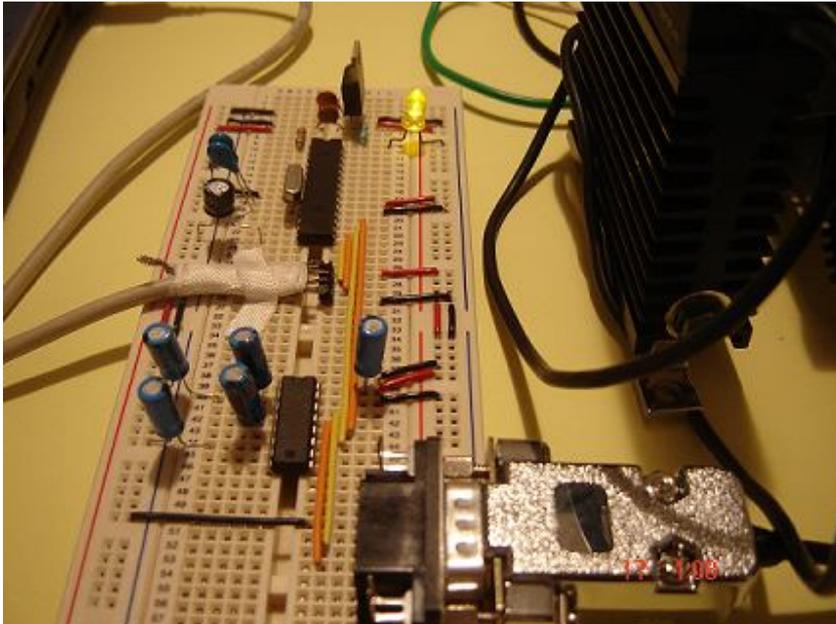


Figura 6.8 Circuito armado en protoboard (interfaz hardware)

6.3 Programando el PIC18F2550

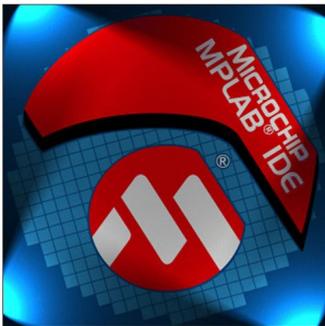
Para lograr que el PIC funcione de acuerdo a lo que queremos que realice, es necesario programarlo, es decir introducir el programa con las instrucciones del trabajo que va a realizar, para esto Microchip nos proporciona una herramienta muy valiosa y gratuita llamada MPLAB IDE, con la cual desarrollaremos un proyecto y dentro de éste escribiremos el programa fuente que regirá nuestro microcontrolador, alterno a este proyecto simularemos, el circuito, para asegurar antes de alambrear, que tanto el programa como el circuito funcionan de forma correcta, una vez completados estos pasos, nos auxiliaremos de un grabador de PIC's, para descargar el programa fuente en el microcontrolador y poder pasar a la fase de alambreado del circuito.

6.3.1 MPLAB-IDE V8.63 (Ambiente de Desarrollo Integrado).

Un *Ambiente de Desarrollo Integrado* o IDE (*Integrated Development Environment*) por sus siglas en inglés, es un sistema en software que permite la administración de múltiples herramientas como compiladores, simuladores y programadores, así

como de archivos de código fuente, bibliotecas de código y documentación, para la gestión de proyectos o soluciones desde una misma interfaz de usuario.

Para desarrollar soluciones tecnológicas con microcontroladores de Microchip podemos obtener de manera gratuita el sistema MPLAB IDE en su última versión 8.63. MPLAB IDE será nuestra herramienta de batalla para desarrollar aplicaciones con microcontroladores PIC así que vale la pena entender cómo



funcionan las características básicas de la interfaz de usuario.

MPLAB IDE, es una Plataforma de Desarrollo Integrada bajo Windows, con múltiples prestaciones, que permite escribir el programa para los PIC en lenguaje ensamblador o en lenguaje C, para desarrollar este proyecto lo escribí en lenguaje C. Para escribir un programa, hay que seguir una serie de pasos sencillos que se describen a continuación.

Paso 1. Abrir el programa MPLAB IDE.

Nos recibirá la primer pantalla (Untitled Workspace, espacio de trabajo sin título)

Paso 2. Abrir el asistente de Proyectos.

Seleccione el comando *Project Wizard* del menú *Project*. Aparecerá un cuadro de diálogo dándole la bienvenida al asistente, dar clic en siguiente.

Paso 3. Seleccionar un dispositivo.

En el paso 1 del asistente seleccione el modelo PIC18F2550, dar clic en siguiente.

Paso 4. Seleccionar el compilador.

Seleccione *CCS C Compiler for PIC10/12/14/16/18/24/dsPIC30/dsPIC33* como herramienta activa (*Active Toolsuit*), de no estar activados los componentes (*Toolsuit Contects*) será necesario buscarlos en la ruta de instalación del programa *C:\Archivos de programa\PICCC\Ccsc.exe*, dar clic en siguiente.

Paso 5. Indicar el nombre del proyecto.

Se nombró el proyecto como *Plantilla usb*, para poder tener una mejor administración de nuestros proyectos creé una carpeta con la ruta *C:\Documents and Settings\Usuario\Mis documentos\Plantilla usb*, donde estará alojado todo lo que respecte al programa, dar clic en siguiente.

Paso 6. Agregar archivos existentes.

En este caso no existen archivos previos que agregar a nuestro proyecto, simplemente se pasa al siguiente cuadro de diálogo.

Paso 7. Revisar opciones y generar el proyecto.

Se revisa que las opciones sean las indicadas y se genera el proyecto, de ser correctas dar clic en *Finalizar*.

Paso 8. Visualizar el administrador de archivos.

Ejecute el comando *Project* del menú *View* para visualizar la ventana de administración de archivos.

Paso 9. Agregar un nuevo archivo de código fuente.

Se ejecuta el comando *Add New File to Project*, del menú *Project*, se nombra el archivo como *main.c* y se guarda en la ruta *C:\Documents and Settings\Usuario\Mis documentos\Plantilla usb*.

Paso 10. Capturar el código fuente.

Se puede capturar directamente aquí el código fuente, o copiar del procesador de texto donde se haya elaborado y copiar aquí, al terminar se guarda con el comando *Save* del menú *File*. Figura 6.9

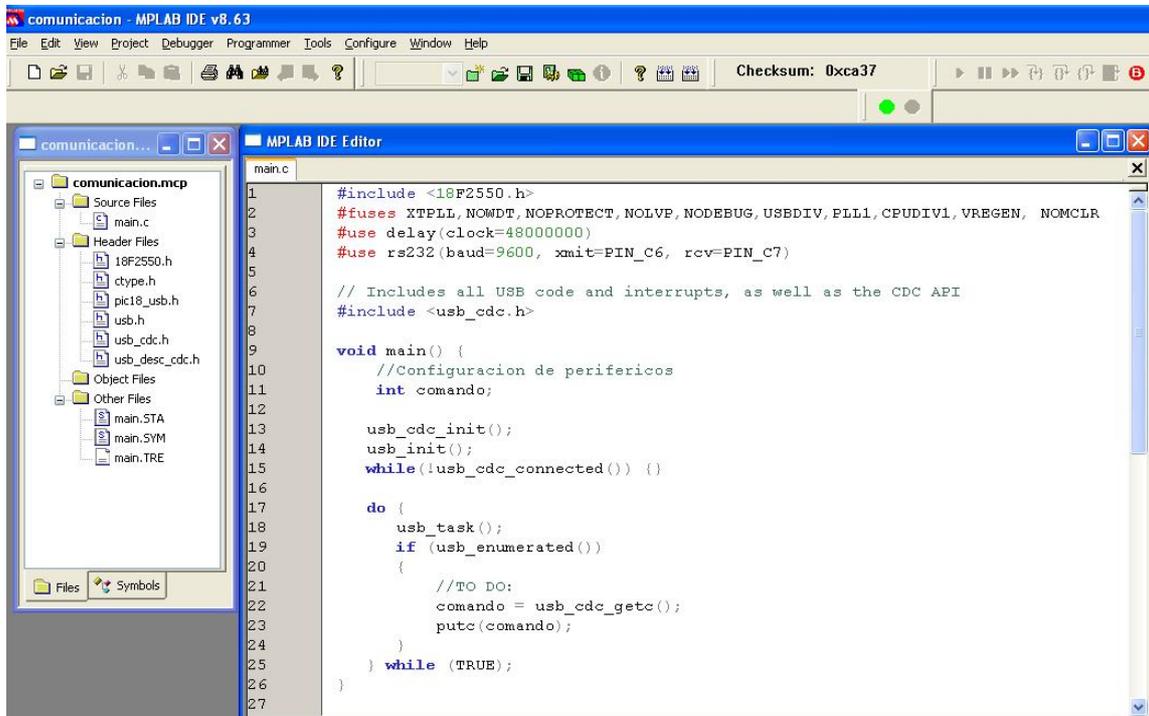


Figura 6.9 Editor de MPLAB-IDE

Paso 11. Compilar el código fuente.

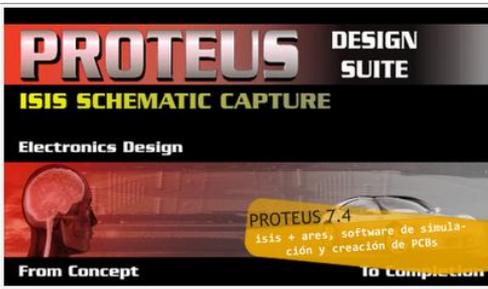
Se ejecute el comando *Build All* del menú *Project*. Si la compilación termina sin errores a parecerá en la ventana *Output* la leyenda *BUILD SUCCEEDED*, en caso contrario se indicarán en esta misma ventana los errores que no permitieron la compilación, se revisa, si es el caso, el código y se corrige cualquier error.

Paso 12. Obtener el archivo de código máquina.

Busque el archivo *main.c* dentro de la misma carpeta donde se generó el archivo de proyecto, este archivo será programado en un microcontrolador en una etapa siguiente.

6.3.2 Proteus 7.4

Proteus es un software de diseño electrónico desarrollado por Labcenter Electronics que consta de tres potentes subentornos como son ISIS para el diseño gráfico, VSM (Virtual System Modelling) para la simulación y ARES para el diseño de placas, para desarrollar mi simulación use VSM e ISIS.



Mediante ISIS podemos diseñar el circuito que deseemos con componentes muy variados, desde una simple resistencia hasta un microcontrolador, incluyendo fuentes de alimentación, generadores de señales y muchas otras prestaciones. Los diseños

realizados en ISIS pueden ser simulados en tiempo real, sin embargo la simulación la efectué abriendo VSM desde MPLAB, aplicación con la cual pude simular, en tiempo real, utilizando nuevamente la herramienta MPLAB.

ISIS es la herramienta ideal para una rápida realización de complejos diseños de esquemas electrónicos destinados tanto a la construcción de equipos electrónicos como a la realización de tareas de simulación y prueba, es muy fácil de trabajar, solo basta buscar los componentes del circuito en el menú de modos de trabajo y después en selección de bibliotecas damos click y aparecen los dispositivos a seleccionar para ir armando el circuito en la pantalla del entorno de trabajo del programa ISIS, figura 6.10

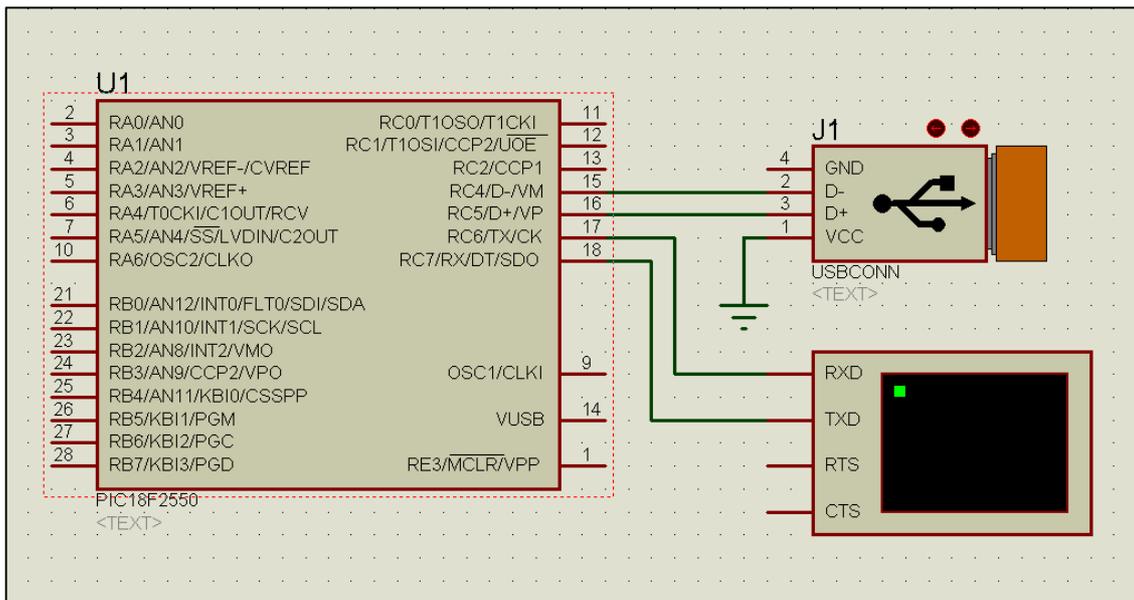


Figura 6.10 Circuito para control de temperatura cableado

De esta forma está listo nuestro circuito para la simulación usando VSM desde MPLAB-IDE.

6.3.2.1 Simulación con VSM (Virtual System Modelling, Modelado de Sistema Virtual)

Una vez que hemos realizado el programa y diseñado el gráfico del circuito, procederemos a llevar a cabo la simulación, esto nos ahorrará tiempo y nos asegurará antes de alambrear nuestro circuito que funcionará de forma correcta.

La forma de hacerlo es la siguiente:

1. Buscamos nuestro proyecto llamado Plantilla usb, dando toda la ruta C:\Documents and Settings\Usuario\Mis documentos\Plantilla usb, una vez aquí seleccionamos el archivo comunicación.mcp y damos doble click, esto nos abrirá MPLAB-IDE y nos mostrará el programa main.c.
2. Damos doble click en main.c y el programa se abre, una vez abierto lo compilamos para evitar errores a la hora de la simulación figura 6.11.

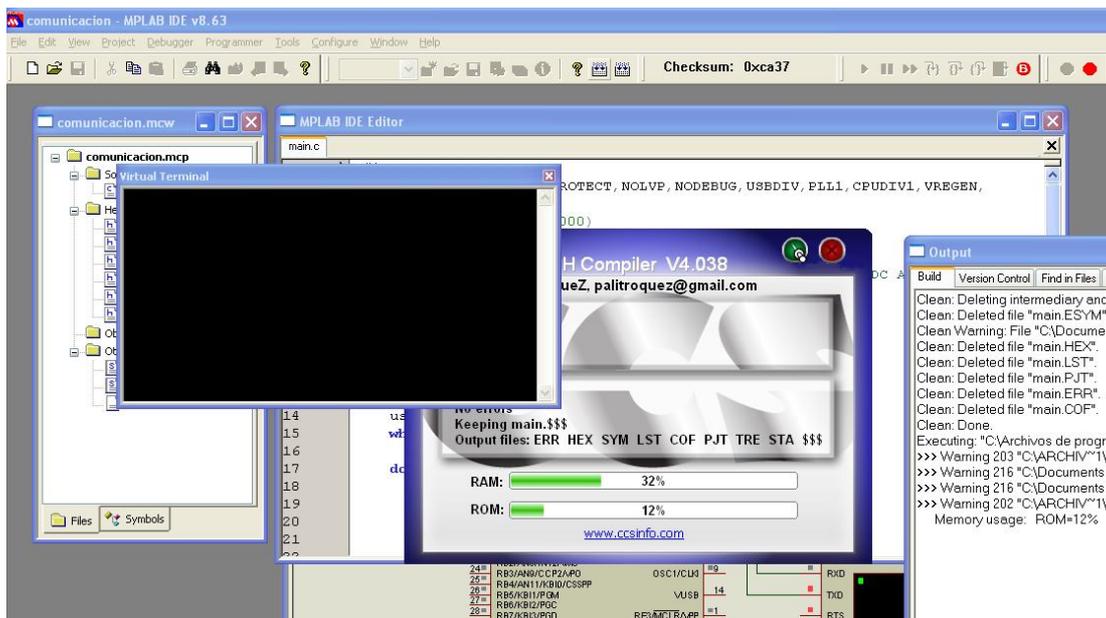


Figura 6.11 Pantalla de compilación

3. Ahora seleccionamos el icono *Start simulation* (comenzar simulación) y aparecerá el gráfico que diseñamos en Proteus, corremos la simulación dando click en el icono run (correr), podemos notar que la simulación comienza a

correr por la barra de progreso que se localiza en la parte inferior izquierda, al correr la simulación el programa levanta el puerto COM4 (puerto virtual). Y aparece una terminal virtual, que viene a ser el control de temperatura con el que nos conectamos por el puerto serie.

4. Para poder establecer la comunicación y ver si realmente está funcionando el circuito, abrimos la hiperterminal, nos aparece una pantalla figura 6.12 y aquí nos va a pedir un nombre, podemos darle el que sea, en este caso el nombre es *comunicación*, enseguida damos aceptar.



Figura 6.12 Pantalla de nueva conexión



Figura 6.13 Pantalla para seleccionar COM4

5. Aparece otra pantalla figura 6.13, aquí buscamos el COM4 si aparece quiere decir que el programa si levantó el COM4 y esto es indicio de que vamos bien, seleccionamos el COM4 y cambiamos los bits por segundo a 9600 baudios figura 6.14 y en control de flujo seleccionamos ninguno figura 6.15, damos aceptar y aparece la pantalla de la hiperterminal desde donde podemos escribir y lo que escribamos aquí debe aparecer en la terminal virtual

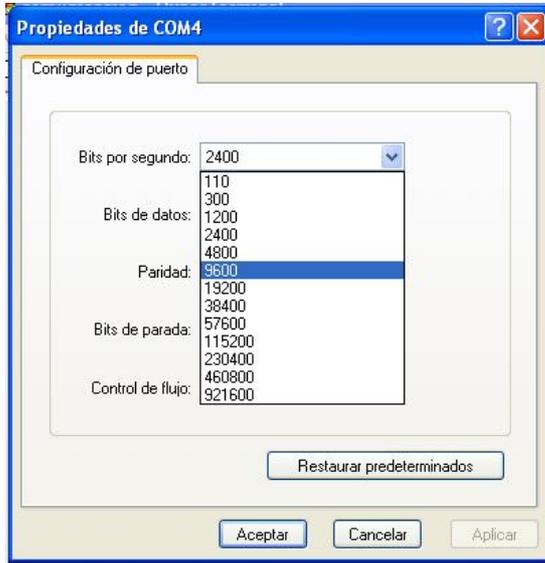


Figura 6.14 configuración de baudios



Figura 6.15 configuración del control de flujo

6. Escribimos en la pantalla de la hipertextual y aparece en la terminal virtual figura 6.16, con lo cual comprobamos que nuestro circuito y nuestro programa funcionan de manera correcta y han establecido comunicación, por lo cual podemos proceder con el alambrado con la seguridad de que tendrá éxito.

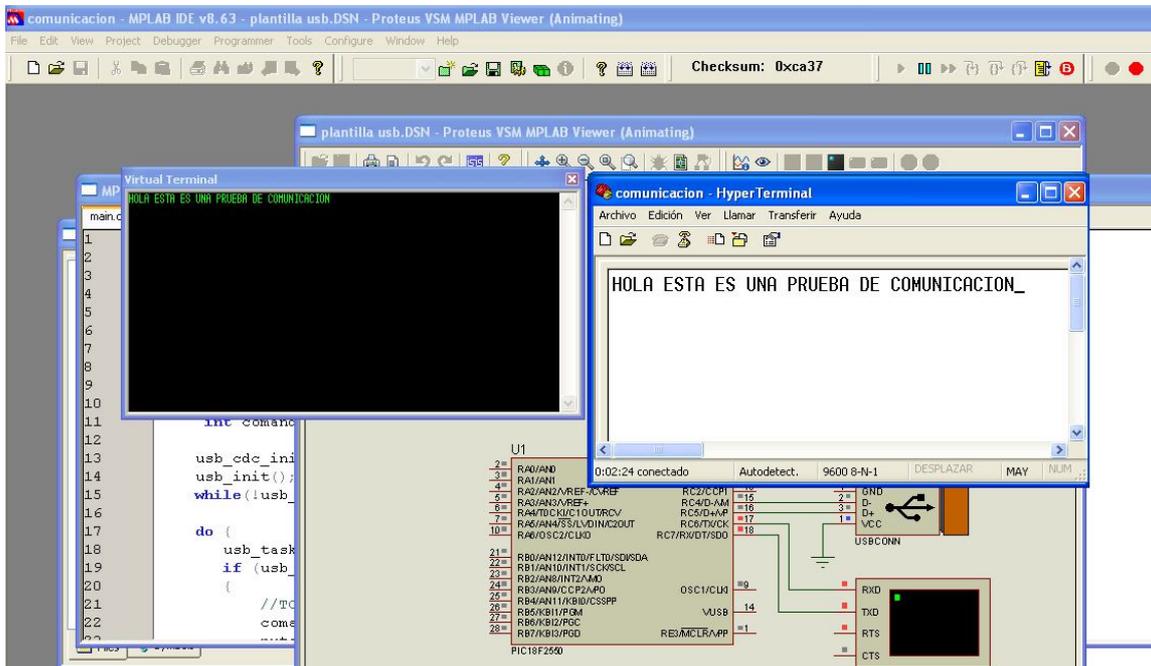


Figura 6.16 Comunicación hipertextual/terminal virtual

6.3.3 Grabando el PIC18F2550

Finalmente se debe programar el microcontrolador, es decir el programa que realizamos en C tenemos que introducirlo al PIC, para que éste, realice el trabajo que se especifica en el programa.

Para descargar las instrucciones a la memoria del programa del microcontrolador es necesario un dispositivo en hardware llamado **grabador** o **programador** que se conecta a una computadora personal a través del puerto serie, paralelo o USB. Para programar el PIC18F2550, utilice el **Pickit 2 v2.61** figura 6.17, que es un programador de fácil manejo, con una interfaz gráfica muy amigable proporcionada por Microchip.



Figura 6.17 Programador de PIC's **PICKit 2 v2.61**



Figura 6.18 Grabador conectado a la Lap-Top

Como primer paso colocamos el PIC18F2550 en la base del grabador, adecuada para este PIC, cuidando que todas los pines o patillas del PIC se inserten correctamente en la base, ya que de lo contrario si un pin se dobla o no se inserta de forma correcta, ocasionará que no se grabe el PIC o que el programa del Pickit 2 no lo detecte.

Una vez insertado el PIC se conecta la tarjeta a la Lap-Top, mediante su cable USB y en el programador debe iluminarse el led correspondiente a power (poder), figura 6.18 lo que nos indica que está encendido y listo para grabar.

A continuación abrimos el programa de Pickit 2 y aparece la pantalla mostrada en la figura 6.19, desde aquí no tenemos más que seguir algunos sencillos pasos dando click en algunos iconos para grabar nuestro PIC

Como medida de precaución damos click en el botón Erase (borra), esto con el fin

de eliminar la probable basura que pudiera traer el PIC en su memoria, una vez que haya ejecutado este paso aparecerá la leyenda Erasing device complete (borrado del dispositivo completo), ahora ya es seguro programar el PIC.

El siguiente paso es buscar la familia del dispositivo que vamos a grabar figura 6.20, que en este caso es la familia PIC18, inmediatamente el programa reconocerá el dispositivo y desplegará el nombre completo del PIC en la parte correspondiente a *Device*

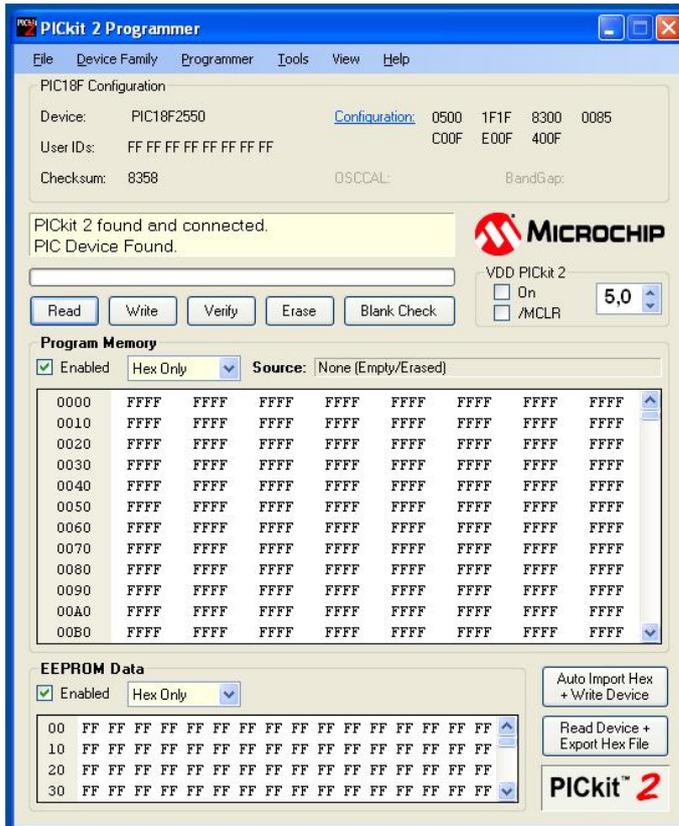


Figura 6.19 Interfaz gráfica

Ahora lo que tenemos que hacer es importar el programa con extensión .HEX que se generó cuando compilamos nuestro programa desde MPLAB-IDE y que por lo tanto se creó en la misma dirección donde está alojado el programa fuente main.c, damos click en *File* y enseguida en Import Hex, Figura 6.21.

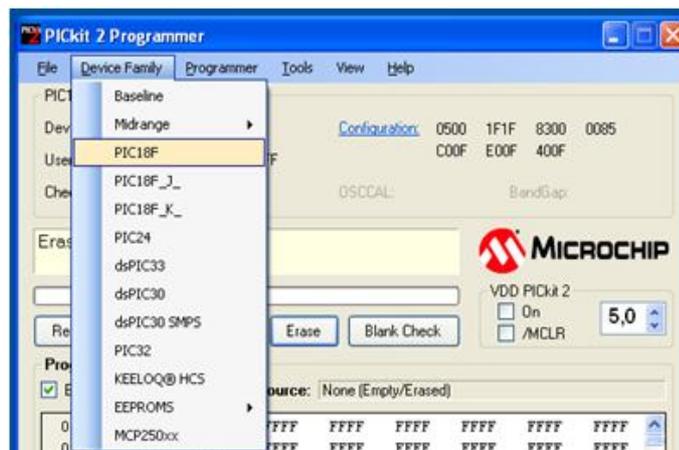


Figura 6.20 Familia PIC18

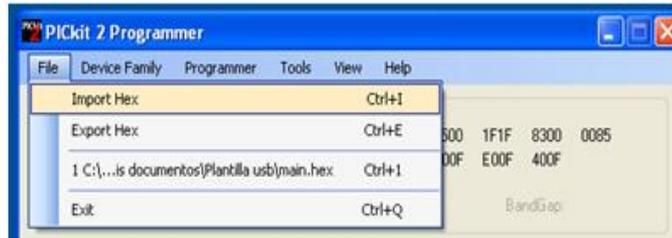


Figura 6.21 Importar Hex

Buscamos el archivo.Hex en la carpeta Plantilla usb figura 6.22, seleccionamos el archivo, abrimos y el archivo será transferido al programa Pickit 2 y nos informará si el archivo con extensión .Hex fue importado con éxito

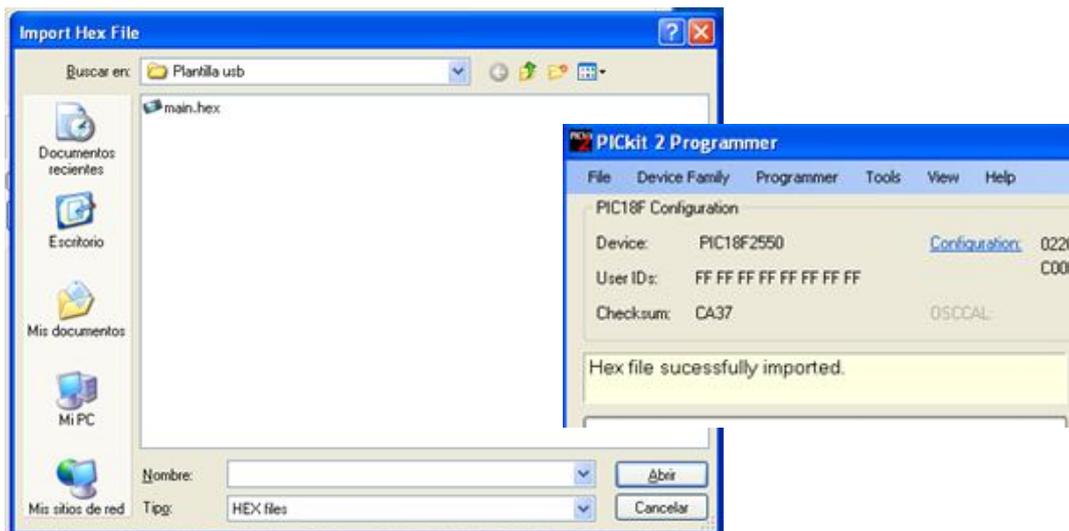


Figura 6.22 Archivo main.hex

Ahora lo que tenemos que hacer es dar click en el ícono write (escribir), para que el programa Pickit 2 comience a grabar el archivo .Hex dentro del PIC.

Al momento de estar grabando el archivo en el PIC, la barra de progreso comenzará a desplazarse una vez que se haya completado, si todo salió bien la pantalla del programa desplegará la leyenda **Programming Successful**, figura 6.23, lo cual quiere decir que se programó con éxito nuestro PIC

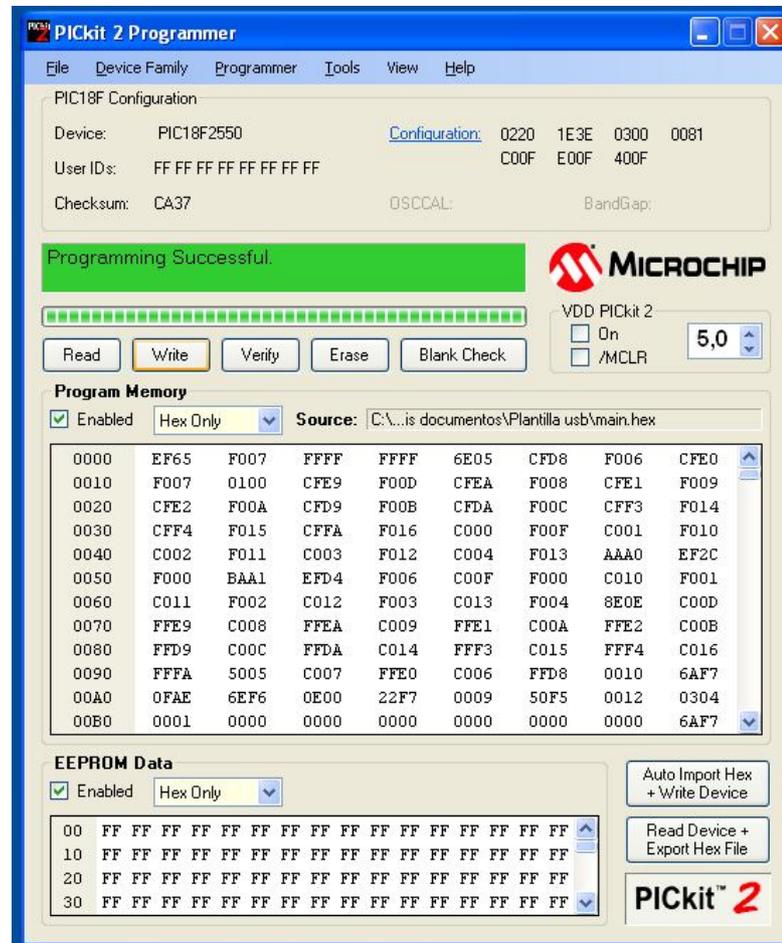


Figura 6.23 Pantalla de Programación exitosa

Por último sólo nos resta desconectar el PIC del programador para ensamblarlo en el circuito que previamente diseñamos y echarlo a funcionar, esperando los mejores resultados.

6.4 Prueba de la interfaz en hardware (Circuito alambrado)

En el capítulo No 2 se mencionaron los componentes eléctricos y electrónicos con los cuales se lleva a cabo el calentamiento de los tanques de maduración de gelatina, así como la forma en que estos están conectados, esto se hizo por medio de dos diagramas, figuras 2.10 y 2.11 del capítulo 2, a partir de estos diagramas, construimos un modelo usando los mismos instrumentos que componen el control de temperatura; resistencia, sensor, control de temperatura FUJI PXG-4 y relevador de estado sólido, además agregamos un cable serial RS232 con

conector DB-9 para el control FUJI, que como ya se explicó al principio de este capítulo, es el único tipo de cable con que cuentan este tipo de instrumentos, véase la figura 6.24

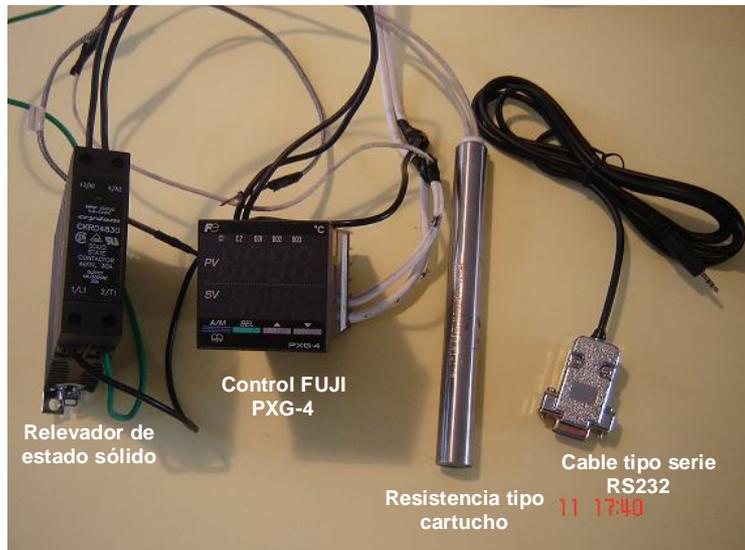


Figura 6.24 componentes del control de temperatura

6.4.1 Conexión de componentes

Usando el cable USB conectamos la lap-top al circuito o interfaz hardware y a su vez utilizando el cable serieRS232 conectamos el circuito al control de temperatura como se muestra en la figura 6.25

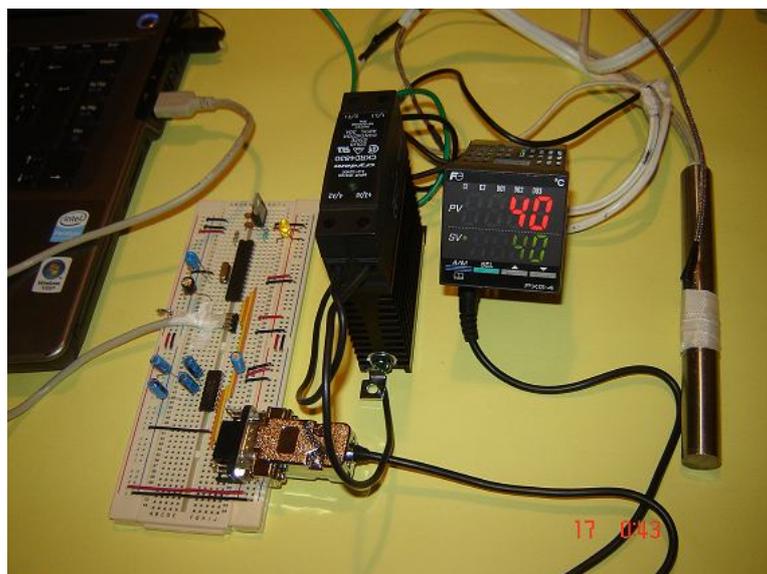


Figura 6.25 Interfaz hardware conectada a Lap-Top y al control de temperatura

Ahora podemos comunicar nuestra lap-top con el control de temperatura y enviar los comandos desde la interfaz de software, dando los parámetros de temperatura y tiempo con los que va a trabajar el control de temperatura.

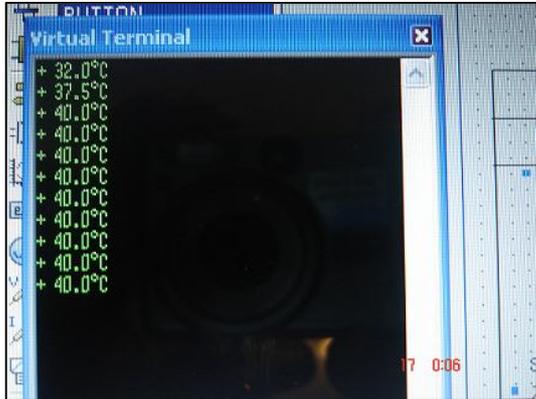


Figura 6.26 Terminal virtual

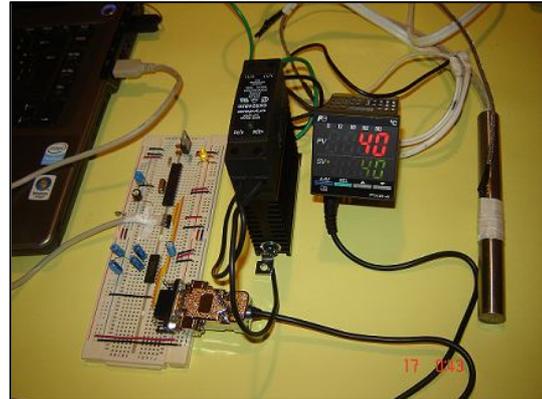


Figura 6.27 Control de temperatura

Al mandar el comando de temperatura desde la hiperterminal, es recibido tanto por la terminal virtual figura 6.26 como por el control de temperatura figura 6.27, con lo cual tenemos la lectura que se genera en el control en nuestro monitor, de esta forma podremos saber en cualquier momento del proceso la temperatura que está marcando el control de temperatura.

6.5 Conclusión capítulo 6

De esta forma vemos concluido el armado de nuestro circuito o interfaz hardware, habiendo obtenido los mejores resultados, una vez que se hicieron todas las pruebas, tanto físicas, como simuladas en la computadora, con lo cual pudimos comprobar que el circuito trabajaba de forma óptima, de acuerdo al fin para el cual fue creado.

Como se pudo ver a lo largo de este capítulo, para llevar a cabo esta parte del proceso, nos auxiliamos de varias herramientas, que fueron de gran utilidad y que nos ayudaron a garantizar el éxito del alambrado.

Haciendo un recuento, para poder programar el PIC18F2550 y hacer la simulación, contamos con los siguientes programas:

MPLAB-IDE Ambiente de Desarrollo Integrado donde se elaboró el proyecto

Proteus (ISIS) Para el diseño gráfico del circuito

Proteus (VSM) Para simular el funcionamiento del circuito

Pickit 2 (v2.61) Para Grabar el PIC18F2550

El uso de estas herramientas no sólo garantizó un buen alambrado, sino que nos ahorraron mucho tiempo, pues evitamos una serie de problemas que se pueden presentar a la hora de alambrear un circuito.

CONCLUSIONES.

Al llegar al término de este proyecto, pude apreciar en su totalidad la gama de herramientas que tuve que utilizar, para poder llevarlo a cabo. Las herramientas que usé fueron enfocadas tanto en el área del software como del hardware, lo que me hizo sentir que apliqué lo aprendido a lo largo de la carrera, lo cual me llenó de satisfacción, sin embargo también caí en cuenta, que se tiene que investigar más allá de lo aprendido en las aulas de clase, para encontrar las soluciones a los problemas que se nos plantean durante el desarrollo de un proyecto y poder ir encontrando las soluciones que nos harán llegar hasta el final con éxito.

Desde que comencé a plantear las posibles soluciones, al problema del control adecuado de las temperaturas de los tanques de maduración de gelatinas, tuve claro que la aplicación ideal era utilizar los microcontroladores PIC de microchip, por su facilidad para programar, ya que lo podemos programar en lenguaje ensamblador o en lenguaje C, su costo accesible y que son de uso específico, le dan una gran potencialidad y una ventaja sobre otros circuitos integrados. En un principio pensé en utilizar el PIC16F877A ya que estaba muy familiarizado con él, pero al llegar a la parte del desarrollo hardware y hacer la simulación éste PIC no era el ideal para dar solución al circuito que habíamos planteado, ya que no cuenta con terminales para el manejo de interfaz USB, por lo cual me di a la tarea de buscar un PIC para simplificar el circuito y que además se adaptara tanto al uso de interfaz USB como de interfaz RS232, para esto sólo tuve que investigar en la página web de microchip (<http://www.microchip.com/>), buscando un PIC con estas características y encontré el PIC18F2550, que como vimos en el capítulo 6 presentaba terminales para el manejo de interfaz USB y RS232, lo que nos facilitó el alambrado y el diseño de la interfaz que comunica a la computadora encargada de los comandos a través de la interfaz gráfica de usuario y al control de temperatura encargado del calentamiento del tanque de maduración.

Para diseñar la interfaz o circuito se requieren los conocimientos básicos de electrónica aprendidos en materias como: diseño lógico, diseño de sistemas digitales, microcomputadoras y control digital; pues se requiere conocer y manejar componentes como: resistencias, capacitores, reguladores de voltaje, leds, protoboards, circuitos integrados como el MAX232, y por supuesto los microcontroladores PIC de microchip.

También es necesario tener las nociones básicas de eléctrica, para conocer y saber cómo funcionan y están conectados los componentes que forman parte del control de temperatura de los tanques de maduración y que son: relevador de estado sólido, resistencia de inmersión, sensor de temperatura y control de temperatura, esto fue con la finalidad de poder armar un modelo del mismo, el cual nos permitió trabajar sin la necesidad de tener un tanque de maduración a la mano.

Como ya se mencionó los PIC's pueden ser programados en C o en ensamblador, en cuanto al programa elegido para la programación del PIC18F2550, decidí usar lenguaje C, porque es el más adecuado para la solución del problema, además de que una vez escrito el programa es más fácil de analizar y entender, sobre todo si ya se está familiarizado con este lenguaje.

Para la interfaz de usuario utilicé Visual C# por presentar características muy parecidas a Visual Basic, pero con una mayor estabilidad, así como mejores herramientas.

C# elimina muchos elementos añadidos por otros lenguajes y que facilitan su uso y comprensión, como por ejemplo ficheros de cabecera, o ficheros fuentes. Es por ello que se dice que C# es autocontenido, además, no se incorporan al lenguaje elementos poco útiles, como por ejemplo macros, herencia múltiple u operadores diferentes al operador de acceso a métodos (operador punto) para acceder a miembros de espacios de nombres.

Al ser C# un lenguaje de última generación, incorpora elementos que se ha demostrado a lo largo del tiempo que son muy útiles para el programador, como

tipos decimales o booleanos, un tipo básico string, así como una instrucción que permita recorrer colecciones con facilidad (instrucción foreach).

C# como lenguaje de última generación, y de propósito general, es orientado a objetos. C# no permite la inclusión de funciones ni variables globales que no estén incluidos en una definición de tipos, por lo que la orientación a objetos es más pura y clara que en otros lenguajes como C++. Además, C# soporta todas las características del paradigma de la programación orientada a objetos, como son la encapsulación, la herencia y el polimorfismo. La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular. La sintaxis de C# incluye por ejemplo formas de definir propiedades, eventos o atributos.

Una vez realizada la interfaz gráfica de usuario y probada, se hicieron las pruebas de la interfaz de hardware, cuando se comprobó que las dos interfaces trabajaban de forma correcta, se procedió a mandar los primeros parámetros al control de temperatura, sin que se presentará ningún inconveniente, pudiendo controlar desde la computadora la temperatura y el tiempo con el que trabajaría el control de temperatura, de esta forma llegamos al fin del proyecto, obteniendo los mejores resultados, gracias a la investigación realizada, pero sobre todo y como en todo proyecto, al esfuerzo, el trabajo y la dedicación.

BIBLIOGRAFÍA.

- [1] ANGULO, José Ma. *Microcontroladores PIC. La Clave del Diseño*. Thomson. España. 2003.
- [2] ANGULO, José Ma. *Microcontroladores PIC. Diseño Práctico de aplicaciones* Mc Graw-Hill. España. Tercera edición. 2006
- [3] CEBALLOS, Francisco J. *Visual C++ Programación Avanzada en Win32*. Alfaomega. España 1999.
- [4] GARCÍA, Eduardo. *Compilador C CCS y simulador PROTEUS para Microcontroladores PIC* Alfaomega Primera Edición 2008
- [5] PALACIOS, Enrique. *Microcontrolador PIC16F84 Desarrollo de Proyectos*. Alfaomega Ra-Ma. Primera Edición, México. 2004.
- [6] PEÑALOZA, Ernesto, *Fundamentos de Programación C/C++*, Alfaomega, Cuarta Edición, México, 2004.
- [7] TANENBAUM, Andrew S. *Organización de Computadoras. Un Enfoque Estructurado*. Pearson Educación. Cuarta Edición. 2000.
- [8] VALDÉS, Fernando y PALLAS, Ramón. *Microcontroladores Fundamentos y Aplicaciones con PIC*. Alfaomega Primera Edición 2007

MESOGRAFÍA

- [1] Atmel Corporation (2011) Microcontroladores
<http://www.atmel.com>

- [2] Estándar RS232 (2010) Estándar de comunicación RS-232C
<http://www.euskalnet.net>

- [3] Freescale Semiconductor (2011) Información Microcontroladores 8 y 16 bits
<http://www.freescale.com>

- [4] MAX232 (2010) Hoja de datos del circuito MAX232
<http://www.datasheetcatalog.com>

- [5] Microchip Inc (2011) MPLAB-IDE (ambiente de desarrollo integrado)
<http://microchip.com>

- [6] Microchip Inc (2011) Hoja de datos PIC18F2550
<http://microchip.com>

- [7] PICKit 2 (2011) Programador PICKit 2 v2.61
<http://microchip.com>

- [8] Visual C# Express Edition (2011) Compilador C#
<http://www.microsoft.com/express/Windows>