



**UNIVERSIDAD AUTONOMA DE SAN LUIS POTOSÍ**

**FACULTAD DE CIENCIAS**

**INSTITUTO DE INVESTIGACIÓN EN COMUNICACIÓN ÓPTICA**

**AUTOMATIZACIÓN DEL SISTEMA DE CRECIMIENTO DE CRISTALES  
POR EPITAXIA EN FASE LÍQUIDA**

**TESIS**

**QUE PARA OBTENER EL GRADO DE**

**MAESTRO EN**

**CIENCIAS APLICADAS**

**PRESENTA:**

**ING. FRANCISCO SÁNCHEZ NIÑO**

**ASESOR:**

**DR. FRANCISCO J. DE ANDA SALAZAR**

**SAN LUIS POTOSÍ, S. L. P., AGOSTO DEL 2009**

## **Dedicatoria**

Gracias Dios mío por iluminar mi camino.

A mis padres que sin ellos no hubiera sido posible.

Gracias por ser la fuente de motivación de salir adelante.

Gracias por su apoyo siempre incondicional, su comprensión y cariño a lo largo de mi formación académica.

## **Agradecimientos**

Al Dr. Francisco J. de Anda Salazar por darme la oportunidad y el apoyo para realizar el presente proyecto de tesis.

A mis hermanos Graciela, Javier, Roció, Alejandro, Miguel y Oscar por su comprensión y apoyo durante este tiempo.

A todos mis profesores, compañeros y amigos que estuvieron conmigo en el trayecto de mi formación académica y personal.

Gracias por tu respaldo, comprensión y cariño durante este tiempo Eve.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por su apoyo al otorgar una beca para estudios de posgrado durante el período de septiembre de 2006 a agosto de 2008.

## **Resumen**

En la actualidad el desarrollo tecnológico brinda la oportunidad de encontrar diferentes soluciones a diversos problemas en el ámbito industrial, comercial, científico, escolar, etcétera. Una de las soluciones que ofrece la tecnología es optimizar los procesos involucrados en las actividades propias de cada una de las disciplinas mencionadas con la finalidad de lograr mejores resultados en cada uno de sus proyectos o trabajos, logrando con esto una mayor rentabilidad en los sistemas utilizados.

Este trabajo de tesis consiste en el diseño e implementación del software y hardware de control de los sistemas de crecimiento por epitaxia en fase líquida del laboratorio de crecimiento de cristales. Se utilizó un microcontrolador PIC18F4550 como cerebro principal de las tarjetas de control realizando la transferencia de datos entre el PIC y la PC por medio del puerto USB que es una de las novedades presentes en el trabajo de tesis. El software de control se realizó en el ambiente de programación gráfica "LabVIEW" ya que ofrece un ambiente de interacción con el usuario muy agradable y versátil.

## INDICE GENERAL

### CAPÍTULO I

#### INTRODUCCIÓN

1.1	Epitaxia en fase líquida.....	7
1.2	Aplicaciones de la Epitaxia en fase líquida.....	8
1.3	Descripción de los objetivos del trabajo de tesis.....	8
1.4	Ventajas de automatizar los sistemas de crecimiento.....	9

### CAPÍTULO II

#### SISTEMA EXPERIMENTAL DEL LABORATORIO

2.1	Descripción de los sistemas de crecimiento de cristales.....	10
2.2	Proceso operativo.....	13
2.3	Parámetros físicos involucrados.....	17

### CAPÍTULO III

#### HARDWARE DE CONTROL

3.1	Circuito de potencia.....	19
3.2	Circuito de control electrónico.....	21
3.3	Circuito de adquisición de señales.....	24
3.4	Fuentes de alimentación.....	30
3.5	USB.....	31
	3.5.1 Características de transmisión.....	32
	3.5.2 Compatibilidad y conectores.....	34
	3.5.3 PIC18F4550 y USB.....	35
3.6	Circuito impreso.....	36

## **CAPÍTULO IV**

### **SOFTWARE DE CONTROL**

4.1	Programa desarrollado en el microcontrolador.....	39
4.2	LabVIEW.....	42
4.3	Programa principal de LabVIEW.....	45
4.4	Modos de operación del software de la PC.....	47
4.4.1	Modo manual.....	47
4.4.2	Modo válvulas.....	49
4.4.3	Modo bloques.....	51
4.4.4	Modo automático.....	53

## **CAPÍTULO V**

### **RESULTADOS**

5.1	Pruebas realizadas.....	55
-----	-------------------------	----

## **CAPÍTULO VI**

### **CONCLUSIONES**

6.1	Conclusiones.....	65
6.2	Trabajo futuro.....	66

### **APÉNDICE**

A.	Programa de Control del Microcontrolador PIC18F4550.....	67
B.	Programa principal de control de la PC implementado en LabVIEW.....	82
B.1.	Subprograma modo Automático.....	84
B.2.	Subprograma modo Válvulas.....	84
B.3.	Subprograma modo Manual.....	89
B.4.	Subprograma modo Bloques.....	90

<b>BIBLIOGRAFÍA.....</b>	<b>93</b>
--------------------------	-----------

# CAPÍTULO I

## INTRODUCCIÓN

### 1.1 Epitaxia en fase líquida.

El término epitaxia (del griego *epi*, sobre, *taxis*, orden) apareció por primera vez hace aproximadamente 50 años. Este término se refiere a un proceso de crecimiento orientado de una película sobre un sustrato, que puede ser del mismo material que la película (homoepitaxia) o bien de un material diferente (heteroepitaxia). A los procesos de cristalización de películas sobre la superficie de un sustrato se les llama, en forma genérica, métodos de crecimiento epitaxial de cristales.

En el método epitaxial por fase líquida la cristalización de las capas crecidas tiene lugar a partir de una solución saturada con un material semiconductor, en la que se utiliza un solvente previamente escogido. Para el crecimiento de capas epitaxiales semiconductoras de los grupos III-V y las soluciones sólidas que se obtienen es muy cómodo utilizar en calidad de solvente el Ga y el In.

Estos elementos tienen temperaturas de fusión bajas, y además entran en la composición de las películas crecidas. Tanto el Ga como el In tienen presión de vapor muy baja en el intervalo de temperatura que generalmente es utilizado en el proceso de crecimiento por el método de fase líquida. Además, estos dos elementos se pueden obtener muy puros y a las temperaturas usuales de crecimiento ni el Ga ni el In interactúan con el grafito, material del cual están hechos generalmente los contenedores de las fases líquidas.

Para llevar a cabo el proceso de epitaxia es necesario conocer el correspondiente diagrama de fase que determina la relación entre las composiciones de las fases Líquida y sólida a diferentes temperaturas.

El método epitaxial por fase líquida fue utilizado por primera vez por H. Nelson para el crecimiento de uniones p-n en base al GaAs (1963).

Cabe señalar que la epitaxia por fase líquida, desde el punto de vista técnico, es bastante sencillo y por esto es posible realizar el montaje experimental en condiciones normales de laboratorio. Precisamente, la técnica de crecimiento por fase líquida revolucionó las posibilidades de la época para mejorar los parámetros de dispositivos ya existentes y crear otros nuevos en base a heteroestructuras de capas múltiples, como es el caso de los heteroláseres.

## **1.2 Aplicaciones de la Epitaxia en fase líquida.**

En la tecnología actual de crecimiento de heteroestructuras láser las técnicas de crecimiento epitaxial son utilizadas de manera extensiva. Estas técnicas ocupan un lugar muy importante en la tecnología de semiconductores relacionada con el diseño de circuitos integrados, así como de dispositivos semiconductores entre los que se cuentan fotodetectores, fotodiodos y transistores de alta frecuencia.

Dependiendo de la forma de transportar el material a crecer desde la fuente hasta el sustrato, todos los procesos epitaxiales se dividen en tres tipos: (a) epitaxia por fase líquida, (b) epitaxia por fase gaseosa, (c) epitaxia por haces moleculares.

Hoy en día todos estos procesos son utilizados en el crecimiento de los más diversos dispositivos y estructuras semiconductoras. Por otro lado, también son útiles en la búsqueda de nuevos materiales semiconductores. Debido a que cada uno de estos procesos tiene una serie de particularidades muy específicas.

## **1.3 Descripción de los objetivos del trabajo de tesis.**

El objetivo principal de este trabajo de tesis es el de automatizar los procesos involucrados en el sistema de crecimiento de cristales por epitaxia en fase líquida para la obtención de mejores resultados y mejorar la reproducibilidad.

Esto se pretende realizar a través del monitoreo de los parámetros físicos involucrados en el proceso de crecimiento, tales como: humedad relativa dentro del sistema y la humedad relativa en el laboratorio, temperaturas de crecimiento y temperatura ambiente así como las presiones dentro del sistema que se generan al hacer fluir hidrógeno y cuando se le hace vacío a este.

Este monitoreo de variables físicas se realizará a lo largo de todo el proceso. En función de las lecturas obtenidas al leer el estado de las variables mencionadas anteriormente, el programa de control, tomará una serie de decisiones para llevar a cabo cada uno de los procesos involucrados en el sistema de crecimiento de cristales por epitaxia en fase líquida, tales como vacío al sistema, flujo de hidrógeno, crecimiento y apagado del sistema.



#### 1.4 Ventajas de automatizar los sistemas de crecimiento.

Algunas de las ventajas de automatizar los sistemas de crecimiento de cristales por epitaxia en fase líquida son las siguientes:

- Permite un registro de datos a lo largo de todo el experimento.
- Un mejor control de los procesos involucrados en el sistema.
- Monitoreo constante del crecimiento.
- En función de los datos registrados se podrán evaluar las condiciones óptimas para realizar un crecimiento.
- Permitirá reducir errores en el accionamiento de las Electroválvulas ya que estas se accionarán electrónicamente y no en forma manual.
- Obtener mejores resultados en los crecimientos de cristales.
- Y en función de todo esto se espera obtener una mayor reproducibilidad de los experimentos.
- Mayor seguridad.

## CAPÍTULO II

### SISTEMA EXPERIMENTAL DEL LABORATORIO

#### 2.1 Descripción de los sistemas de crecimiento de cristales.

Epitaxia en fase líquida es una técnica que ha sido ampliamente estudiada para la obtención de capas epitaxiales de materiales semiconductores y de materiales con dominios magnéticos y ferroeléctricos. Una de las principales ventajas que ofrece esta técnica es que permite hacer el crecimiento a temperaturas por debajo de la temperatura de fusión de los cristales.

Cabe mencionar que el sistema experimental está constituido por dos sistemas de crecimiento los cuales son idénticos, por lo que a lo largo de la descripción del trabajo de tesis solo se hará mención a uno de los sistemas.

El sistema experimental consta de un horno cilíndrico con resistencia de Kanthal para elevar la temperatura, un termopar tipo K para medir la temperatura al lado de la resistencia, un controlador – programador conectado a un SSR para regular dicha temperatura, además contiene un bote de grafito tipo panel el diseño más sencillo de este se muestra en la figura 2.1, el cual se compone de dos partes: el tronco y el deslizador; el tronco generalmente no se mueve y en sus cavidades se alojan las fases líquidas de diferentes composiciones

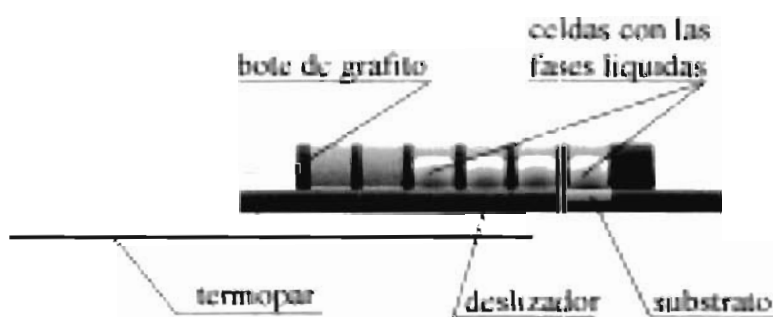


Figura 2.1. Con ayuda de este bote de grafito es posible crecer estructuras de capas múltiples. Los sistemas modernos de fase líquida utilizan este tipo de botes; su presentación puede variar según lo que se desee crecer

El deslizador tiene una cavidad en la cual se deposita el sustrato, éste se desliza en relación con el tronco y de esta manera el sustrato se pone en contacto con cada una de las fases líquidas contenidas en las cavidades del tronco.

Este tipo de bote de grafito se utiliza con el objetivo de obtener estructuras más complicadas que contengan varias capas con diferentes composiciones.

Además consta de un reactor de cuarzo dentro del cual está el bote de grafito donde fluye hidrógeno a través de él durante el crecimiento. Para tener un flujo de Hidrógeno constante y controlado, se requiere un controlador de flujo de gases.

Para garantizar un alto grado de pureza en el hidrógeno, se necesita un purificador con celda difusora hecha de paladio y plata.

Es necesario monitorear las concentraciones de vapor de agua presentes en el sistema ya que concentraciones altas pueden alterar el experimento, por lo que se requiere un analizador de partes por millón de vapor de agua. Se hace necesario también un sistema de vacío para limpiar el sistema cuando se requiera, como en el caso en el cual se abre el reactor al medio ambiente, para este caso se utilizan dos tipos de bomba con las cuales se hace vacío al sistema, la primera de ellas es una bomba tipo Jet la cual genera un vacío burdo y para realizar un vacío más fino se utilizan unas bombas llamadas de adsorción.

En la figura 2.2 se muestra el esquema general de los sistemas de crecimiento de cristales por epitaxia en fase líquida.

## ESQUEMA GENERAL DEL SISTEMA

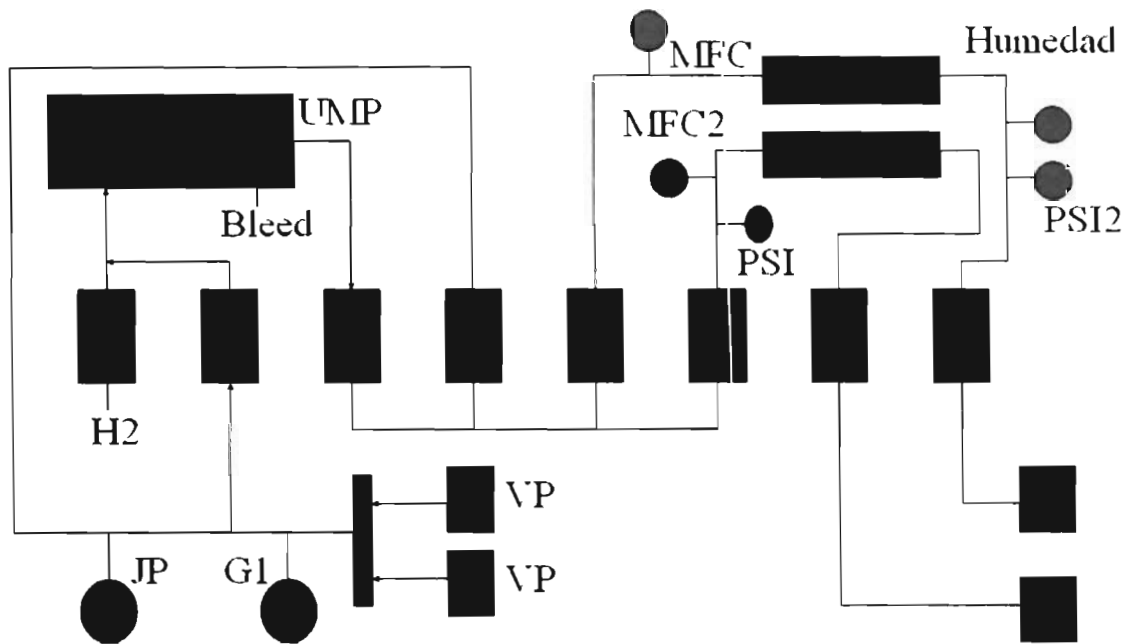


Figura 2.2. En esta figura se muestra el esquema general del sistema de crecimiento.

Este sistema consta de una serie de tuberías interconectadas entre sí por medio de válvulas neumáticas las cuales son accionadas por electroválvulas esto con el objetivo de realizar los diferentes procesos involucrados en el sistema.

Cabe mencionar que se hace necesaria la utilización de válvulas neumáticas debido a que a través del sistema fluye hidrógeno el cual es altamente inflamable y no debe tener una interacción con O<sub>2</sub>.

## 2.2 Proceso operativo.

En esta sección se describirá de una forma breve la manera en la cual se realiza un crecimiento de cristales esto con el fin de dar una perspectiva muy generalizada de los pasos involucrados en el proceso.

Esta técnica se basa en que la solubilidad de un constituyente diluido en un solvente líquido, decrece cuando disminuye la temperatura; es por esto que el enfriamiento de una solución saturada conlleva a la aparición de la fase sólida en ésta. La técnica de EFL aprovecha ese fenómeno para colocar en contacto en ese momento la solución con el sustrato, y generar así una deposición del material en fase sólida sobre él. La fase sólida se adhiere al sustrato de manera ordenada, conservando las características cristalinas del mismo.

En primer lugar es necesario preparar una solución saturada la cual se colocará en el tronco del bote de grafito y el sustrato, donde se depositarán los cristales, se coloca en el deslizador del bote de grafito, una vez realizado esto se introduce el bote de grafito en el tubo de cuarzo y se cierra el reactor.

Como el reactor estuvo expuesto al medio ambiente a la hora que se colocó el material, es necesario, realizar un vacío al sistema, esto con el fin de extraer todo el aire que se introdujo dentro del reactor, para esto se realiza primero un vacío no muy fino con la bomba Jet, con la cual se obtienen vacíos en el orden de los dos PSI o menos, en la figura 2.3 se muestra la forma de como se realiza este vacío.

### Vacío Bomba Jet

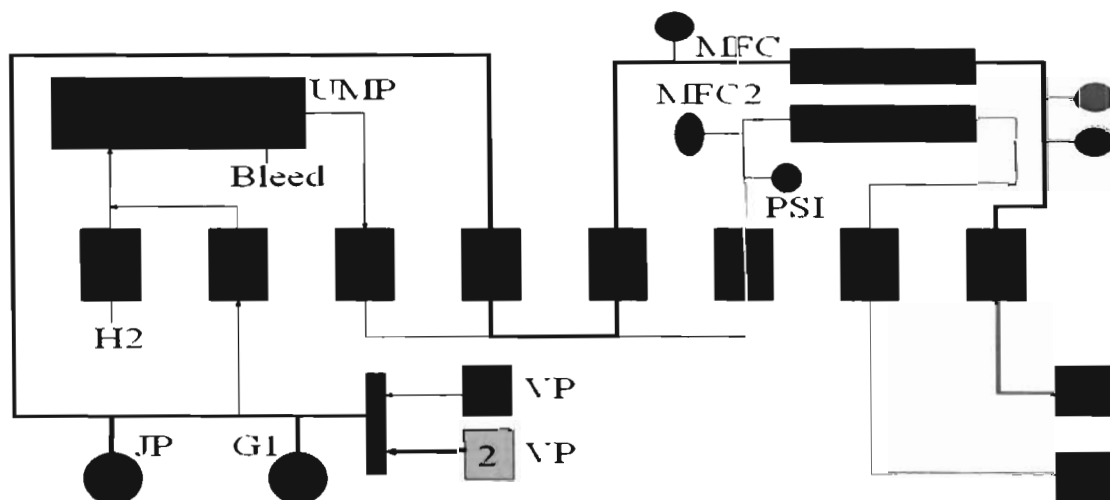


Figura 2.3. Accionamiento de válvulas para la realización de vacío con la Bomba Jet.

Se abre la válvula de paso de la bomba Jet (JP) y se accionan las válvulas cuatro y cinco para realizar el vacío al sistema tal como se muestra en la figura 2.3, la tubería y válvulas en color oscuro muestran el sentido del proceso.

Una vez realizado este paso se procede a realizar el vacío fino por medio de las bombas de adsorción (VP 1 y 2) una vez más se accionan las válvulas cuatro y cinco con este vacío se obtienen lecturas en el orden de los 100 mili torr, esto se observa en la figura 2.4 aquí termina el proceso de aplicar vacío al sistema.

## Vacío Bomba De Adsorción

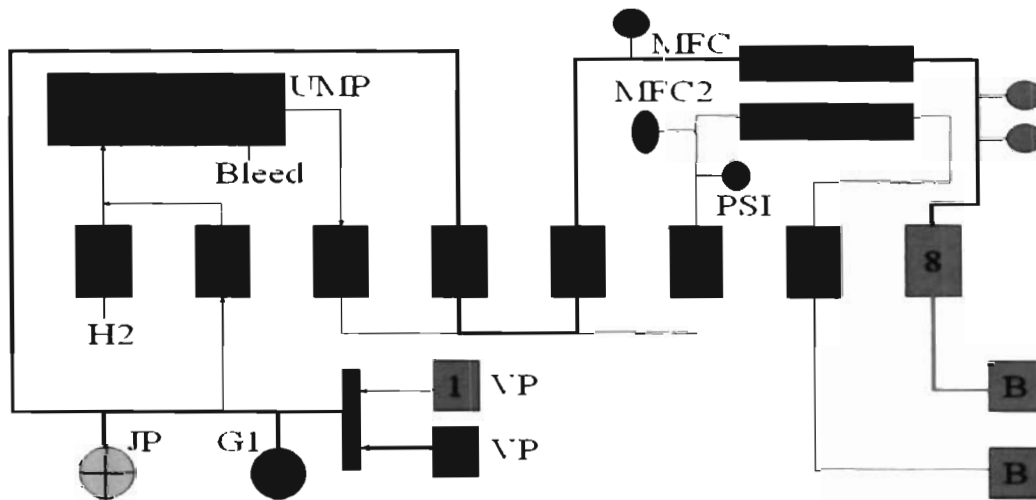


Figura 2.4. Accionamiento de válvulas para la realización de vacío con las Bombas de adsorción.

Las bombas de adsorción cumplen con la función de absorber la mayor cantidad de partículas que se introdujeron a la hora de introducir el material en el reactor. Una de las características de este tipo de bomba es que entre más fría se encuentre mayor será la capacidad de adherencia que tendrán, es por eso que se requiere del uso de nitrógeno líquido para enfriarlas antes de ser utilizadas, una vez que las bombas se saturan, en ese momento termina la generación de vacío con estas.

Una vez realizado el vacío al sistema se procede a la purificación del sistema el cual consiste en hacer circular hidrógeno purificado a través del reactor de cuarzo. La utilización del hidrógeno permite eliminar las películas de óxido que casi siempre están presentes en la superficie del material que se utiliza como solvente así como sobre la superficie del sustrato y del material utilizado para saturar la fase líquida. La existencia de estas películas de óxido no permite que haya un buen contacto entre el sustrato y la solución. En la figura 2.5 se muestra el esquema de circulación de hidrógeno en el sistema.

## Flujo de Hidrogeno

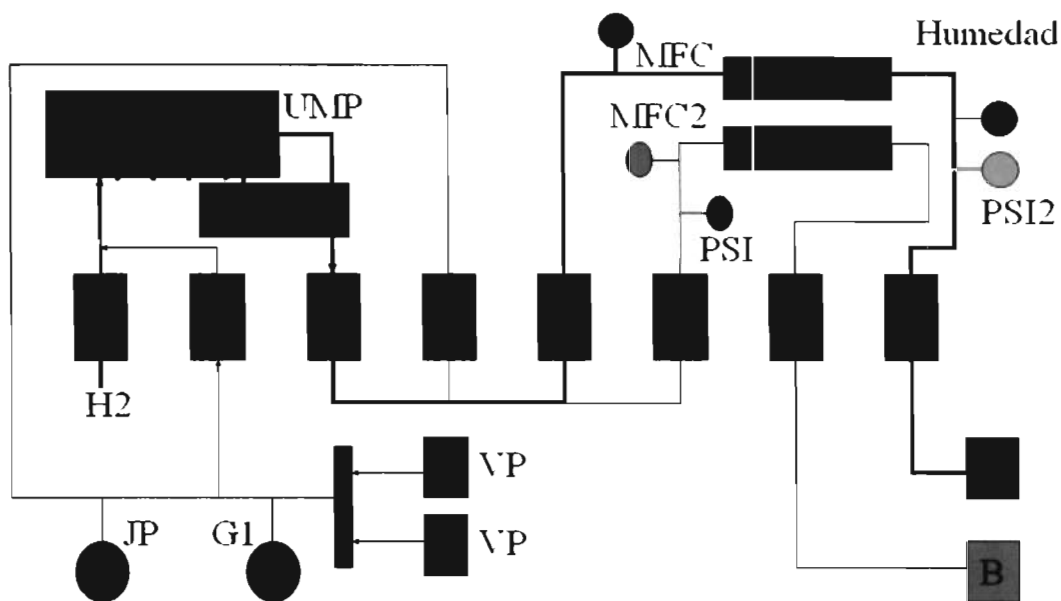


Figura 2.5. Accionamiento de válvulas para la purificación del reactor.

Hay que señalar que la purificación del sistema se realiza con la apertura de la válvula uno la cual permite la entrada del hidrógeno proveniente del tanque, este pasa a través del purificador obteniendo a la salida de éste hidrógeno ultra purificado. Posteriormente se accionan las válvulas tres y cinco con lo cual se permite el paso de hidrógeno hacia el reactor del tubo de cuarzo donde se contiene el material y una vez que se alcanza una presión interna de 15 PSI se abre la válvula ocho y se deja fluir el hidrógeno hasta el final del experimento.

Finalmente una vez que se han realizado estos pasos se comienza con la etapa de crecimiento la cual consiste en implementar un diagrama de temperatura, tal como se muestra en la figura 2.6.

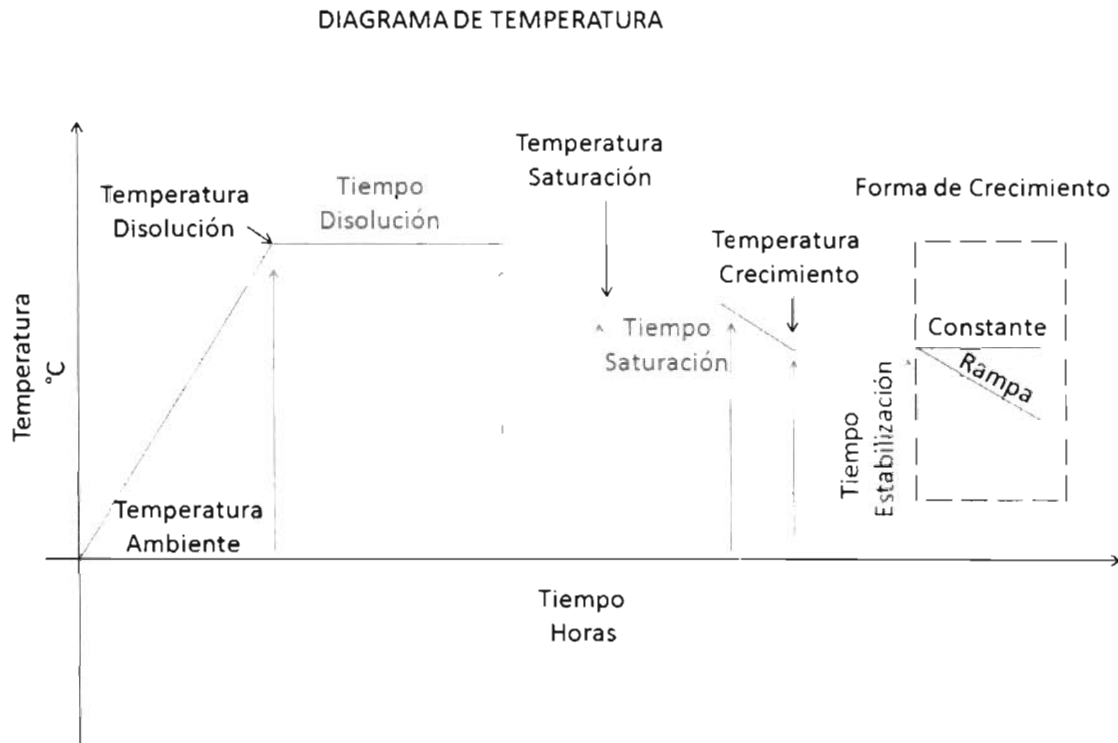


Figura 2.6. Diagrama de temperatura generalizado para la realización del crecimiento. Hay que señalar que los tiempos y temperaturas varían de acuerdo a los tipos de materiales y objetivos a lograr en el experimento

A continuación se describe de una forma breve la manera en la cual se realiza el crecimiento de cristales por el método de epitaxia en fase líquida: El reactor se localiza dentro de un horno, con ayuda del cual se calienta la solución hasta la temperatura deseada. Al alcanzar la temperatura necesaria, el deslizador se desliza en relación con el tronco el cual contiene el sustrato y de esta manera el sustrato se pone en contacto con cada una de las fases líquidas contenidas en las cavidades del tronco. Si al mismo tiempo se baja la temperatura del horno, entonces sobre la superficie del sustrato se van a cristalizar capas epitaxiales; la composición de estas capas depende de la temperatura y de las composiciones de las fases líquidas.

Naturalmente que para crecer estas películas epitaxiales de diferentes materiales es necesario conocer sus diagramas de temperatura, según sea el caso.



En la actualidad, estos diagramas han sido estudiados con detalle tanto experimental como teóricamente para casi todos los sistemas de materiales que representan algún interés práctico. Conociendo los datos de los diagramas de temperatura que establecen la relación de composiciones entre las fases líquida y sólida a diferentes temperaturas, se pueden regular las propiedades físicas de las capas crecidas y de las estructuras de capas múltiples en general. El espesor de algunas de las capas se puede regular variando el intervalo de la temperatura en el que se realiza el proceso de cristalización o bien variando el volumen de la fase líquida.

Una vez que se ha terminado el experimento se baja la temperatura del horno hasta cero grados esperando a que la temperatura dentro sistema llegue a los 80 grados para poder apagar el sistema terminando de esta forma el experimento.

### **2.3 Parámetros físicos involucrados.**

Las variables físicas a monitorear son las siguientes:

- **Flujo de Hidrógeno:** es importante monitorear durante todo el proceso el flujo de hidrógeno, ya que este no se debe de hacerse cero o bajar del valor predefinido por el usuario, si esto sucediera entonces se debe detener el proceso ya que esto indicaría una falla en el sistema, ya sea una fuga, una válvula dañada o alguna otra falla que sería difícil de determinar en ese momento lo cual implicaría errores en el crecimiento por lo que es necesario apagar el sistema.

Además se utiliza para purificar al sistema tal como se mencionó ya anteriormente y este tipo de crecimiento se realiza bajo una atmósfera de hidrógeno.

- **Presión:** es necesario adquirir la presión que se genera al hacer vacío con la bomba Jet y las bombas de adsorción para saber el momento en el cual debe concluir esta etapa, además cuando se realiza el flujo de hidrógeno se debe vigilar que la presión interna en el sistema no suba por arriba de los 16 PSI ya que esto provocaría una presión excesiva dentro del reactor de tubo de cuarzo provocando que este se rompa lo cual sería un peligro para el usuario y el sistema.

- **Temperatura Interna:** Es importante no solo monitorear sino también controlar ya que de esta depende el crecimiento del cristal. Las temperaturas a controlar y monitorear son las siguientes: temperatura de disolución que es en la cual los cristales se disuelven en el solvente, una vez que se han disuelto totalmente los cristales se baja la temperatura unos cuantos grados esto con el fin de saturar la solución y posteriormente se coloca el horno en la temperatura de crecimiento que es aquí donde se realiza la deposición de los cristales en el sustrato.
- **Temperatura Ambiente:** este parámetro físico se mide con el objetivo de poder analizar cómo es que afecta la temperatura ambiente en el desarrollo de los experimentos, este es uno de los objetivos de la tesis, mejorar la reproducibilidad, conociendo precisamente las condiciones óptimas para la realización de los experimentos.
- **Humedad relativa dentro del laboratorio:** esta variable física también es de vital importancia monitorear ya que se desea saber cómo afecta en el proceso de crecimiento. Sobre todo a la hora en la que se abre el reactor para introducir el material, se desea saber los niveles de humedad en ese momento, con el fin de observar cómo afecta está a la hora de hacer fluir el hidrógeno en el sistema para mejorar los niveles de humedad interna en el sistema.
- **Humedad interna en el sistema:** esta es probablemente una de las variables físicas más importantes involucradas en el proceso, ya que entre valores más bajos se tengan se podrá obtener mejores resultados, y es precisamente lo que se quiere hacer con la adquisición y monitoreo de todas las demás variables, ver cómo es afectada la humedad interna en función del estado de los otros parámetros.

## **CAPÍTULO III**

### **HARDWARE DE CONTROL**

#### **3.1 Circuito de Potencia**

El circuito de potencia es el encargado de accionar las válvulas neumáticas, es a través de la apertura de estas válvulas, como se llevan a cabo la ejecución de los diferentes procesos involucrados en el sistema, por ejemplo al accionar las válvulas uno, tres, cinco y ocho se realiza el flujo de hidrógeno para purificar el sistema y hacer el crecimiento bajo una atmosfera de hidrogeno ultra purificado.

Los principales componentes utilizados en el circuito de potencia para el accionamiento de las válvulas son los siguientes.

- Electroválvulas marca Festo.
- Transistor de Potencia "TIP32C".
- Transistor de señal "2N2222A".
- Diodo 1N4007.
- Resistencias.

En la figura 3.1 se muestra el diagrama electrónico del circuito de potencia implementado en el sistema.

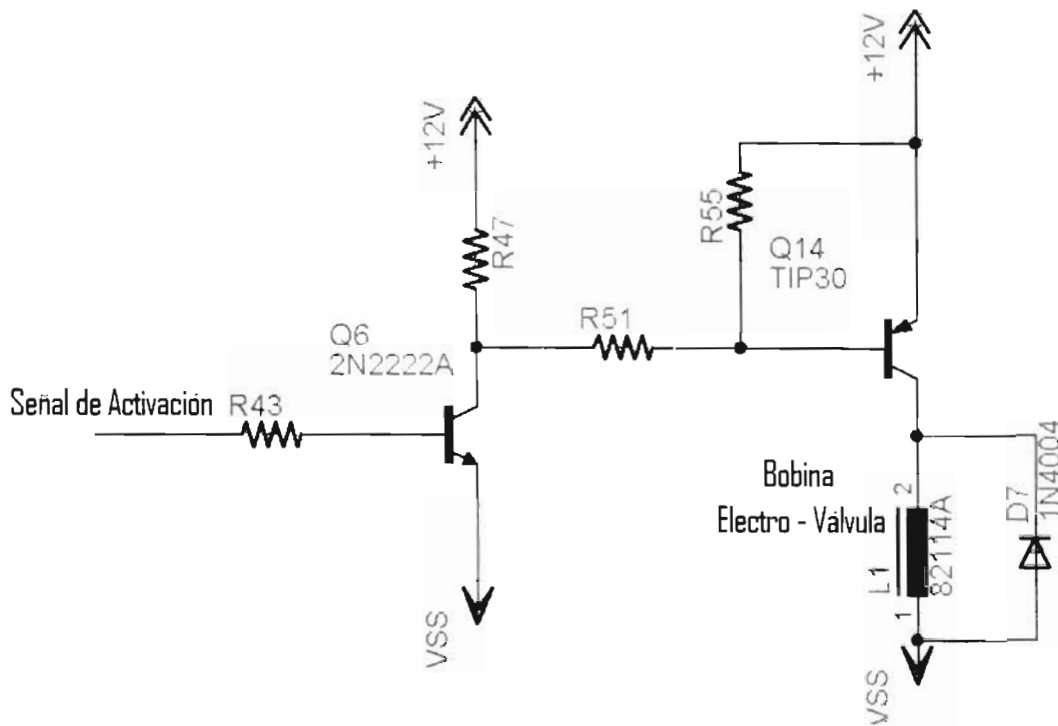


Figura 3.1 Diagrama electrónico de potencia implementado.

En el diagrama electrónico tenemos R43 que es la resistencia de base del transistor de señal Q6 (2N2222A) que está conectado a la salida de la compuerta OR que manda la señal de activación de la válvula a accionar, esta señal puede provenir de uno de los pines del microcontrolador o de uno de los switch utilizados en el modo de operación manual. Cabe mencionar que tanto el transistor de señal como el transistor de potencia se utilizan en el modo de operación de switch, es decir, se estarán trabajando en las regiones de corte y saturación. Por lo cual al haber un uno lógico en la base del transistor de señal habrá un cero lógico en la base del transistor de potencia teniendo doce volts en el colector con lo cual se activa la electroválvula (L1) que está conectada a tierra, el diodo D7 (1N4007) sirve de protección a la electroválvula.

Para desactivar la válvula neumática simplemente se envía un cero lógico a la base del transistor de potencia. Este circuito se implementó para todas la válvulas neumáticas utilizadas en los dos sistemas de crecimiento.

### 3.2 Circuito de control electrónico

Este se encarga de controlar el funcionamiento de los circuitos de potencia, adquisición de señales y comunicación con la PC por medio del puerto USB.

El dispositivo encargado de realizar el control de la tarjeta es el microcontrolador PIC18F4550. Se eligió este dispositivo debido a que cuenta con un módulo de comunicación USB interno, con el cual se realiza la comunicación con la computadora y se realiza el intercambio de información entre el PIC y la PC, es decir, el PIC recibe los comandos que se le envían desde la PC y la PC recibe la información que le solicita al microcontrolador.

A continuación se dará una breve explicación de las características que presenta este microcontrolador de Microchip.

En la figura 3.2 se muestra la imagen del PIC18F4550 con sus respectivos pines.

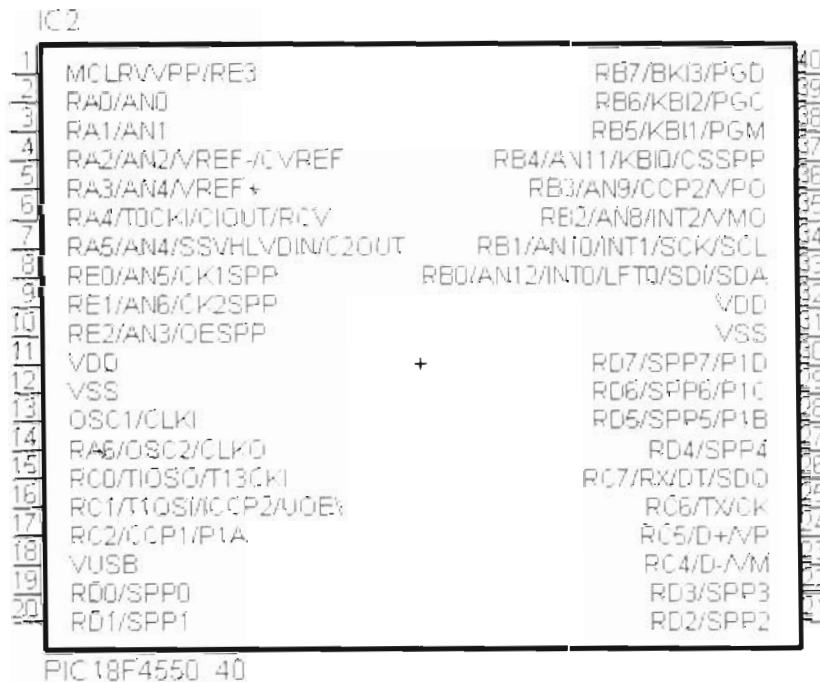


Figura 3.2. PIC18F4550

Entre las características principales del PIC18F4550 se encuentran las siguientes:

- Arquitectura RISC avanzada Harvard: 16 bits con 8 bits de datos.
- Set de 77 instrucciones.
- 40 pines.
- Memoria de programa flash de 64 Kb.
- Memoria de datos RAM 3968 bytes.
- Memoria de datos EEPROM 1 Kb.
- Frecuencia máxima de reloj de 40 MHz
- Múltiples fuentes de interrupción.
- Periféricos de comunicación avanzados (USB y CAN).
- 5 Puertos E/S A, B, C, D y E.
- Módulos de PWM.
- Líneas de entrada para el ADC de 8 o 10 bits.
- Pila de 32 niveles.
- Módulo de comunicación serie.

El microcontrolador PIC18F4550 dispone de 5 puertos de entrada/salida que incluyen un total de 35 líneas digitales de E/S.

Todas las líneas digitales de entrada/salida disponen de al menos una función alternativa asociada a alguna circuitería específica del microcontrolador. Cuando una línea trabaja en el modo alternativo no puede ser utilizada como línea digital de E/S estándar. En la figura 3.3 se muestra el diagrama a bloque del PIC18F4550.

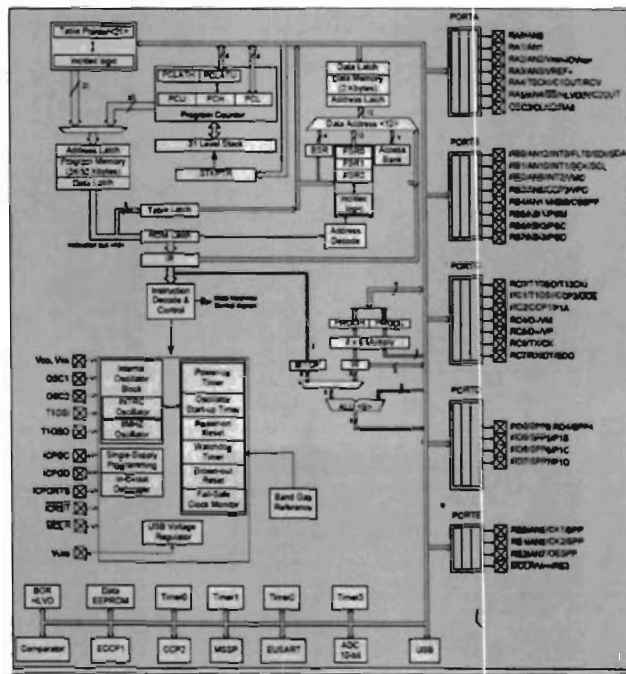


Figura 3.3 Diagrama a bloques del PIC18F4550.

En la figura 3.4 se muestra el esquema del circuito de control.

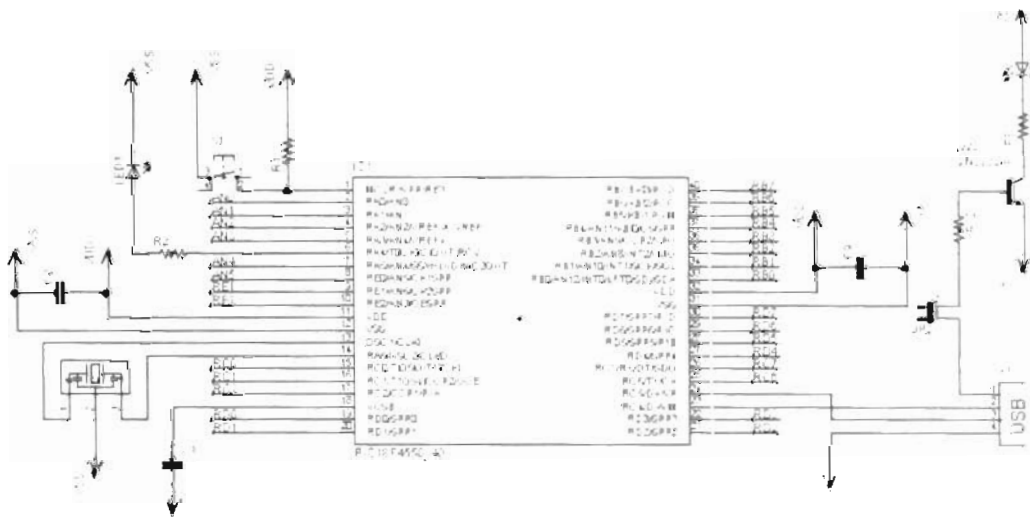


Figura 3.4 Circuito de control.

### 3.3 Circuitos de Adquisición de Señales

Para la medición de la temperatura ambiente se utilizó el sensor "TMP36", sus condiciones de operación son las siguientes: provee un voltaje de salida lineal a la temperatura medida, esto es en grados centígrados. El rango de operación es de -40°C a 125 °C, a 25°C se tiene una señal de 750 mv en la salida del sensor. El rango de la fuente de alimentación es de 2.7 V hasta 5.5 V máximo, el sensor se muestra en la figura 3.5.

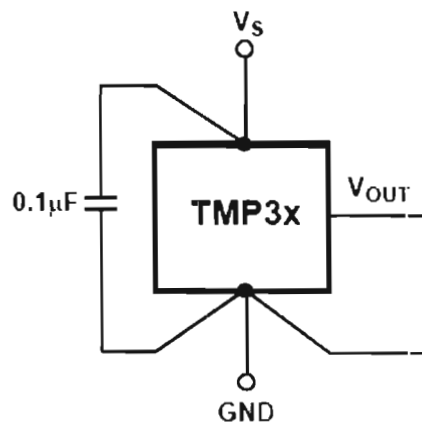


Figura 3.5 Sensor de temperatura TMP36.

Es necesario conectar un capacitor de 0.1 uf en las terminales alimentación del sensor para eliminar ruido tal como se muestra en la anterior figura.

El acondicionamiento de la señal de salida del sensor de temperatura se realiza por medio del amplificador operacional OP491 que está configurado como seguidor unitario. Esta configuración lleva a cabo dos funciones: la primera es la de aislar al sensor, esto se logra debido a la alta impedancia de entrada que ofrece la configuración de seguidor unitario y la segunda es la de evitar que en un momento dado se pudiera caer la señal de medición de temperatura al entrar al convertidor.

Para el cálculo de la temperatura en función del voltaje a la salida del sensor se utilizó la fórmula uno, la cual se obtuvo en forma empírica.

$$Temperatura = \left( \frac{\text{voltaje}}{0.01} \right) - 60 \quad (1)$$



Para la medición de la humedad relativa en el laboratorio se utilizó el sensor "HIH-4000-001". Este sensor tiene una señal de voltaje de salida lineal. Requiere de una fuente de alimentación de alrededor de 4 V a 5.8 V con un consumo de corriente máxima de 500 uA. El rango de operación va de 0 % al 100 % de humedad, operando en un rango de temperatura de - 40°C a 85°C. El voltaje de salida correspondiente a la humedad relativa medida es de 0.981 V al 0% de HR hasta 3.9 V 100 % de HR.

Los circuitos de medición de humedad y temperatura se muestran en la figura 3.6, el pin de salida del sensor se conecta a la entrada no inversora del amplificador operacional OP491 el cual está configurado como seguidor unitario el cual básicamente tiene la función de aislar al sensor.

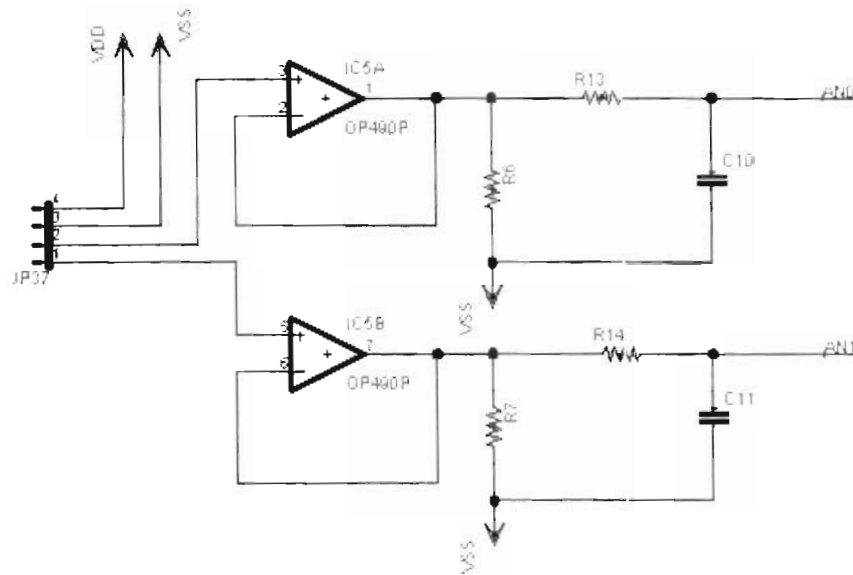


Figura 3.6 Circuito de adquisición de temperatura y humedad dentro del laboratorio

La señal de salida del sensor de temperatura se conecta en el PIN2 de JP37 y la señal del sensor de humedad se conecta en el PIN1 de JP37, como se ve en esquema, el PIN2 se conecta al operacional IC5A y el PIN1 al operacional IC5B respectivamente.

Para el cálculo de la humedad en el laboratorio se utilizó la fórmula dos que es proporcionada por el fabricante.

$$HumedadRelativa = \frac{1}{0.0062} \left( \frac{VoltajeSalida}{VoltajeAlimentación} - 0.16 \right) \quad (2)$$

Para la medición de la temperatura dentro del sistema se utilizan los controladores – programadores “Eurotherm 818” y el “OMRON E5CN” para los sistemas uno y dos respectivamente. Estos controladores no solo se utilizan para leer la temperatura actual del sistema sino que además se utilizan para controlar la temperatura deseada dentro de este, es decir, es por medio de estos dispositivos que se inyecta la corriente necesaria para calentar el horno a las temperaturas de disolución, saturación y crecimiento. Las principales características de estos son la siguientes: proveen una salida de control de 4 – 20 mA requerida por la etapa de potencia conectada al horno, tarjeta de comunicación con la PC, protocolo RS232, control PID y cinco dígitos de resolución en el control de temperatura.

Para la medición de las presiones generadas cuando se realiza vacío y flujo de hidrógeno se utilizan los siguientes sensores: el sensor de la serie PC 302A-VR marca PURERON de alta pureza con el cual se mide el vacío generado en el sistema en el rango de 15 PSI a 0 PSI esto cuando se utiliza la bomba Jet, además se cuenta con un indicador analógico modelo PSD-02A también marca PURERON con el cual también se monitorea este valor de presión así como el valor de presión que se genera dentro del sistema cuando se hace fluir hidrógeno; hay que mencionar que este indicador dispone de una señal de salida analógica de voltaje que se utiliza para desplegar el valor de presión en la PC a través de la adquisición de esta señal por medio del PIC18F4550. Este indicador se utiliza en el sistema uno, para el dos, se utiliza un indicador marca Eurotherm modelo 2408i este también cuenta con una señal de salida analógica que se adquiere y convierte con el PIC18F4550 para ser transmitido y desplegado en la PC, el sensor en ambos sistemas es el mismo.

Cuando se utilizan las bombas de Adsorción para hacer vacío se utiliza el sensor Thermocouple Vacuum Gauge modelo 1518 el cual es capaz de medir vacíos más finos, estos valores van de 0 a 1000 mili Torr. Para visualizar este valor se utiliza un indicador marca MDC modelo CT el cual es capaz de leer estos valores de presión algo que el indicador PURERON ya no es capaz de hacer. Este indicador también cuenta con una señal de salida analógica que es adquirida por medio del microcontrolador para ser desplegada en la PC.

Los circuitos de adquisición de estas señales se muestran en la figura 3.7, el diagrama que se encuentra en la parte superior corresponde al utilizado para adquirir el valor generado por la bomba Jet y el de la parte inferior es el que adquiere la señal que se genera cuando se utiliza la bomba de Adsorción.

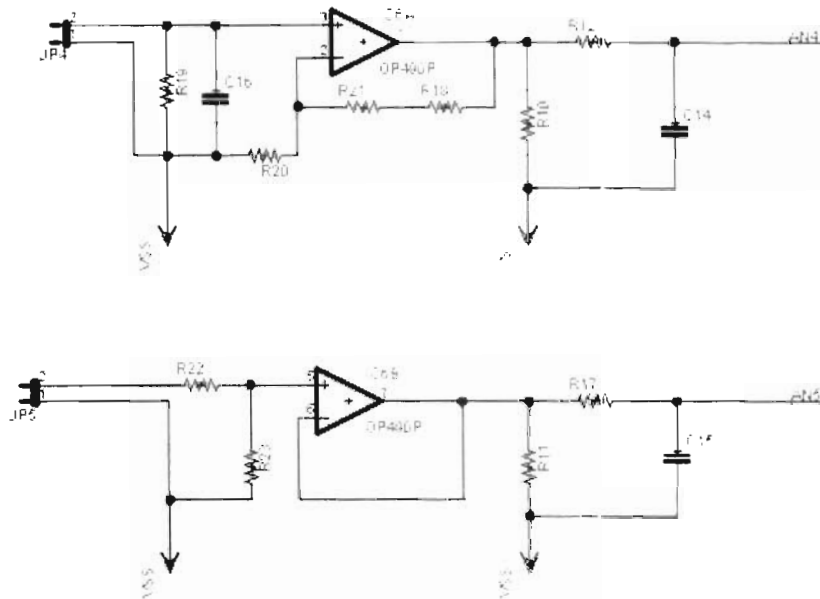


Figura 3.7 Circuito de adquisición.

Para obtener el valor en PSI una vez que se adquirió la señal entregada por sensor PC302A-VR a través del indicador PURERON por medio del microcontrolador hacia la PC, se hace uso de la ecuación tres la cual se obtuvo en forma empírica, es decir, se tomó la lectura de voltaje en los puntos de máxima y mínima presión en el sistema, al ser lineal el sensor, se introdujeron las señales leídas en la ecuación de la recta y se calculó la pendiente con lo cual se obtuvo la fórmula tres.

$$\text{vacíoJet} = (\text{voltaje} * -5.7) + 100 \quad (3)$$

En lo que respecta a la obtención del parámetro físico en mili Torr para los valores de voltajes que se presentan al utilizar las bombas de Adsorción se hace uso de la fórmula cuatro, esta se obtuvo en base a una tabla de voltajes contra vacío en mili torr proporcionada por el fabricante, estos datos se graficaron y ajustaron mediante un programa para matemáticas, el cual arrojó precisamente la ecuación cuatro, con la cual se obtenía el mejor ajuste de los valores graficados.

$$\text{vacíoabsorción} = \left( a_1 * \exp^{-\text{voltaje}/\lambda_1} \right) + \left( a_2 * \exp^{-\text{voltaje}/\lambda_2} \right) + \left( a_3 * \exp^{-\text{voltaje}/\lambda_3} \right) + y_0 \quad (4)$$

Donde  $a_1, a_2, a_3, t_1, t_2, t_3$  y  $y_0$  son constantes.

En lo que respecta a la medición de flujo de hidrógeno se utiliza el sensor UFC 1400 el cual entrega una señal de salida de 0 a 5 volts en el rango de 0 a 100% de flujo, este dispositivo cuenta con una válvula que abre en cierto porcentaje de acuerdo al valor de flujo que se desea, la válvula abre del 0 a 100 % de acuerdo al voltaje que se le aplique, y puede variar de 0 a 5 volts en función del porcentaje deseado. Este sensor también cuenta con una terminal de salida que se conecta al sistema de adquisición del PIC18F4550 para su visualización en la PC.

El diagrama electrónico de adquisición se muestra en la figura 3.8.

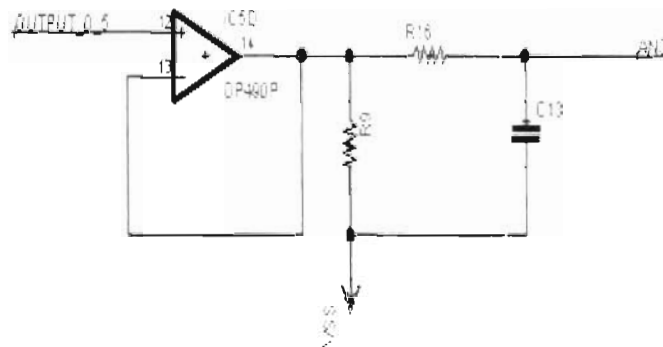


Figura 3.8. Diagrama eléctrico de adquisición de flujo de Hidrógeno.

La fórmula utilizada para convertir del voltaje adquirido a su valor correspondiente en porcentaje de flujo de hidrógeno leído por el sensor y desplegado en la PC es la fórmula cinco.

$$flujo = \frac{voltajesalida * 100}{voltajedereferencia} \quad (5)$$

En lo que respecta a la medición de la humedad interna en el sistema se utiliza el Moisture Monitor Series 3 que al igual que los demás sensores e indicadores este también cuenta con una señal de salida analógica que se conecta al circuito de adquisición mostrado en la figura 3.9.

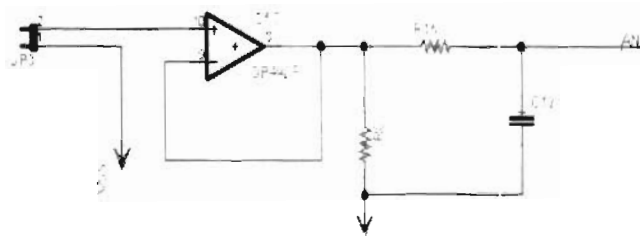


Figura 3.9. Circuito de adquisición de la humedad interna.

En lo que respecta a la conversión de voltaje a parámetro físico leído se utiliza la fórmula seis. Esta fórmula se obtuvo en base a la calibración del sensor, es decir, se definieron los rangos de lectura máximo y mínimo, en base a esto se obtuvo la resolución del sensor y se calculó la fórmula.

$$\text{humedadinterna} = \left( \frac{\text{voltajesalida}}{0.02} \right) + [-110] \quad (6)$$

### 3.4 Fuentes de Alimentación

Para la alimentación de los componentes se utiliza una fuente de 5 V. En la figura 3.10 se muestra el esquema que compone la fuente de alimentación para la tarjeta de control. Además se implementó una fuente de  $\pm 15V$  para alimentar el controlador de flujo de hidrógeno, que requiere estos voltajes de alimentación. El transformador baja de 127 volts de alterna a 24 volts de alterna y estos son rectificadas para obtener los voltajes de CD que serán regulados por los componentes 7805, 7815 y 7915 para obtener los voltajes respectivos de alimentación.

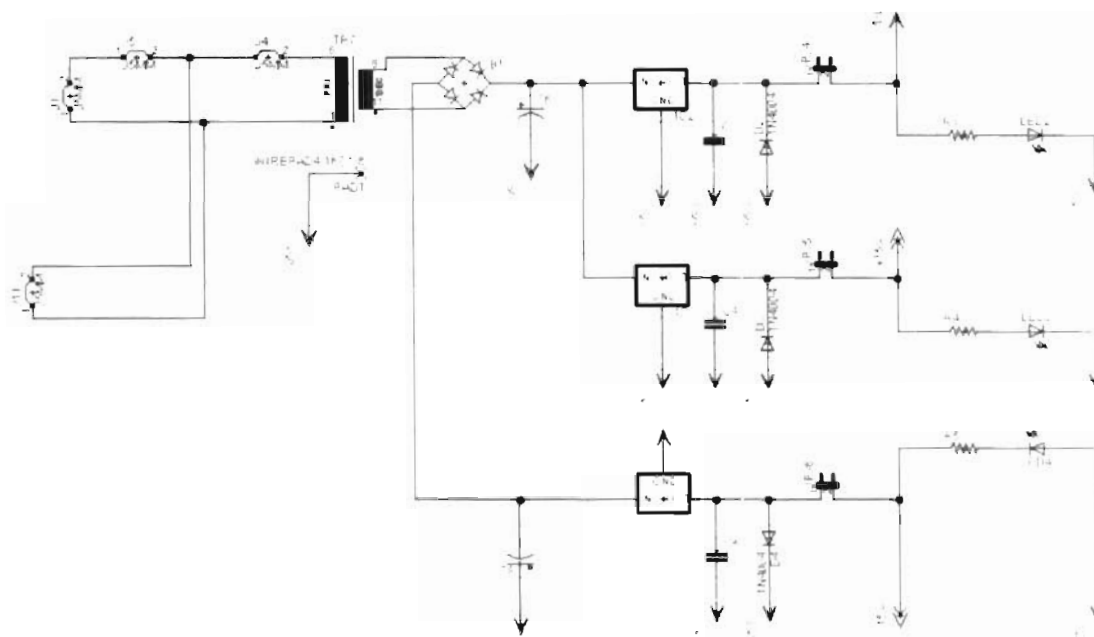


Figura 3.10. Fuente de alimentación.

### 3.5 USB

El **Universal Serial Bus** (bus universal en serie) o **Conductor Universal en Serie (CUS)**, abreviado comúnmente **USB**, es un puerto que sirve para conectar periféricos a una computadora. Fue creado en 1996 por siete empresas: IBM, Intel, Northern\_Telecom, Compaq, Microsoft, Digital\_Equipment\_Corporation y NEC.

El estándar incluye la transmisión de energía eléctrica al dispositivo conectado. Algunos dispositivos requieren una potencia mínima, así que se pueden conectar varios sin necesitar fuentes de alimentación extra. La gran mayoría de los concentradores incluyen fuentes de alimentación que brindan energía a los dispositivos conectados a ellos, pero algunos dispositivos consumen tanta energía que necesitan su propia fuente\_de\_alimentación. Los concentradores con fuente de alimentación pueden proporcionarle corriente eléctrica a otros dispositivos sin quitarle corriente al resto de la conexión (dentro de ciertos límites).

El diseño del USB tenía en mente eliminar la necesidad de adquirir tarjetas separadas para poner en los puertos bus\_ISA o PCI, y mejorar las capacidades plug and play permitiendo a esos dispositivos ser conectados o desconectados al sistema sin necesidad de reiniciar. Cuando se conecta un nuevo dispositivo, el servidor lo enumera y agrega el software necesario para que pueda funcionar.

El USB puede conectar los periféricos como ratones, teclados, escáneres, cámaras digitales, teléfonos móviles, reproductores multimedia, impresoras, discos duros externos, tarjetas de sonido, sistemas de adquisición de datos y componentes de red. Para dispositivos multimedia como escáneres y cámaras digitales, el USB se ha convertido en el método estándar de conexión. Para impresoras, el USB ha crecido tanto en popularidad que ha desplazado a un segundo plano a los puertos paralelos porque el USB hace mucho más sencillo el poder agregar más de una impresora a una computadora personal.

En el caso de los discos duros, es poco probable que el USB reemplace completamente a los buses (el ATA (IDE) y el SCSI), pues el USB tiene un rendimiento más lento que esos otros estándares. Sin embargo, el USB tiene una importante ventaja en su habilidad de poder instalar y desinstalar dispositivos sin tener que abrir el sistema, lo cual es útil para dispositivos de almacenamiento externo. Hoy en día, una gran parte de los fabricantes ofrece dispositivos USB portátiles que ofrecen un rendimiento casi indistinguible en comparación con los ATA (IDE). Por el contrario, el nuevo estándar Serial ATA permite tasas de transferencia de hasta aproximadamente 150/300 MB por segundo, y existe también la posibilidad de extracción en caliente e incluso una especificación para discos externos llamada eSATA.

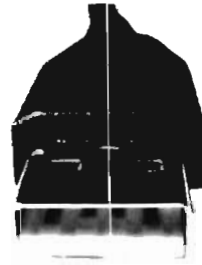


Figura 3.11 Símbolo de USB y conector USB tipo A. macho.

### 3.5.1 Características de Transmisión

Los dispositivos USB se clasifican en cuatro tipos según su velocidad de transferencia de datos:

- Baja velocidad (1.0): Tasa de transferencia de hasta 1'5 Mbps (192 KB/s). Utilizado en su mayor parte por dispositivos de interfaz humana (*Human interface device*, en inglés) como los teclados, los ratones y los joysticks.
- Velocidad completa (1.1): Tasa de transferencia de hasta 12 Mbps (1'5 MB/s). Ésta fue la más rápida antes de la especificación USB 2.0, y muchos dispositivos fabricados en la actualidad trabajan a esta velocidad. Estos dispositivos dividen el ancho de banda de la conexión USB entre ellos, basados en un algoritmo de búferes FIFO.
- Alta velocidad (2.0): Tasa de transferencia de hasta 480 Mbps (60 MB/s).
- Súper alta velocidad (3.0): Actualmente en fase experimental y con tasa de transferencia de hasta 4.8 Gbps (600 MB/s). Esta especificación será lanzada a mediados de 2009 por Intel, de acuerdo con información recabada de Internet. La velocidad del bus será diez veces más rápida que la del USB 2.0, debido a la que han incluido 5 conectores extra, desechando el conector de fibra óptica propuesto inicialmente, y será compatible con los estándares anteriores. Se espera que los productos fabricados con esta tecnología lleguen al consumidor en 2009 o 2010.<sup>12</sup>



**Pin Nombre Color del cable Descripción**

1	VCC	Rojo	+5v
2	D-	Blanco	Data -
3	D+	Verde	Data +
4	GND	Negro	Tierra

Figura 3.12 Señales del conector USB

Las señales del USB se transmiten en un cable de par trenzado con impedancia de  $90 \Omega \pm 15\%$ , cuyos hilos se denominan D+ y D-. Estos, colectivamente, utilizan señalización diferencial en half dúplex para combatir los efectos del ruido electromagnético en enlaces largos. D+ y D- suelen operar en conjunto y no son conexiones simples. Los niveles de transmisión de la señal varían de 0 a 0'3 V para bajos (ceros) y de 2'8 a 3'6 V para altos (unos) en las versiones 1.0 y 1.1, y en  $\pm 400$  mV en alta velocidad (2.0). En las primeras versiones, los alambres de los cables no están conectados a masa, pero en el modo de alta velocidad se tiene una terminación de  $45 \Omega$  a tierra o un diferencial de  $90 \Omega$  para acoplar la impedancia del cable. Este puerto sólo admite la conexión de dispositivos de bajo consumo, es decir, que tengan un consumo máximo de 100 mA por cada puerto; sin embargo, en caso de que estuviese conectado un dispositivo que permite 4 puertos por cada salida USB (extensiones de máximo 4 puertos), entonces la energía del USB se asignará en unidades de 100 mA hasta un máximo de 500 mA por puerto. Los cables de datos son un par trenzado para reducir el ruido y las interferencias.



Figura 3.13. Cable de datos USB

### 3.5.2 Compatibilidad y conectores

El estándar USB especifica tolerancias para impedancia y de especificaciones mecánicas relativamente bajas para sus conectores, intentando minimizar la incompatibilidad entre los conectores fabricados por distintas compañías —una meta a la que se ha logrado llegar. El estándar USB, a diferencia de otros estándares también define tamaños para el área alrededor del conector de un dispositivo, para evitar el bloqueo de un puerto adyacente por el dispositivo en cuestión.

Las especificaciones USB 1.0, 1.1 y 2.0 definen dos tipos de conectores para conectar dispositivos al servidor: A y B. Sin embargo, la capa mecánica ha cambiado en algunos conectores. Por ejemplo, el IBM UltraPort es un conector USB privado localizado en la parte superior del LCD de las computadoras portátiles de IBM. Utiliza un conector mecánico diferente mientras mantiene las señales y protocolos característicos del USB. Otros fabricantes de artículos pequeños han desarrollado también sus medios de conexión pequeños, y ha aparecido una gran variedad de ellos, algunos de baja calidad.

Una extensión del USB llamada "USB-On-The-Go" (*sobre la marcha*) permite a un puerto actuar como servidor o como dispositivo - esto se determina por qué lado del cable está conectado al aparato. Incluso después de que el cable está conectado y las unidades se están comunicando, las 2 unidades pueden "cambiar de papel" bajo el control de un programa. Esta facilidad está específicamente diseñada para dispositivos como PDA, donde el enlace USB podría conectarse a un PC como un dispositivo, y conectarse como servidor a un teclado o ratón. El "USB-On-The-Go" también ha diseñado 2 conectores pequeños, el mini-A y el mini-B, así que esto debería detener la proliferación de conectores miniaturizados de entrada.



5 cm

Figura 3.14. Tipos diferentes de conectores USB (de izquierda a derecha): micro USB macho, mini USB tipo B macho, Tipo B macho, Tipo A hembra, Tipo A macho

### 3.5.3 PIC18F4550 y USB

El USB es directamente soportado por el microcontrolador PIC18F4550. De esta forma se permite que el microcontrolador sea interconectado a una PC por medio del USB para virtualmente cualquier uso que necesite este tipo de conectividad. Se utilizan librerías ya predefinidas en el compilador de PICC de la compañía Custom Computer Services CCS.

Para llevar a cabo la comunicación se hace uso de un driver y una DLL desarrollados por microchip. El driver contiene todas las características del dispositivo que se va a conectar en este caso el microcontrolador, mientras que la DLL contiene todas las funciones necesarias para comunicar al dispositivo con una aplicación en cuyo caso consiste el programa implementado en LabVIEW del cual se hablará en el siguiente capítulo.

Es necesario utilizar este driver de instalación para el reconocimiento del nuevo dispositivo conectado (PIC18F4550) debido a que se está trabajando la interfase USB a una alta velocidad, en este modo de trabajo es necesaria la instalación del driver, el cual se instala solo la primera vez que se conecta el dispositivo. Cabe señalar que tanto la DLL como el driver utilizados son compatibles con Windows XP.

La figura 3.15 ilustra la estructura interna de la interfase USB con la que cuenta el microcontrolador PIC18F4550 y en la figura 3.16 se muestra la interfase entre la PC y el PIC18F4550 por medio del conector USB.

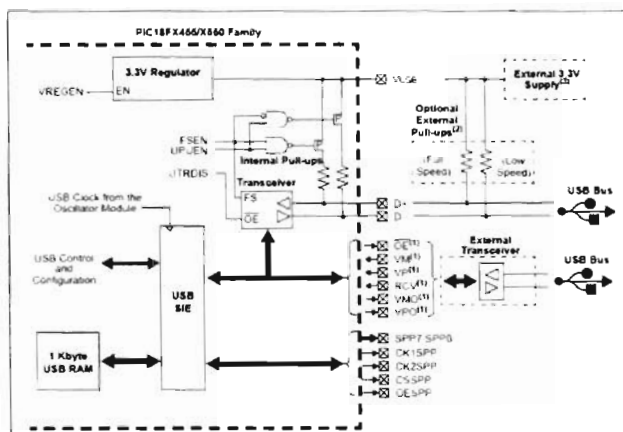


Figura 3.15. Estructura interna del módulo USB.

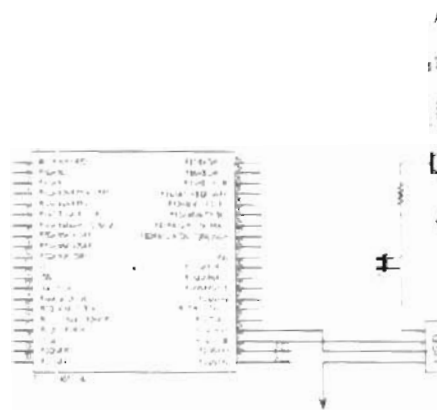


Figura 3.16. PIC18F4550 conectado al puerto USB.

### 3.6 Circuito impreso

En la figura 3.17 se muestra el circuito impreso de la tarjeta de control del sistema uno y en la figura 3.18 la tarjeta de control del sistema de crecimiento dos, como se puede notar ambas tarjetas son casi idénticas.

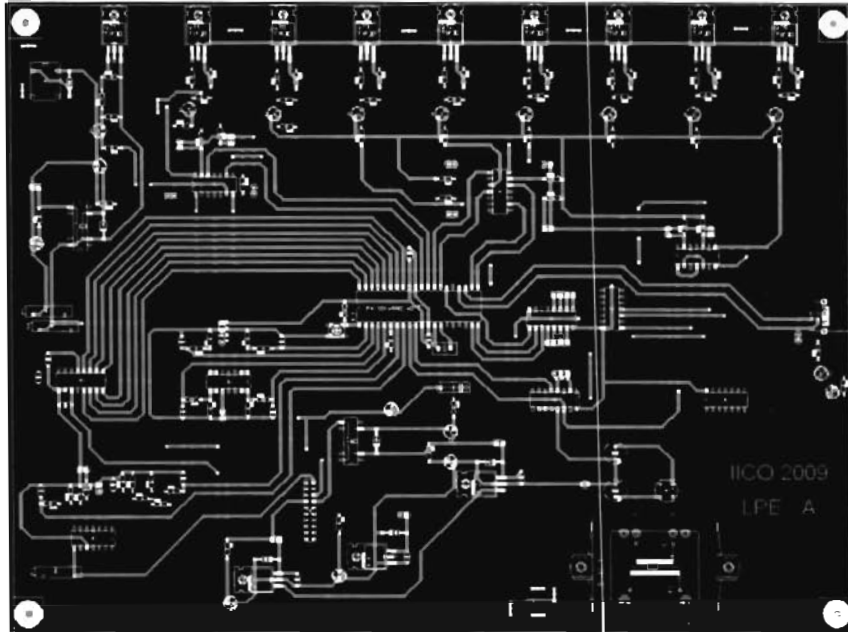


Figura 3.17 Tarjeta de control del sistema uno.

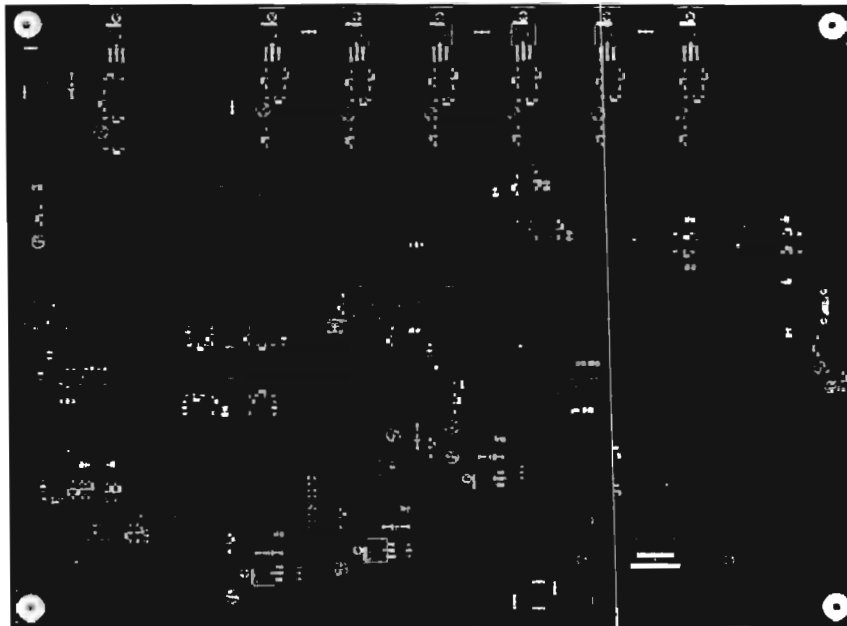


Figura 3.18. Tarjeta de control del sistema dos.

En la figura 3.19 y 3.20 se muestran las tarjetas de control implementadas de los sistemas uno y dos de crecimiento de cristales por el método de epitaxia en fase líquida.

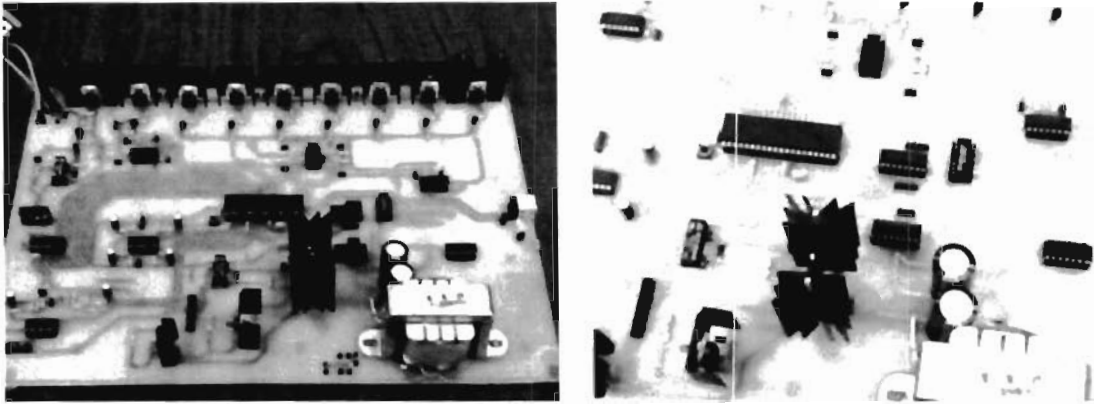


Figura 3.19. Tarjeta de control del sistema de crecimiento de cristales uno.

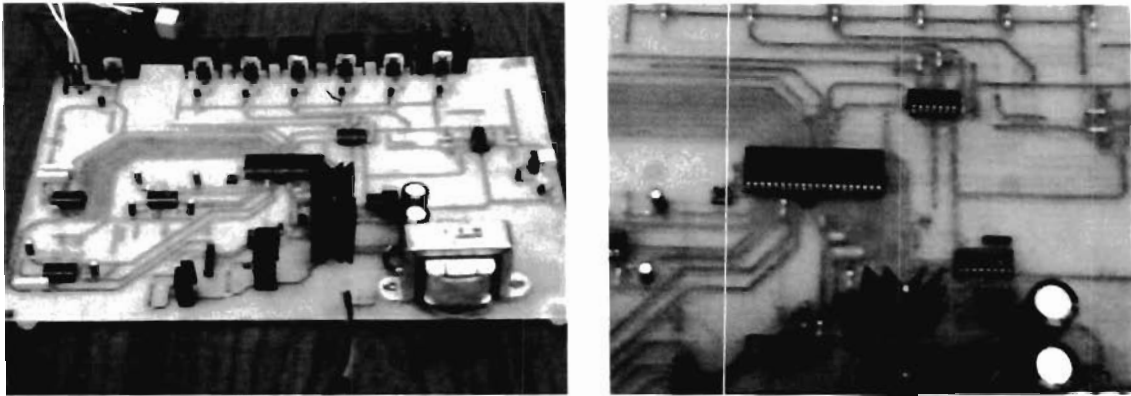


Figura 3.20. Tarjeta de control del sistema de crecimiento de cristales dos.

En la figura 3.21 se muestra un diagrama a bloques general del hardware de control implementado en el sistema de crecimiento de cristales.

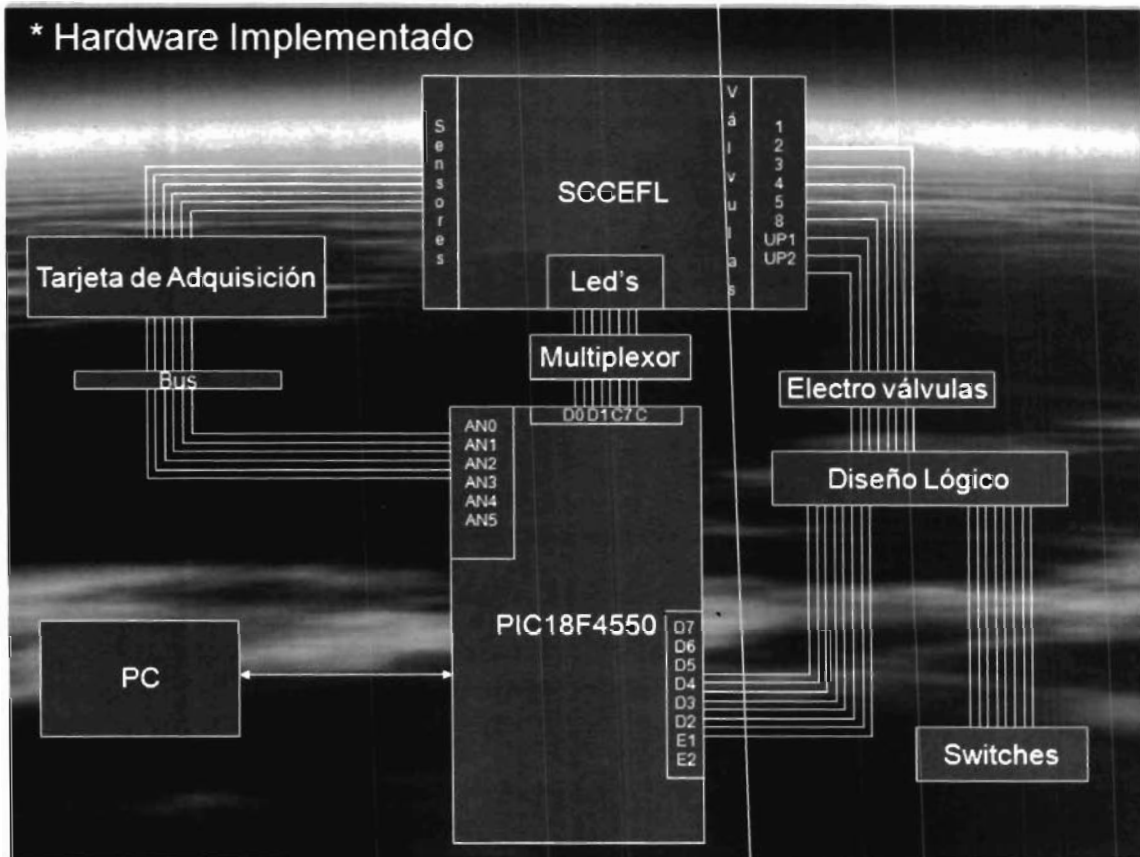


Figura 3.21. Diagrama a bloques del Hardware de control.

## CAPÍTULO IV

### SOFTWARE DE CONTROL

#### 4.1 Programa desarrollado en el microcontrolador

En esta sección se dará una breve explicación del programa desarrollado en el microcontrolador PIC18F4550, el cual se utiliza como una interfase entre el sistema y la PC, es decir, de acuerdo a los comandos enviados desde la PC el microcontrolador ejecutará la acción indicada en el comando la cual puede ser adquirir una señal física por medio del convertidor análogo/digital, accionar las válvulas neumáticas a través del circuito de potencia para comenzar a realizar alguno de los procedimientos involucrados en el sistema o alguna otra acción de acuerdo al comando a ejecutar.

El programa se realizó en lenguaje C, para lo cual se utilizó el "PIC C compiler", en este compilador se puede programar el microcontrolador en un lenguaje de alto nivel como lo es el lenguaje de programación "C". El "PIC C compiler" fue desarrollado para cumplir con las especificaciones del lenguaje ANSI C.

En la figura 4.1 se muestra el diagrama de flujo del programa principal del PIC, a continuación se explica el desarrollo del programa:

1. Inicio: En esta parte se declaran todas las directivas a utilizar tales como archivos incluidos como lo son las funciones para operaciones matemáticas, el archivo "#include <18F4550.h>" que contiene declarados todos los registros internos del microcontrolador, además de las librerías para configuración del módulo USB, define la resolución a utilizar en el convertidor análogo/digital, declara los fusibles de programación del microcontrolador, variables globales a usar en el programa y las constantes a usar por el preprocesador.
2. Inicializa el Microcontrolador PIC18F4550: En esta parte se configura la forma en la que trabajaran los puertos, es decir, cuales trabajarán como puertos de entrada y cuales como puerto de salida, configura el módulo interno del convertidor análogo/digital.
3. Configura el módulo interno USB para ser utilizado: Esta parte es muy importante, es aquí donde se mandan llamar las funciones encargadas de configurar e inicializar el módulo USB; dichas funciones son las siguientes: la función `usb_init()` que es la encargada de configurar el módulo interno para la comunicación USB, esta es la que prepara todo las partes físicas internas del PIC para que realicen la función de la comunicación USB; la

función `usb_task()` que es la que distribuye todas las funciones a realizar durante la comunicación USB y finalmente la función `usb_wait_for_enumeration()` que es la que nos indica cuando la PC ya reconoció y enumeró al dispositivo.

4. Dispositivo enumerado: Cuando el dispositivo ya fue enumerado y reconocido por la PC la función `usb_wait_for_enumeration()` manda un valor booleano (TRUE o FALSE) a la función `usb_enumerated()` de esta forma si esta función contiene un "TRUE" quiere decir que el dispositivo fue enumerado y se continua con la siguiente parte del programa de lo contrario tenemos un "FALSE" y el programa se queda esperando a que el dispositivo sea enumerado por la PC.

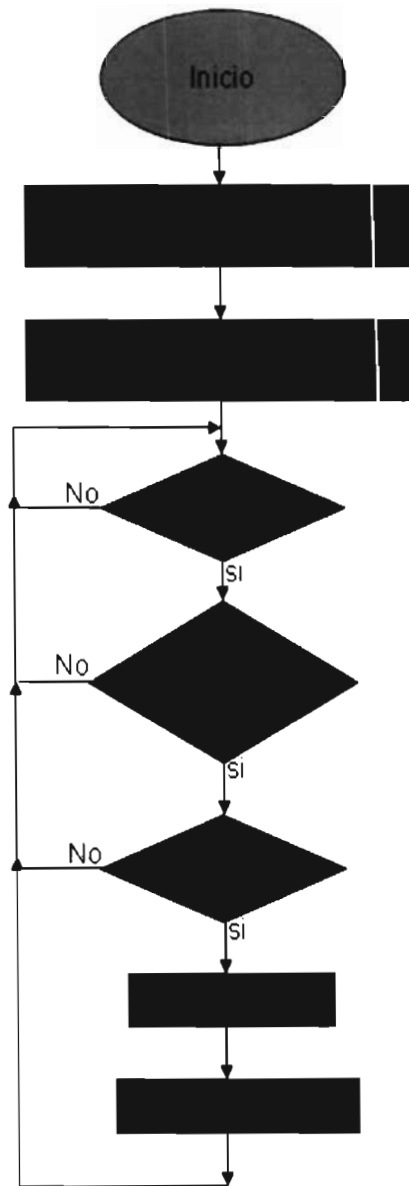


Figura 4.1. Diagrama de flujo del programa principal del microcontrolador PIC18F4550.



5. Datos en el Buffer: Una vez que el dispositivo fue enumerado entonces esperamos a que se envíen datos de la PC al PIC18F4550, que en este caso, esperamos que el programa realizado en LabVIEW envíe un comando para que el PIC realice alguna operación. Si no hay datos en el buffer entonces volvemos a verificar que el PIC continúe enumerado por la PC es decir que no haya problemas en la comunicación y esperamos que haya datos en el buffer para continuar con el programa.
6. Comando Válido: Si se recibe un comando que sea válido entonces se realiza la acción indicada por este de lo contrario se vuelve a la etapa de enumeración.
7. Transmite información: Finalmente si el dispositivo fue enumerado y se recibió un comando válido desde la PC y se ha realizado la acción definida por el comando entonces finalmente se transmiten datos desde el PIC18F4550 a la PC, este dato puede ser uno de los valores físicos a monitorear o una señal de validación de que se activaron en forma correcta las válvulas a la hora de comenzar o detener alguno de los procesos involucrados en el sistema de crecimiento.

La ejecución de las funciones está previamente definida en una lista de comandos los cuales se realizan dentro de una estructura de selección como lo es un "switch" el cual se ejecuta en la parte de validación de comando. En las tablas mostradas en las figuras siguientes se muestran todos los comandos empleados así como las acciones a realizar por el PIC18F4550 de acuerdo al comando indicado además de la función de cada uno de los pines del microcontrolador utilizado.

Lista de Comandos Utilizados		
Comando	Función	PIN
		18F4550
0x00	Lee temperatura externa.	A0
0x01	Lee humedad relativa.	A1
0x02	Lee humedad interna.	A2
0x03	Lee flujo de Hidrógeno.	A3
0x04	Lee vacío generado con la bomba Jet.	A5
0x05	Lee vacío generado con las bombas de absorción.	E0
0x06	ON válvula 1.	D7
0x07	ON válvula 2.	D6
0x08	ON válvula 3.	D5
0x09	ON válvula 4.	D4
0x10	ON válvula 5.	D3
0x11	ON válvula 8.	D2
0x12	ON purga	C1
0x13	ON horno.	A4

Figura 4.2. Primera parte de la lista de comandos usados en el programa del microcontrolador.

0x14	ON UP1.	E1 Control E2 Señal
0x15	ON UP2.	E1 Control E2 Señal
0x16	OFF válvula 1.	D7
0x17	OFF válvula 2.	D6
0x18	OFF válvula 3.	D5
0x19	OFF válvula 4.	D4
0x20	OFF válvula 5.	D3
0x21	OFF válvula 8.	D2
0x22	OFF purga.	C1
0x23	OFF horno.	A4
0x24	OFF UP1.	E1 Control E2 Señal
0x25	OFF UP2.	E1 Control E2 Señal
0x26	Activa modo PC	C0
0x27	Activa modo manual.	C0
0x28	Envía valor de flujo deseado.	B7 - B0

Figura 4.3. Segunda parte de la lista de comandos usados en el programa del microcontrolador.

## 4.2 LabVIEW

Cuando se habla de instrumentos de medida, es normal pensar en una carcasa rígida, en la que destaca su panel frontal lleno de botones, leds y demás tipos de controles y visualizadores. En la cara oculta del panel están los contactos de esos controles que los unen físicamente con la circuitería interna. Esta circuitería interna se compone de circuitos integrados y otros elementos que procesan las señales de entrada en función del estado de los controles, devolviendo el resultado a los correspondientes visualizadores del panel frontal.

Un instrumento virtual es un módulo software que simula el panel frontal del instrumento que antes se mencionó y apoyándose en elementos de hardware accesibles por el ordenador (tarjetas de adquisición, tarjetas DSP, instrumentos accesibles vía GPIB, RS-232, USB), realiza una serie de medidas como si se tratase de un instrumento físico.

De este modo cuando se ejecuta un programa que funciona como un instrumento virtual o VI (Virtual Instrument) el usuario ve en la pantalla de su ordenador un panel cuya función es idéntica a la de un instrumento físico, facilitando la visualización y el control del aparato.

A partir de estos datos reflejados en el panel frontal, el VI debe actuar recogiendo o generando señales, como lo haría su homólogo físico.

El control de instrumentación por ordenador no resulta nuevo; incluso el uso de la PC en sistemas de medida se usaba en los setenta mediante la interfase de bus IEEE 488 o GPIB (General Purpose Interface Bus). Pero ha sido en los noventa cuando los procesadores de 16 y 32 bits se han incorporado a equipos accesibles, consiguiendo altas velocidades y grandes capacidades de memoria.

Esta popularización de ordenadores de altas prestaciones ha traído consigo un fuerte desarrollo de potentes paquetes software que simplifican la creación de aplicaciones.

Hasta hace poco la tarea de construcción de un VI se llevaba a cabo con paquetes de software que ofrecían una serie de facilidades, como funciones de alto nivel y la incorporación de elementos gráficos, que simplificaban la tarea de programación y de elaboración del panel frontal, sin embargo, el cuerpo del programa seguía basado en texto, lo que suponía mucho tiempo invertido en detalles de programación que nada tienen que ver con la finalidad de un VI. Con la llegada del software de programación gráfica LabVIEW de National Instruments, Visual Designer de Burr Brown o VEE de Agilent Technology, el proceso de creación de un VI se ha simplificado notablemente, minimizándose el tiempo de desarrollo de las aplicaciones.

Cuando se crea un VI en LabVIEW trabajamos con dos ventanas: Una en la que se implementará el panel frontal (figura 4.4) y otra que soportara el nivel de programación (figura 4.5). Para la creación del panel frontal se dispone de una librería de controladores e indicadores de diferente tipo y la posibilidad de crear más, diseñados por el propio usuario.

Cuando un control es pegado desde la librería del panel frontal se acaba de crear una variable cuyos valores vendrán determinados por lo que el usuario ajuste desde el panel; inmediatamente aparece un terminal en la ventana de programación representándolo.

El nivel de programación del VI consistirá en conectar estos terminales a bloques funcionales hasta obtener un resultado que deseemos visualizar, por ejemplo un led de alarma. Los bloques funcionales son iconos con entradas y salidas que se conectan entre sí mediante cables ficticios por donde fluyen los datos, constituyendo el nivel de programación del VI.

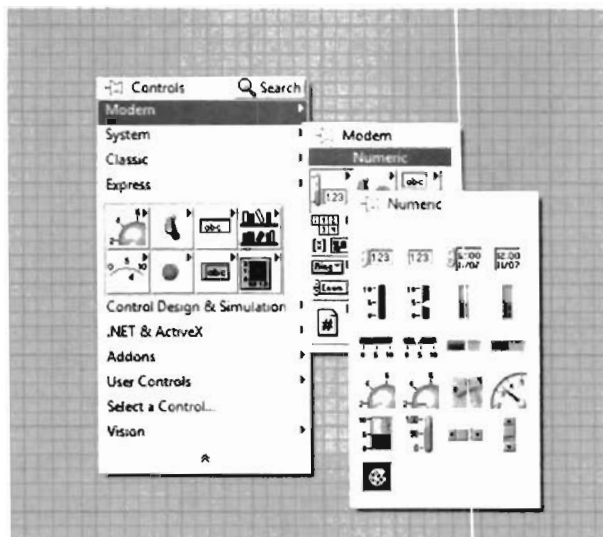


Figura 4.4 Panel frontal

Se puede comparar la ventana de programación con una placa de circuito impreso, donde las terminales del panel frontal se cablean a bloques funcionales (circuitos impresos) que se interconectan para generar los datos que se desean visualizar. A su vez, estos circuitos integrados contienen bloques circuitales conectados entre sí, al igual que un icono está formado por la interconexión de otros iconos. La programación gráfica permite diseñar un VI de manera intuitiva, vertiendo las ideas directamente a un diagrama de bloques, como se haría sobre una pizarra.

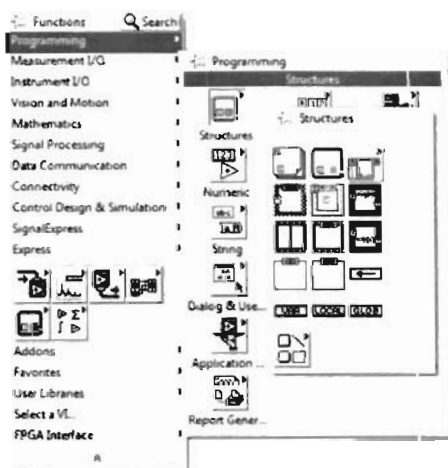


Figura 4.5 Diagrama de bloques

### 4.3 Programa principal de LabVIEW

En esta sección se dará una breve explicación del software desarrollado en la computadora, el programa se realizó en lenguaje de alto nivel, solo que en este caso se utiliza un lenguaje de programación gráfica como lo es LabVIEW. Se decidió utilizar este programa debido a que ofrece un ambiente gráfico para el usuario muy agradable y versátil además de permitir la visualización de los datos de una forma muy eficaz.

En la figura 4.6 se muestra el diagrama de flujo del programa principal de la PC, a continuación se explica el desarrollo del programa:

1. Inicio: Inicializa el panel frontal de control del programa.
2. Clave: El acceso al programa está protegido por medio de una clave de usuario, esto con el fin de que personas ajenas al laboratorio llegaran a hacer uso del sistema.
3. Enumerado: Se espera a que el PIC18F4550 se conecte a la PC para que sea enumerado y se establezca la comunicación entre ambos.
4. Modo de trabajo: En esta parte el software espera a que el usuario elija la forma en la que desea que opere el sistema, es decir que elija cualquiera de los cuatro modos de trabajo (manual, válvulas, bloques, y automático).
5. Modo Manual: Si el usuario elige este modo de trabajo el sistema opera a través de un conjunto de switches, este subprograma se explica en la siguiente sección.
6. Modo válvulas: Consiste en manipular los procesos involucrados en el sistema en forma virtual desde la PC, se verá con mayor detalle el funcionamiento de este en la siguiente sección.
7. Modo automático: El programa toma por completo el control del sistema bajo los parámetros indicados por el usuario.
8. Modo bloques: Este es muy similar al automático solo que aquí el crecimiento se realiza por etapas.
9. Ejecuta subprograma: Se ejecuta el modo de trabajo elegido por el usuario.
10. Fin modo: Se detiene el modo de trabajo y se retorna al programa principal.
11. Stop: Se espera a que se detenga el programa o se elija de nueva cuenta otra forma de trabajo.
12. Fin: Termina de ejecutarse el programa.

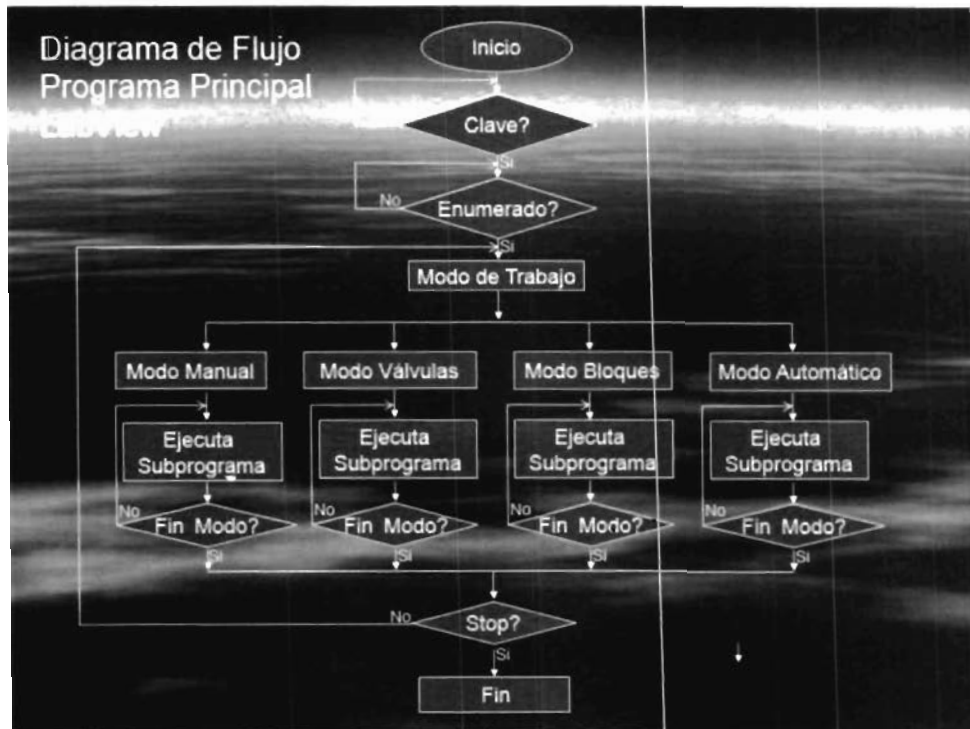


Figura 4.6. Diagrama de flujo del programa principal de la PC implementado en LabVIEW

En la figura 4.7 se muestra el panel frontal del programa principal de LabVIEW.

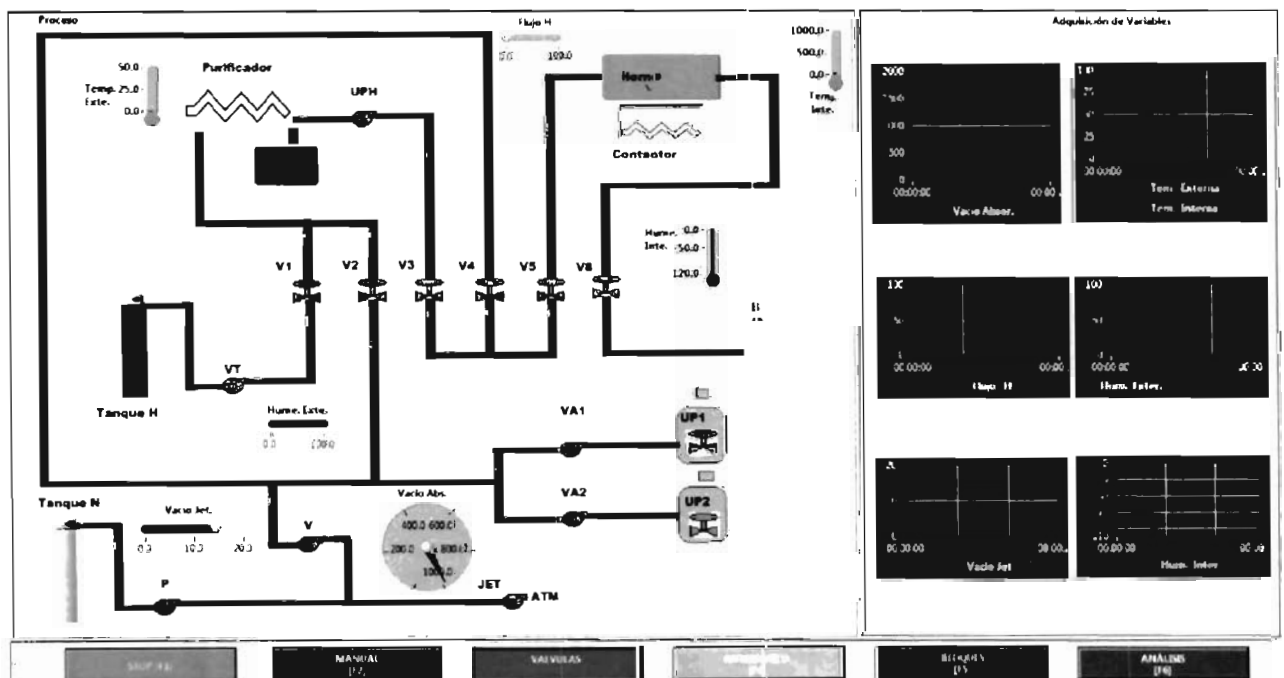


Figura 4.7. Panel de control principal.

## 4.4 Modos de operación del software de la PC:

En esta sección se dará una breve explicación de los programas implementados para las cuatro formas de trabajo del sistema. Se mostrará de una forma general los diagramas de bloques desarrollados en estas aplicaciones para comprender mejor como es que funciona la automatización de los procesos.

### 4.4.1 Modo manual

La forma de operar del sistema en este modo de trabajo consiste en que el usuario tendrá el control del sistema utilizando un conjunto de switches, por lo que el operador será quien tome las decisiones en el proceso. En este modo la PC solo realizará el control de flujo de Hidrógeno, es decir, se enviará el valor de voltaje a la válvula para que esta abra de acuerdo al porcentaje que se desea establecer en el flujo de hidrógeno en el sistema y llevará a cabo la adquisición y despliegue de los valores medidos por los sensores. En la figura 4.8 se muestra el diagrama de flujo de este subprograma el cual se explica a continuación.

1. Inicio: Configura panel frontal del modo manual.
2. Mensajes al usuario: Se le indica al usuario que lo primero que haga al iniciar este programa es enviar el comando de activación de la fuente que alimenta los switches con los que se trabajan en esta forma de funcionamiento.
3. Adquisición de datos: En esta parte se realiza la adquisición de todas las señales a monitorear a través de los sensores.
4. Activar switches: En esta parte se espera a que se oprima el botón con el cual se envía el comando para la activación de los switches.
5. Purga On: Se activa la válvula de purga cuando se hace vacío al sistema y cuando este se comienza a llenar de hidrógeno una vez que se alcanza la presión de 15 PSI y se abre la válvula 8 para bajar la presión dentro del sistema, se desactiva la purga por medio del botón de purga off.
6. Purga Off: Desactiva purga.
7. Porcentaje de flujo: Envía el valor deseado de apertura de la válvula para el flujo de hidrógeno, es decir, envía una señal digital de 0 a 255 cuyo valor se despliega en el puerto B del PIC18F4550 el cual está conectado a un convertidor digital análogo, la señal de salida de este se conecta a la válvula de apertura de flujo.
8. Stop: Se detiene este subprograma y se regresa al programa principal.

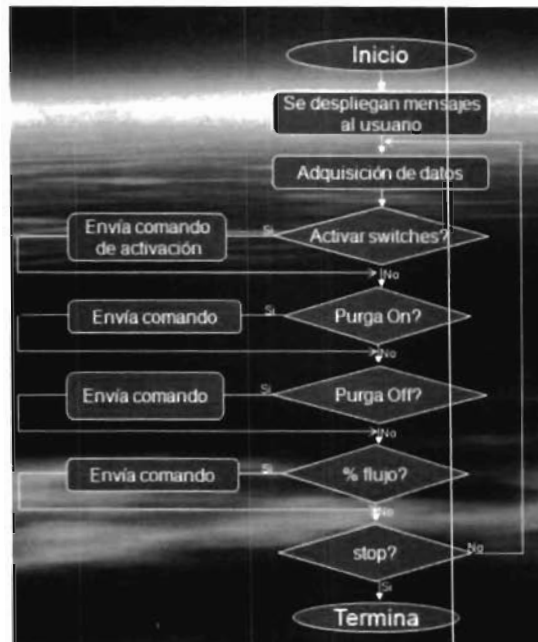


Figura 4.8. Diagrama de flujo del subprograma: modo manual.

En la figura 4.9 se muestra el panel frontal de este programa.

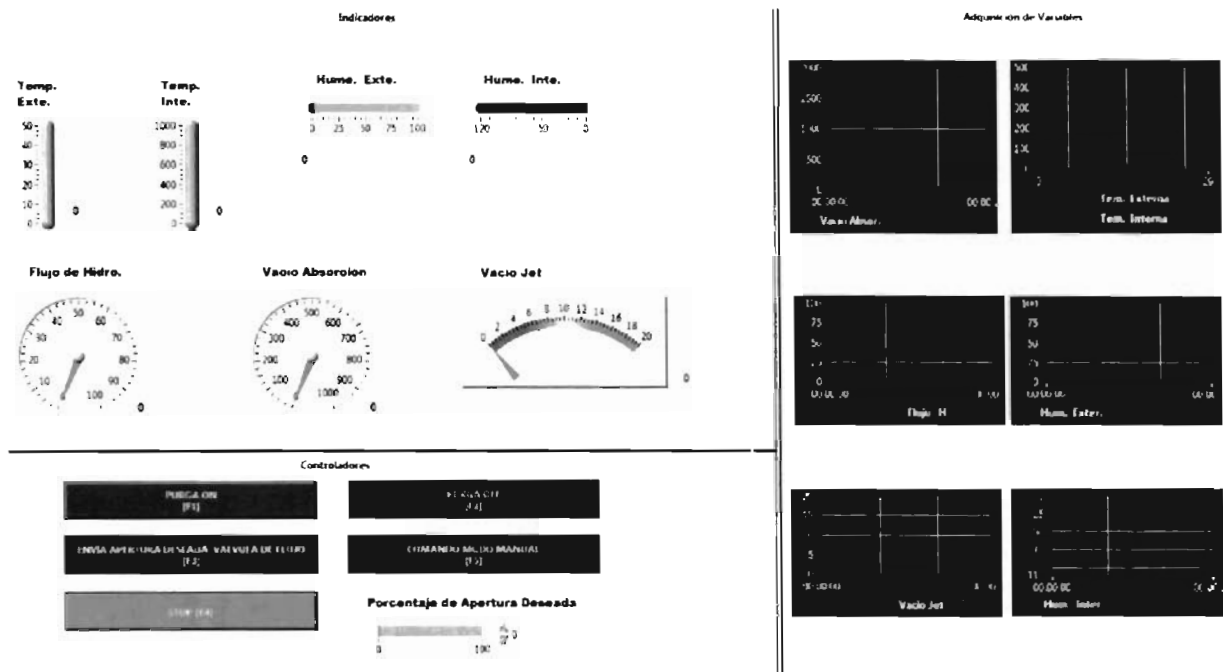


Figura 4.9. Panel frontal del modo manual.



#### 4.4.2 Modo válvulas

Consiste en manipular el conjunto de válvulas con las que se realizan los procesos de vacío y flujo de hidrógeno desde la PC pero sin dejar que la computadora tome el control por completo del proceso, es decir, se implementan switches virtuales que son operados desde la computadora. Además como en el modo de funcionamiento anterior en este también se realiza la adquisición de las variables físicas medidas por los sensores y el control de flujo de Hidrógeno. En la figura 4.10 se muestra el diagrama de flujo de este subprograma el cual se explica brevemente a continuación y en la figura 4.11 se ilustra el panel frontal de este programa.

1. Inicio: Configura panel frontal del modo válvulas.
2. Desactivar Switches físicos: se envía el comando para indicarle al microcontrolador que envíe la señal de desactivación de los switches físicos, esto con el objetivo de evitar posibles conflictos a la hora de utilizar el sistema desde la PC, es una forma de seguridad, se asegura que no se activen las válvulas neumáticas con los switches físicos cuando estas se activan por medio del microcontrolador PIC18F4550 por medio de la etapa de potencia.
3. Adquisición de datos: En esta parte se realiza la adquisición de todas las señales a monitorear a través de los sensores.
4. Activar válvulas: Aquí el programa espera a que el usuario active alguna de las válvulas o varias (válvula uno, dos, tres, cuatro, cinco, ocho, UP1 y UP2) para comenzar cualquiera de los procesos involucrados en el crecimiento, dependiendo de la etapa de crecimiento serán las válvulas que se utilicen, vacío Jet (válvulas usadas: cuatro y cinco), vacío con las bombas de Adsorción (válvulas usadas: cuatro, cinco, UP1 o UP2), flujo de hidrógeno (válvulas usadas: uno, tres, cinco y ocho) y en la parte de crecimiento se utilizan las mismas que en el flujo de hidrógeno.
5. Purga On: Se activa la válvula de purga cuando se hace vacío al sistema y cuando este se comienza a llenar de hidrógeno una vez que se alcanza la presión de 15 PSI y se abre la válvula 8 para bajar la presión dentro del sistema se desactiva la purga por medio del botón de purga off.
6. Purga Off: Desactiva purga.
7. Porcentaje de flujo: Envía el valor deseado de apertura de la válvula para el flujo de hidrógeno, es decir, envía una señal digital de 0 a 255 cuyo valor se despliega en el puerto B del PIC18F4550 el cual está conectado a un convertidor digital análogo, la señal de salida de este se conecta a la válvula de apertura de flujo.
8. Contacor: Se manda comando para encender o apagar el horno.
9. Stop: Se detiene este subprograma y se regresa al programa principal.

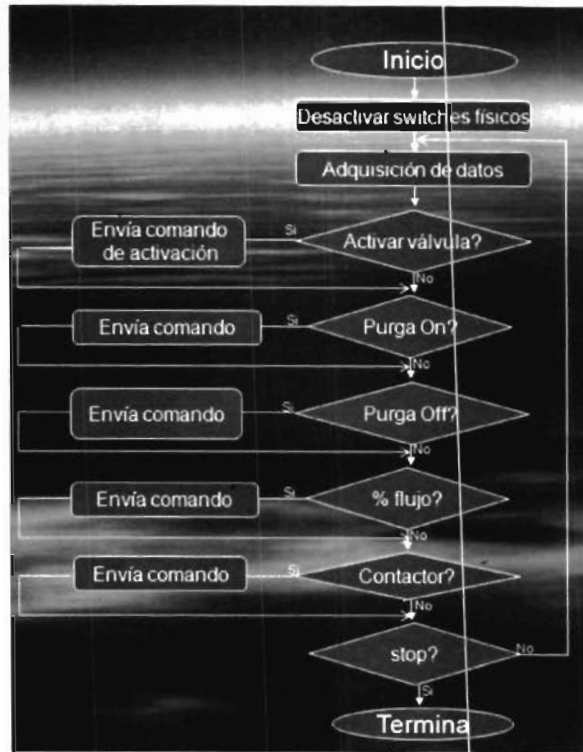


Figura 4.10. Diagrama de flujo del subprograma: modo válvulas.

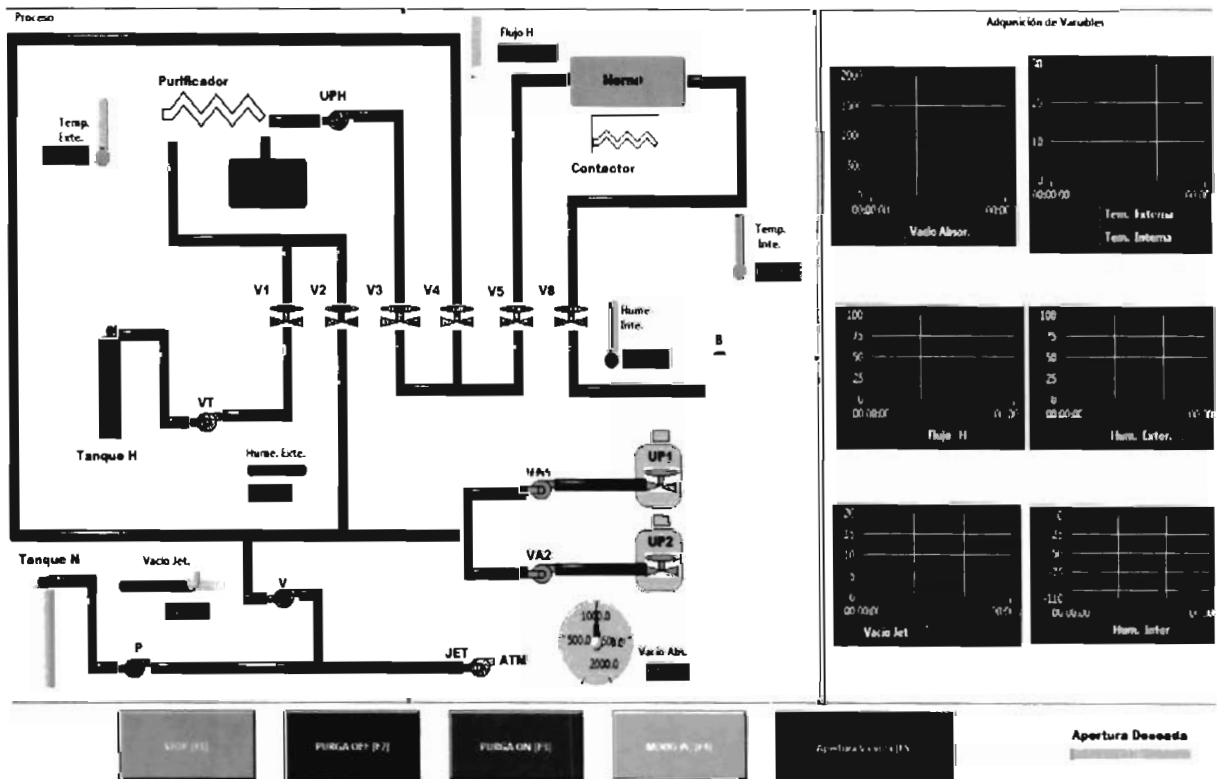


Figura 4.11. Panel frontal del modo válvulas.

#### 4.4.3 Modo bloques

Esta forma de trabajar el sistema es muy similar a la automática que será explicada en la siguiente sección. La diferencia entre esta y la automática es que el usuario podrá realizar el crecimiento por etapas, es decir, puede realizar cada uno de los procesos en forma independiente, esto es, no están enlazados todos los procesos, por ejemplo, abra ocasiones en las que no se necesite llevar a cabo todo el proceso, como lo es aplicar primero vacío tanto con la bomba Jet como con la Bomba de Adsorción después hacer fluir hidrógeno para purificar el sistema y que la humedad en el sistema baje a un valor óptimo de acuerdo a los cálculos del operador para finalmente hacer el crecimiento aplicando el diagrama de fases establecido para el crecimiento que se desea realizar. Esto es lo que se debe realizar cuando se hace un crecimiento, pero hay ocasiones en las que no se realizará un crecimiento como tal si no que solo será necesario hacer por ejemplo fluir hidrógeno durante todo un día, probablemente porque así lo requiera el experimento, o solo hacer vacío al sistema pero no necesariamente con los dos tipos de bomba si no solo con una bomba. Es por eso que se implementa esta forma de trabajo, porque aquí la computadora es quien toma el control del sistema pero se pueden realizar los procesos de uno por uno de acuerdo a las características del crecimiento que se quiere realizar y no necesariamente hacer todo el proceso si no es necesario. En la figura 4.12 se muestra en una forma muy breve el diagrama de flujo de este programa y en la figura 4.13 el panel frontal de este.

1. Inicio: Configura panel frontal del modo bloques.
2. Espera proceso: el programa espera a que el usuario configure los parámetros de crecimiento tales como las temperaturas de saturación, homogenización, crecimiento, humedad interna deseada, porcentaje de flujo de hidrógeno, valor de vacío con la bomba jet a alcanzar, etc. para que posteriormente se elija el proceso a realizar.
3. Desactivar Switches físicos: Se envía el comando para indicarle al microcontrolador que envíe la señal de desactivación de los switches físicos, esto con el objetivo de evitar posibles conflictos a la hora de utilizar el sistema desde la PC, es una forma de seguridad, que impide que se activen las válvulas neumáticas con los switches físicos cuando estas se activan por medio del microcontrolador PIC18F4550 a través de la etapa de potencia.
4. Configuración: Al oprimir este botón se despliega una ventana en la cual el usuario define todos los parámetros de crecimiento deseados.
5. Vacío Jet: Se da comienzo al vacío utilizando la bomba Jet.
6. Vacío Adsorción: Vacío con las bombas de adsorción.
7. Flujo de Hidrógeno: Se purifica el sistema y se espera llegar a la humedad deseada para comenzar el crecimiento.
8. Crecimiento: Se implementa el diagrama de fases a través del control de la temperatura del horno.
9. Stop: Se detiene este subprograma y se regresa al programa principal.

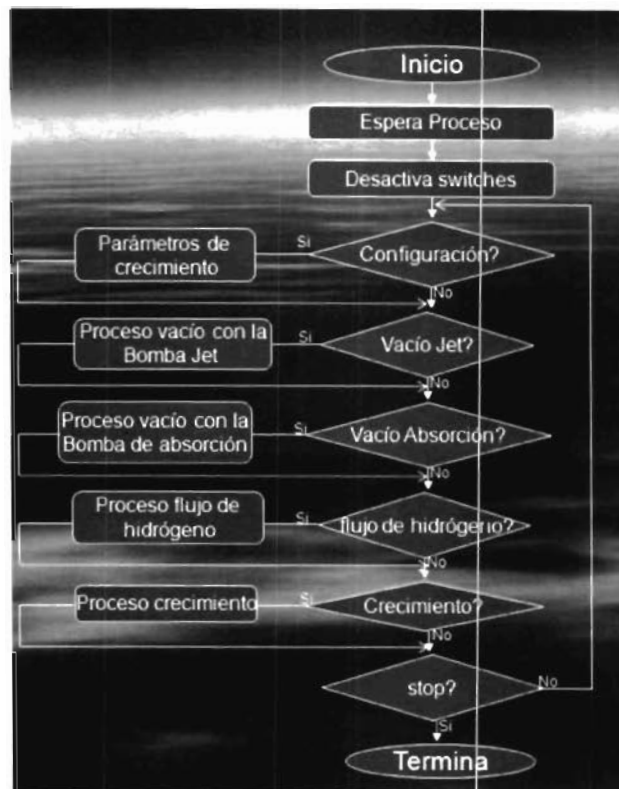


Figura 4.12. Diagrama de flujo del subprograma: modo bloques.

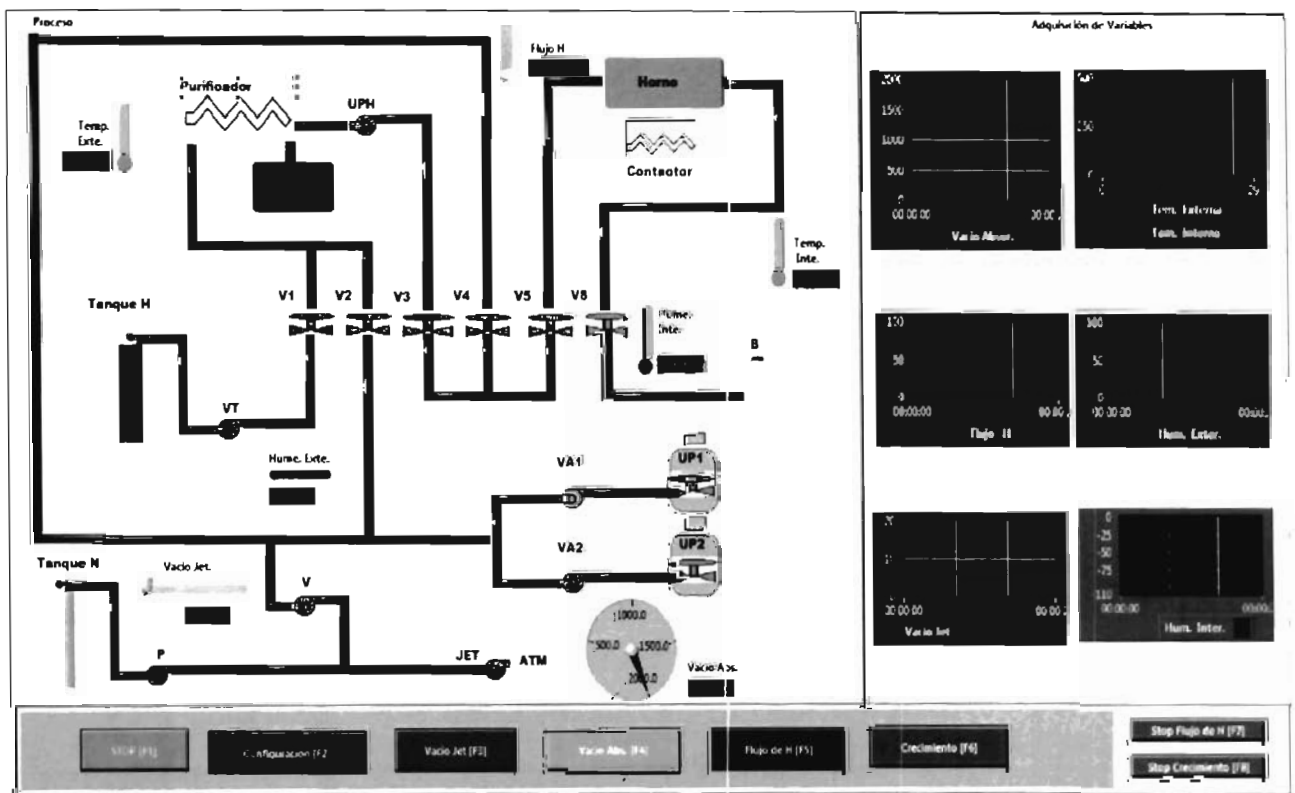


Figura 4.13. Panel frontal del modo bloques.

#### 4.4.4 Modo automático

En esta forma de operar, el programa si toma por completo el control del sistema para realizar el crecimiento de acuerdo a las especificaciones predeterminadas por el usuario. Una vez que han sido configurados los parámetros de crecimiento el software se encarga de realizar todo el proceso, tal como se mencionó en la sección anterior en este programa se enlazan todos los procesos para realizar el crecimiento que es la gran diferencia con el modo de trabajo anterior. En la figura 4.14 se muestra el diagrama de flujo y en la figura 4.15 el panel frontal de este subprograma.

1. Inicio: Configura panel frontal del modo automático.
2. Desactiva Switches: Se envía el comando para indicarle al microcontrolador que envíe la señal de desactivación de los switches físicos, esto con el objetivo de evitar posibles conflictos a la hora de utilizar el sistema desde la PC, es una forma de seguridad que impide que no se activen las válvulas neumáticas con los switches físicos cuando estas se activan por medio del microcontrolador PIC18F4550 a través de la etapa de potencia.
3. Configuración: Al oprimir este botón se despliega una venta en la cual el usuario define todos los parámetros de crecimiento deseados.
4. Vacío Jet: Se da comienzo al vacío utilizando la bomba Jet, si este proceso se realiza de forma correcta, es decir, se alcanza el valor de presión deseado entonces pasa a la siguiente etapa de lo contrario despliega un mensaje de error y termina el proceso.
5. Vacío Adsorción: Vacío con las bombas de adsorción, al igual que la anterior etapa si esta se realiza de forma correcta procede a realizar el flujo de hidrógeno.
6. Flujo de Hidrógeno: Una vez que se ha realizado de forma correcta el vacío en el sistema se procede a purificar este y se espera alcanzar la humedad deseada para poder comenzar con la etapa de crecimiento.
7. Crecimiento: Se implementa el diagrama de fases a través del control de la temperatura del horno.
8. Termina experimento: Se espera a que la temperatura del sistema sea de 80 °C para poder comenzar a cerrar válvulas y poder detener el flujo de hidrógeno ya que ha esta temperatura ya es seguro detener el flujo.
9. Stop: Se detiene este subprograma y se regresa al programa principal.

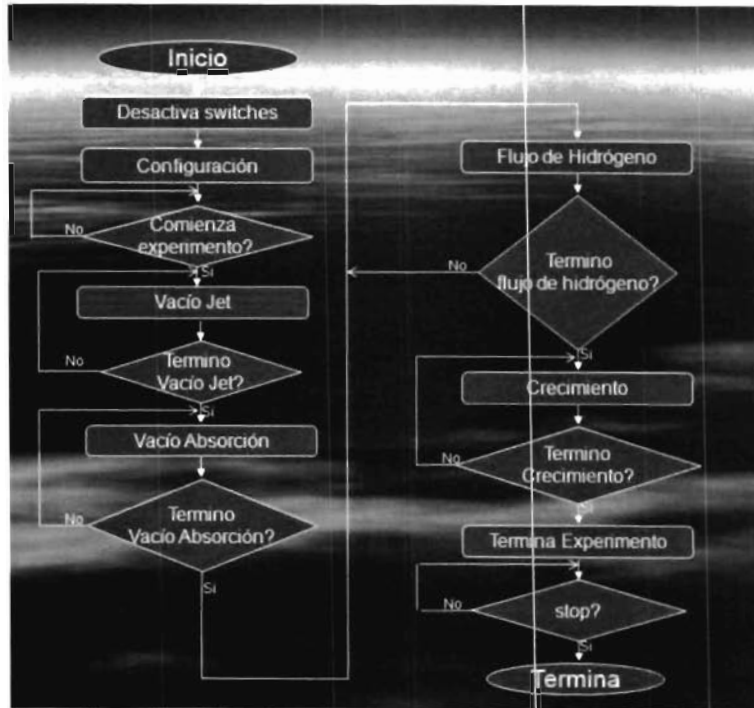


Figura 4.14. Diagrama de flujo del subprograma: modo automático.

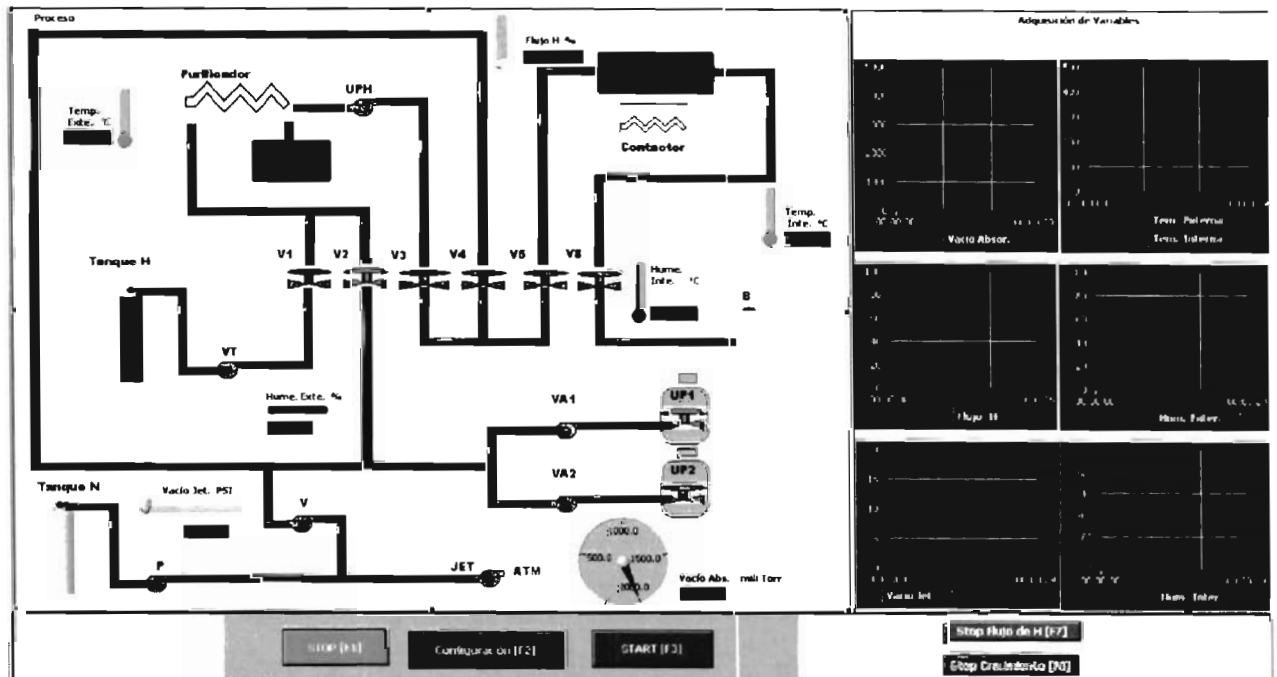


Figura 4.15. Panel frontal del modo automático.

## **CAPÍTULO V**

### **RESULTADOS**

#### **5.1 Pruebas Realizadas**

En este capítulo se muestran los resultados obtenidos durante la realización de las pruebas para la verificación del correcto funcionamiento del sistema, se depura el software de control implementado en LabView y se corrobora el correcto funcionamiento de las tarjetas de control basadas en el microcontrolador PIC18F4550.

En esta sección se muestran algunas de las gráficas y tablas que se obtuvieron de los datos almacenados que fueron adquiridos durante la realización de las pruebas así como también se presentan los paneles frontales de control de los cuatro modos de trabajo publicados en la Web, ésta es una herramienta que proporciona LabVIEW para el control y monitoreo de paneles frontales en forma remota. En párrafos siguientes se ofrece una explicación más detallada de esta aplicación del software de control.

Hay que recordar que uno de los objetivos de la automatización del sistema es el de recaudar y guardar información de los parámetros físicos presentes durante la realización de un crecimiento, esto con la intención de poder analizar con detalle su comportamiento durante todo el experimento y cómo es que afectan a la realización del crecimiento, para de esta forma, tratar de realizar los experimentos bajo las mejores condiciones posibles.

Lo primero que se verificó de la automatización del sistema fue el correcto funcionamiento de las tarjetas de control; básicamente las pruebas realizadas fueron encaminadas a la apropiada adquisición de las variables monitoreadas por los sensores, el adecuado accionamiento electrónico de las válvulas neumáticas así como el de constatar que las tarjetas de control soportaran cargas de trabajo de 24 horas o más, para esto se realizaron varias pruebas en las cuales se dejaba el sistema funcionando a máxima carga durante dos o tres días enteros. En cuanto al accionamiento de las válvulas éstas se activaban desde la PC o en su caso en forma manual a través de los switches, ello con el fin de constatar que la lógica de accionamiento era la apropiada.

En lo que atañe a la apropiada adquisición de las variables físicas se tuvieron varios problemas; la temperatura interna que se adquiere por medio del Eurotherm 818 vía puerto serie (RS232), en una de las pruebas realizadas, envió una mala lectura, la cual provocó que entraran las condiciones de seguridad del software de control (dicha condición de seguridad consiste en que si alguna de las

temperaturas del crecimiento, tales como disolución, saturación y crecimiento suben por encima de un rango predefinido entonces el sistema se apaga, lo cual indicaría que el control de temperatura está fallando y no es factible que se realice el experimento por condiciones de seguridad). El error consistió en que en menos de un segundo se tuvieron lecturas con diferencia entre cada una de 100 °C, dicho evento indica una gran falla, ya que por más corriente que se inyecte a la resistencia del horno es imposible que la temperatura suba 100°C en un segundo. Este error se corrigió colocando una condición en la cual el programa no continúe su ejecución si entre lectura y lectura hay una diferencia de 10°C (para evitar un ciclo infinito si se presentara una lectura errónea constante se metió la condición de que el ciclo se ejecutara un número finito de veces por lo cual si el error continua el programa se detiene).

En lo que respecta a las otras seis variables físicas que son adquiridas por medio del convertidor análogo digital del microcontrolador y que son enviadas a la PC a través del puerto USB también se presentó un problema en su obtención y muestreo ya que en varios de los ensayos realizados presentaban el problema de que se corrían en una posición, este problema también se presentaba cuando se hacia el cambio de un modo de trabajo a otro modo de trabajo. Al igual que el error anterior esto provocaba que entraran las condiciones de seguridad en momentos en los cuales no se requerían. Este problema se resolvió colocando un identificador para cada una de las variables leídas, de tal forma que en LabVIEW se visualizaran los datos solo cuando estos se adquirían en forma correcta, es decir, cuando la señal solicitada al PIC18F4550 por el programa realizado en LabVIEW tuviera el identificador apropiado, era cuando se permitía su visualización y adquisición.

Además de estos problemas de software que se presentaron en la adquisición de las variables se tuvieron otras dificultades para poner a punto el programa de control, estos dilemas se debieron más a errores de programación los cuales fueron corregidos en su momento.

En lo que respecta a las condiciones de seguridad estas se realizan con el objetivo de asegurarnos de que el proceso se haga bajo un ambiente de confianza y tener la certeza de que si se presentara alguna anomalía en el funcionamiento del sistema este se detendrá proporcionando la protección del usuario y el sistema.

Las condiciones de seguridad que se implementaron fueron las siguientes:

- Cuando la presión dentro del sistema llegue a 15 PSI se abre la válvula ocho, esto se hace solo cuando se llena el sistema de hidrógeno.
- Si la presión interna es igual a 16 PSI cuando se está fluyendo hidrógeno se detiene el proceso esta misma condición se implementa en la parte de crecimiento solo que aquí lo primero que se hace es mandar a cero el set point para que comience a enfriarse el horno.
- Si se hace cero el flujo de hidrógeno también se detiene el experimento.



- Otra de las condiciones de seguridad implementadas fue la de apagar el sistema si la temperatura de control (disolución, saturación, crecimiento) sube por encima de un límite establecido.
- La temperatura adecuada para apagar el sistema una vez que se ha realizado el experimento en forma correcta es de 60°C a esta temperatura ya se puede abrir el sistema para retirar las muestras e introducir nuevo material según sea el caso.

Hay que mencionar que durante la realización de las pruebas (estas se comenzaron a realizar desde mediados de mayo) hubo varias ocasiones en las que entraron las condiciones de seguridad con lo cual se pudo constatar de una manera no muy grata que estas funcionan.

A continuación se muestran solo algunas de las gráficas obtenidas durante la realización de las pruebas.

La figura 5.1 muestra datos recopilados durante una prueba realizada en modo manual. Las gráficas que proporcionan mayor información del experimento son la de temperatura interna, y humedad interna la cual es expresada como la temperatura de rocío, ya que en estas se puede ver cómo es que se afecta a la humedad en el sistema, cuando es introducido el material, entra aire al sistema, cuando comienza el flujo de hidrógeno la humedad aumenta drásticamente y poco a poco comienza a disminuir conforme se hace fluir hidrógeno en el sistema tal como se puede ver en la figura 5.1. Se puede observar que cuando la temperatura interna del sistema empieza a subir la humedad también aumenta esto se debe a que al aumentar la temperatura la velocidad de desorción de las moléculas de agua aumenta. El perfil de temperatura realizado en este experimento consiste en elevar la temperatura a 600 °C (disolución) se mantiene veinte minutos en este valor, se baja a 550°C (saturación) y se mantiene quince minutos después se lleva a la temperatura de crecimiento que en este caso es de 540°C y se deja diez minutos para que se estabilice, una vez transcurrido este tiempo se procede a implementar una rampa controlada de medio grado por minuto hasta llegar a 530°C en este punto se mantiene cinco minutos para finalmente poner el set point a cero y de esta forma termina el ensayo, esto se puede observar en la gráfica de temperatura de la figura 5.1.

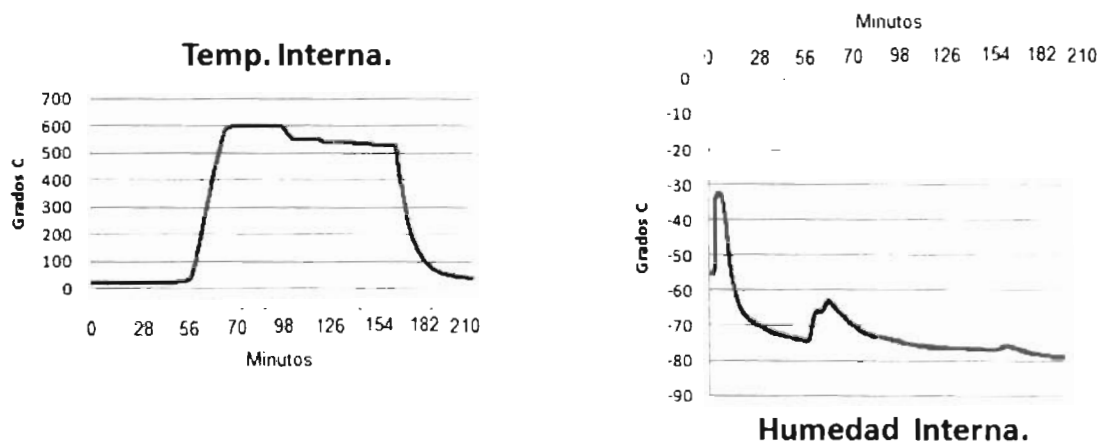


Figura 5.1 comportamiento de la humedad interna y la temperatura interna durante el procedimiento.

La figura 5.2 muestra datos recopilados durante una prueba realizada en modo válvulas. En este caso se muestran las gráficas de temperatura interna, flujo de hidrógeno y humedad interna. El perfil de temperatura realizado en este experimento es básicamente el mismo que el anterior, se eleva la temperatura a 550 °C (disolución) se mantiene veinte minutos, se baja a 500°C (saturación) se mantiene quince minutos para después bajar la temperatura del horno a la temperatura de crecimiento, 490°C, permanece diez minutos en este valor, para después implementar una rampa controlada de medio grado por minuto hasta llegar a 480°C, es en este momento donde se realizan los movimientos para iniciar el crecimiento epitaxial sobre el substrato, una vez que se tiene la temperatura de 530 °C se mantiene cinco minutos para finalmente bajar el set point a cero y terminar el experimento, esto se puede observar en la gráfica de temperatura de la figura 5.2. Se puede observar que la humedad interna no presenta el rápido aumento inicial como el que se ve en la Fig. 5.1 esto se debe a que el sistema no se abrió al iniciar el ciclo. La rápida disminución del flujo que se observa al inicio de la gráfica se debe a que en esos momentos el tubo de cuarzo se está llenando con hidrógeno a la mayor velocidad de flujo posible (determinado por las diferencias de presión externa e interna) durante el llenado del tubo el controlador de flujo solo funciona como medidor.

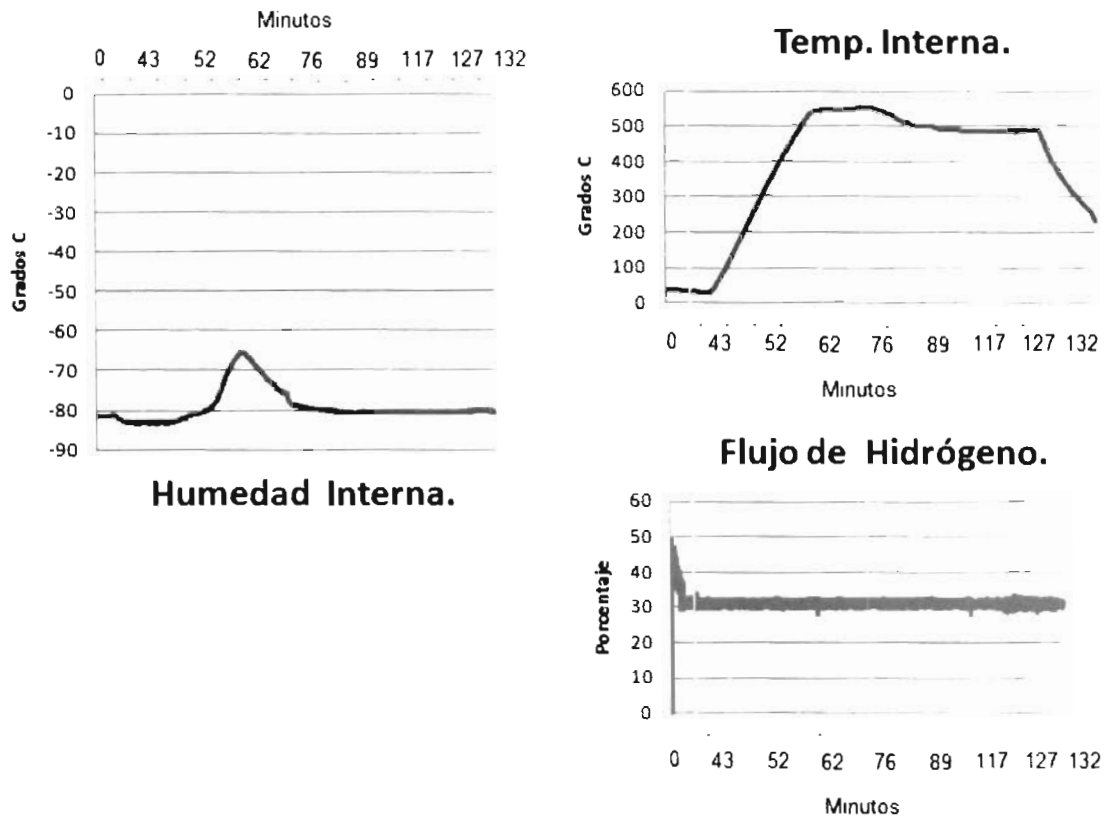


Figura 5.2 Comportamiento de la humedad interna, flujo de hidrógeno y temperatura interna durante el experimento.

La figura 5.3 muestra datos recopilados durante una prueba realizada en modo automático. En este caso se muestran las gráficas de temperatura interna y vacío con la bomba Jet. El perfil de temperatura realizado en este experimento consistió en elevar la temperatura a 550 °C (disolución) se mantiene veinte minutos, se baja a 500°C (saturación) se mantiene quince minutos para después bajar la temperatura del horno a la temperatura de crecimiento, 490°C, permanece diez minutos en este valor, para después implementar una rampa controlada de medio grado por minuto hasta llegar a 480°C, es en este momento donde se realizan los movimientos para efectuar el crecimiento epitaxial sobre el sustrato, una vez que se tiene la temperatura de 480 °C se mantiene cinco minutos para finalmente bajar el set point a cero y terminar el experimento, esto se puede observar en la gráfica de temperatura de la figura 5.3.

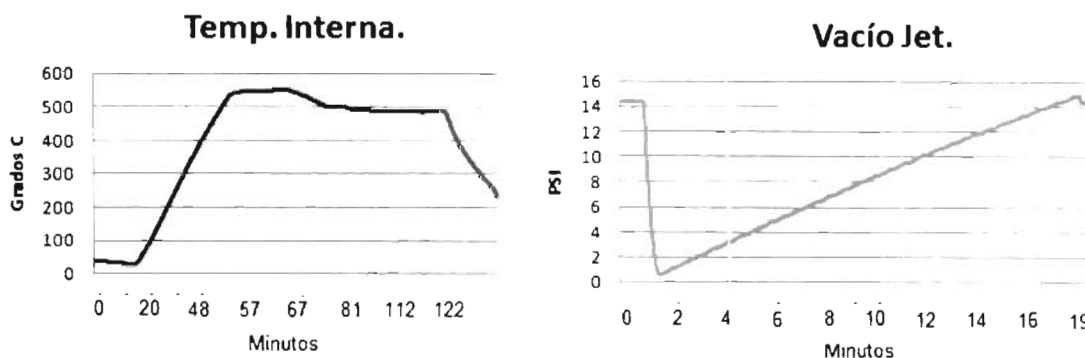


Figura 5.3 Comportamiento de la temperatura interna y vacío con la bomba jet durante el experimento.

La figura 5.4 muestra datos recopilados durante una prueba realizada en modo bloques. En este caso se muestran las gráficas de humedad interna y vacío con las bombas de adsorción. Estas gráficas ilustran los comportamientos de estas variables cuando se hace vacío al sistema, hay que señalar que al día siguiente de que se hizo vacío al sistema se realizó un crecimiento, las gráficas e imágenes del crecimiento se muestran más adelante.

En estas graficas se puede ver que al utilizar las dos bombas de adsorción se logran vacíos muy buenos llegando a valores cercanos a los 40 mili Torr, en la gráfica de la izquierda se puede ver cómo es que el valor del vacío comienza a disminuir hasta llegar a un punto en el cual las bombas de adsorción están saturadas, se puede ver como alrededor de 40 minutos se mantiene un vacío muy bueno después de este tiempo la presión comienza a elevarse hasta alcanzar el valor de inicio esto se debe a las pequeñas fugas presentes en el sistema la diferencia es que ya se ha eliminado una gran cantidad de moléculas de oxígeno del sistema. En la gráfica de la derecha se muestra el comportamiento de la humedad interna, de acuerdo al vacío generado al sistema, se puede ver cómo es que la humedad disminuye conforme el vacío mejora, hay un comportamiento muy interesante entre los 15 y 89 minutos de iniciado el experimento, este consiste en dos picos de valores muy marcados en la gráfica. Este comportamiento se debe al cambio de bomba con la que se realiza vacío, es necesario cambiar de bomba ya que como se mencionó en párrafos anteriores hay un momento en el que las bombas se saturan por lo que es conveniente cambiar de bomba una vez que ambas están saturadas termina el vacío con ellas.

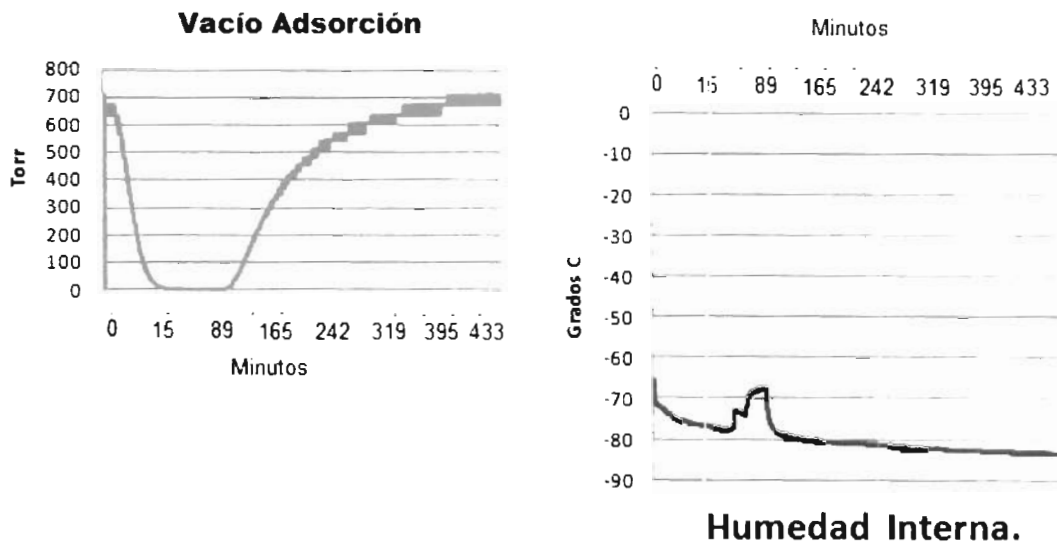
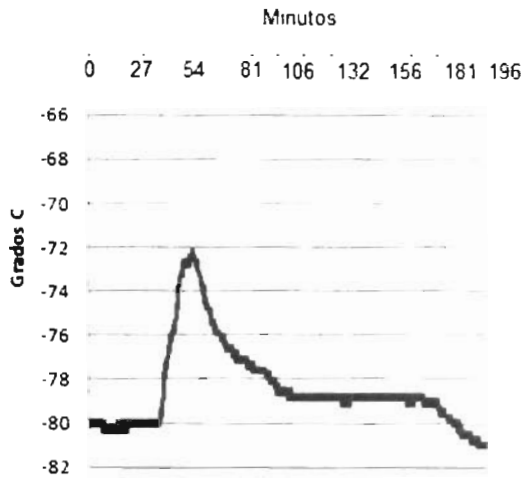


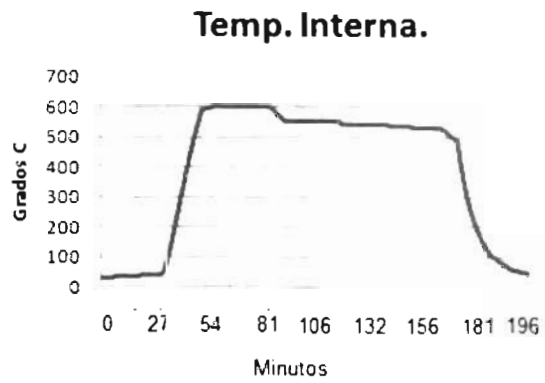
Figura 5.4 Comportamiento de la humedad interna y vacío con las bombas de adsorción durante el experimento.

Como se mencionó anteriormente las imágenes siguientes muestran el crecimiento que se realizó, este se hizo al día siguiente en el que se le hizo vacío al sistema y se dejó fluyendo hidrógeno en el sistema.

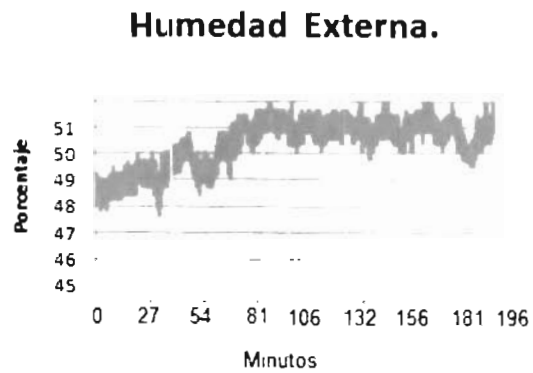
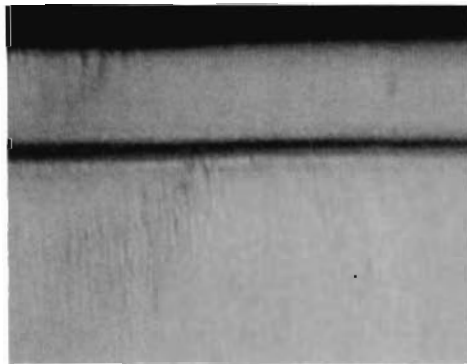
En la figura 5.5 se muestran las gráficas de humedad interna, temperatura interna y humedad externa así como la imagen del crecimiento logrado. El perfil de temperatura implementado fue el siguiente: temperatura de disolución de 600 °C se mantiene alrededor de veinte minutos en esta temperatura, se manda la temperatura de saturación de 550 °C se mantiene quince minutos, se baja la temperatura del horno a la temperatura de crecimiento que para este crecimiento es de 540 °C y se espera a que se estabilice durante diez minutos, para después implementar una rampa controlada de medio grado por minuto hasta llegar a 530 °C, que es en esta parte donde se realizaron los movimientos para hacer el crecimiento, finalmente se mantiene la temperatura en 530 grados durante cinco minutos y se manda el set point a cero grados para dar por terminado el crecimiento.



**Humedad Interna.**



**Temp. Interna.**



**Humedad Externa.**

Figura 5.5 Comportamiento de la humedad interna, temperatura interna y humedad externa durante el crecimiento.

En la figura 5.6 se muestran las gráficas de humedad interna, temperatura interna y temperatura externa así como la imagen de otro crecimiento realizado. En esta ocasión mostramos el comportamiento de las variables en el proceso que se llama recocido de la muestra crecida, lo que se pretende con el recocido es lograr una uniformidad en las capas epitaxiales crecidas de la muestra dicho recocido se realiza a una temperatura menor a la temperatura de disolución, en este caso, el recocido se realizó a 550 °C durante tres horas y media.

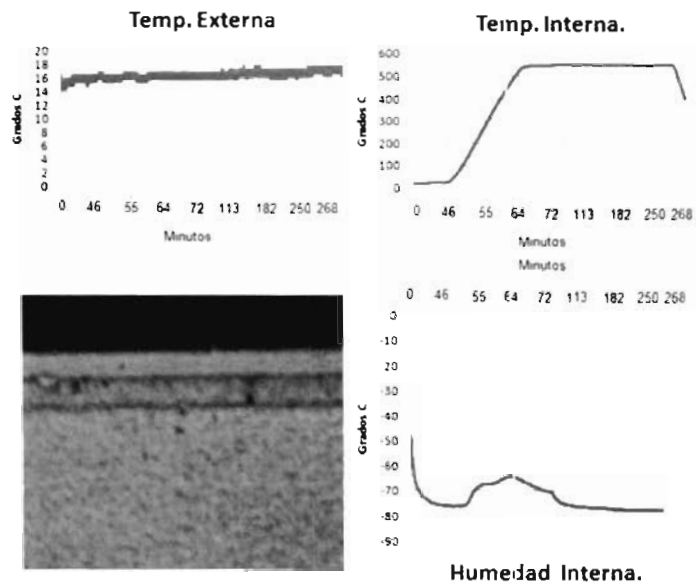


Figura 5.6 Comportamiento de la humedad interna, temperatura interna y temperatura externa durante el recocido de tres horas y media de la muestra.

Una de las opciones que ofrece la automatización del sistema es la de poder visualizar ó en su caso manipular el proceso en forma remota, es decir, gracias a las herramientas que proporciona LabVIEW es posible publicar los paneles frontales de los diferentes modos de trabajo en páginas Web para poder acceder a estos por medio de internet. Los paneles frontales publicados en las páginas Web se muestran en la figura 5.8 y la página Web del programa principal desde donde son llamadas las otras páginas a través de los links correspondientes se muestra en la figura 5.7. Hay que mencionar que solo es posible conectarse a estos paneles en forma remota dentro de la red de la UASLP por cuestiones de política de la misma.

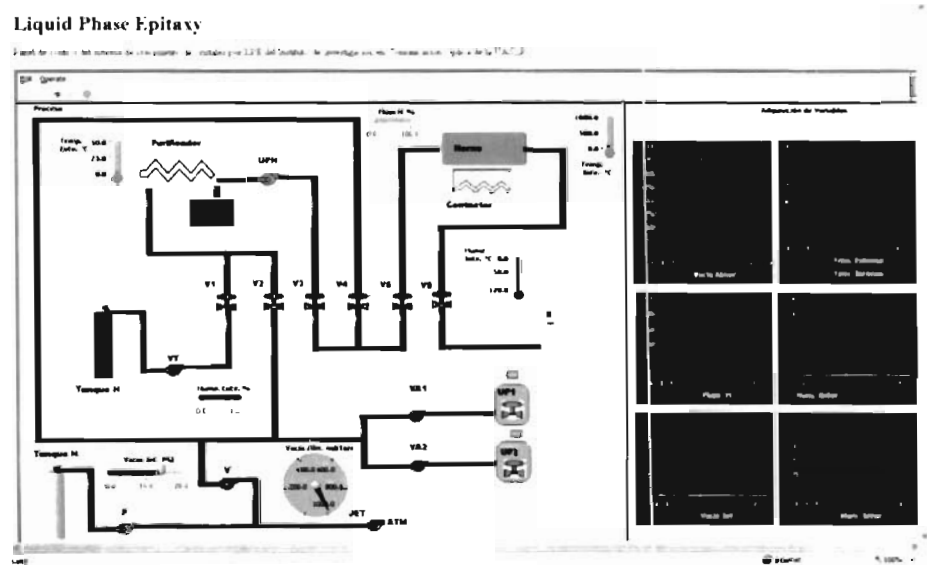


Figura 5.7 Panel principal publicado en la Web.

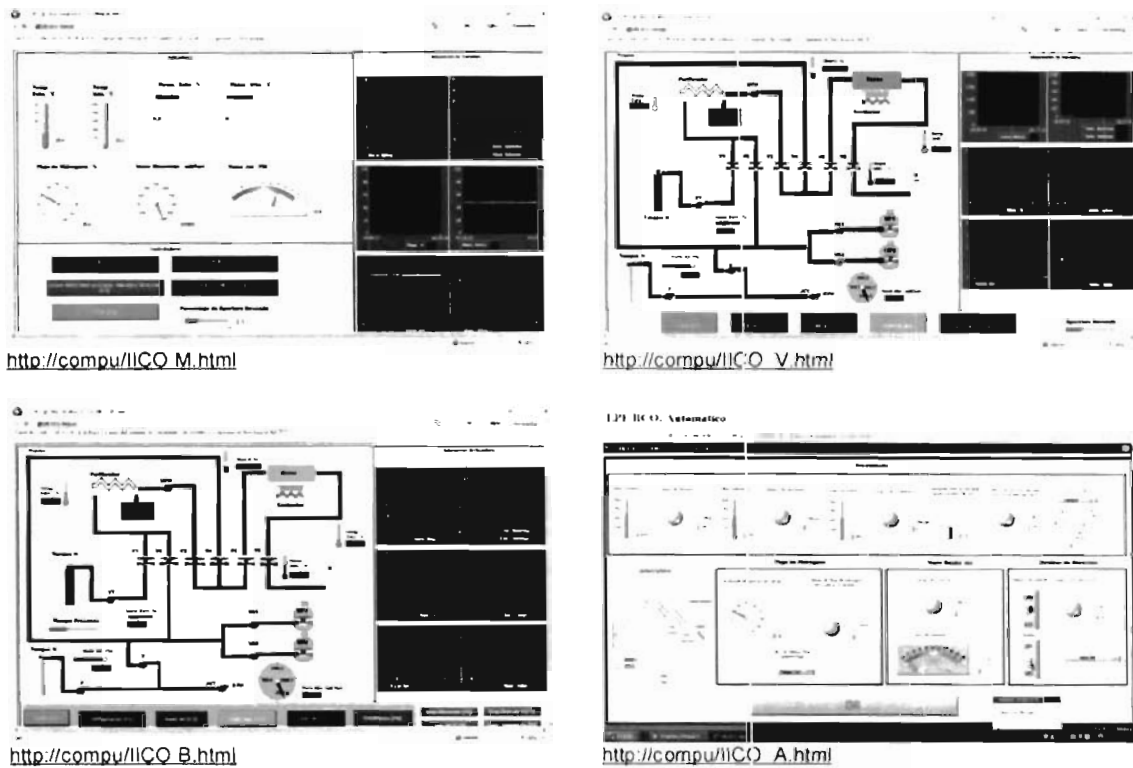
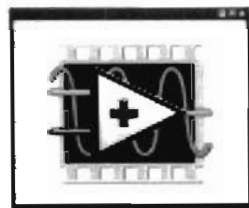


Figura 5.8 Paneles frontales de los modos de trabajo publicados en la Web.

Además de esta opción LabVIEW también proporciona la herramienta para crear la aplicación y el ejecutable del programa implementado en esta plataforma de programación. De esta manera no es necesario el tener instalado LabVIEW solo basta con crear el instalador y la aplicación y de esta forma es posible instalar nuestra aplicación en cualquier máquina y nuestro programa estará funcionando. El icono de nuestro programa se muestra en la figura 5.8.



Liquid Phase  
Epitaxy.exe

- bin
- license
- supportfiles
- nidist.id
- setup.exe
- setup.ini

Figura 5.8 De izquierda a derecha se muestra el icono creado para la aplicación del programa de control y el ejecutable para la instalación del programa en el sistema de la PC.



## **CAPÍTULO VI**

### **CONCLUSIONES**

#### **6.1 Conclusiones**

Este trabajo de tesis se implementó utilizando la plataforma de programación LabVIEW desarrollando en está el programa de control del sistema de crecimiento de cristales. De esta forma se hizo posible automatizar los procesos involucrados en el crecimiento de cristales por la técnica de epitaxia en fase líquida.

Para la automatización del sistema se fabrico una tarjeta de control basada en el microcontrolador PIC18F4550 de Microchip, dicha tarjeta incluye la etapa de potencia para la activación de las válvulas, encendido del horno así como la adquisición de las señales analógicas medidas por los sensores instalados en el sistema.

Finalmente se realizaron una gran cantidad de pruebas para verificar el correcto funcionamiento del sistema, las pruebas realizadas ayudaron a depurar el software de control para de esta manera obtener los resultados esperados, el sistema está automatizado y funciona de acuerdo a las necesidades del usuario el cual configura el sistema en función de las características del experimento que desea realizar.

El sistema está siendo usado para la realización de crecimientos y recocidos de materiales dentro del laboratorio de crecimiento de cristales del Instituto de Investigación en Comunicación Óptica.

## 6.2 Trabajo a futuro

En el actual trabajo de tesis no se han automatizado aun los movimientos que se realizan para la deposición del material sobresaturado en el sustrato para el crecimiento epitaxial de semiconductores, el automatizar estos movimientos seria una gran aportación ya que se eliminarían por completo las vibraciones presentes en el sistema durante la realización del experimento además de tener la posibilidad de hacer crecimientos más controlados.

Otra de las actividades a realizar es la referente a los puntos de líquidus, el poder controlar la medición de los puntos de líquidus es también de gran ayuda ya que este es un proceso muy lento y tedioso para el usuario. Se hace la medición de los puntos de líquidus cuando se requiere conocer la temperatura a la cual los cristales utilizados en el experimento se disuelven por completo en el solvente, esto garantiza mejores resultados ya que se conoce la temperatura exacta a la cual los cristales están totalmente disueltos y no hay partes solidas presentes en la solución.

Finalmente otra de las opciones a trabajar es la de automatizar el pesaje de los materiales utilizados en el crecimiento de cristales debido a que es aquí donde el investigador tiene mayores dificultades esto debido a las proporciones tan pequeñas en peso y medida de los materiales utilizados.

Programa de Control del Microcontrolador PIC18F4550.

LineMane18F.c

A continuación se describe el programa de control para el Sistema de crecimiento de cristales y utilizando la comunicación I2C en modo pulso desensabotado el cual trabaja a una velocidad de transferencia de datos de 40 Kbps.

Espera recibir 3 bytes de la PI. El primer byte es el comando el cual le indica al PIC18F4550 que acción desarrollara. Ya sea adquisición de señales analógicas o la activación de las válvulas de control del sistema. El segundo byte contiene datos que se utilizarán en el programa de control y el tercer byte no se utiliza.

El pic simplemente envia los datos o comando dependiendo de la acción que le indica la PI que realizará.

Configurar el programa en el I2C para recibir 3 bytes.

Ing. Francisco Sánchez Díaz  
2000-0001P  
sanch@laboatua.uioa.mx  
sanch\_sant@yanac.com.mx  
sanch\_sant@notmail.com

```

#include <18F4550.h>
#define ADC = 10
#define HSPLL, NOWDT, NOPROTECT, NOLVP, NODEBUG, USBDIV, PLL5, CPUDI/V1, VREGEN
#define delay(clock=48000000) //Una instruccion en 0.2us

#include <math.h> //Libreria estandar operaciones matematicas

#define USB_HID_DEVICE FALSE //No se trata de un dispositivo HID
#define USB_EP1_TX_ENABLE USB_ENABLE_BULK //Se habilita EP1(EndPoint1) for IN bulk transfers
#define USB_EP1_RX_ENABLE USB_ENABLE_BULK //Se habilita EP1(EndPoint1) for OUT bulk transfers
#define USB_EP1_TX_SIZE 64 //Memoria a reservar para el endpoint tx (Inpipe)
#define USB_EP1_RX_SIZE 3 //Memoria a reservar para el endpoint rx (Outpipe)

#include <pic18_usb.h> //Carga de las librerias USB.
#include <usb_desc_scope_alt.h> //Carga el header con las definiciones.
#include <usb.c>

//Declara constantes a utilizar por el preprocesador

#define comand usb_indata[0]
#define data usb_indata[1]
#define resolucion 0.004848

#define val_1 PIN_D7
#define val_2 PIN_D6
#define val_3 PIN_D5
#define val_4 PIN_D4
#define val_5 PIN_D3
#define val_8 PIN_D2

#define ups_senal PIN_E2
#define ups_cont PIN_E1
#define horno PIN_C2

#define neumatica PIN_C6
#define purga PIN_C1
#define modo PIN_C0
#define led PIN_A4

#define fast_io(A)
#define fast_io(B)
#define fast_io(C)
#define fast_io(D)
#define fast_io(E)

#define PORTA = 0x0F80
#define PORTB = 0x0F81
#define PORTC = 0x0F82
#define PORTD = 0x0F83
#define PORTE = 0x0F84

```

```

//declaracion de variables autilizar dentro del programa

char Datos[8] = {0 0 0 0 0 0 0};

char comando[8] = {0 0 0 0 0 0 0};

int8 i,val_purga;

int16 average;

float celsius,humedad,flujo,vacio,absorcion,interna;

//Funcion utilizada para configurar los puertos del microcontrolador

void inicializar(void)
{
    set_tris_a(0b00101111);    // RA0,1,2,3,5 Analog Input
    output_a(0x00);

    set_tris_b(0b00000000);    // Puerto B como salida
    output_b(0x00);

    set_tris_c(0b01000000);    // Puerto C como salida(Exepto RC6)
    output_c(0x00);

    set_tris_d(0b00000000);    // Puerto D como salida
    output_d(0x00);

    set_tris_e(0b00000001);    // Puerto E como salida(Exepto RE0)
    output_e(0x00);

    setup_adc( ADC_CLOCK_INTERNAL );
    setup_adc_ports(AN0_TO_AN5_ANALOG | VSS_VDD);
}

```

```

//Adquiere temperatura externa
float voltage_0(void)
{
  set_adc_channel(0);
  delay_ms(5);
  average = 0;

  for(i=0;i<64;i++)
  {
    average += read_adc();
  }
  average = average/64;

  return celsius = (float)(((average * resolution)/(0.01)) - (50));

}

//Adquiere humedad externa
float voltage_1(void)
{
  set_adc_channel(1);
  delay_ms(5);
  average = 0;

  for(i=0;i<64;i++)
  {
    average += read_adc();
  }
  average = average/64;

  return humedad = (float)((1/0.0062)*(((average * resolution)/(4.95)) - (0.16)));

}

//Adquiere humedad interna
float voltage_2(void)
{
  set_adc_channel(2);
  delay_ms(5);
  average = 0;

  for(i=0;i<64;i++)
  {
    average += read_adc();
  }
  average = average/64;

  return Interna = (float)(((average * resolution)/(0.02)) + (-110));

}

```

```

//Adquiere flujo de hidrogeno
float voltage_3(void)
{
  set_adc_channel(3);
  delay_ms(5);
  average = 0;

  for(i=0;i<64;i++)
  {
    average += read_adc();
  }
  average = average/64;

  return flujo = (float){((average * resolucion) * (100))/(4.95)};

}

//Adquiere valor de vacio jet
float voltage_4(void)
{
  set_adc_channel(4);
  delay_ms(5);
  average = 0;

  for(i=0;i<64;i++)
  {
    average += read_adc();
  }
  average = average/64;

  return vacio = (float){((average * resolucion) / (21)) * (100)};

}

// Adquiere valor de la bombas de absorcion
float voltage_5(void)
{
  float y_0 = -70.79104;
  float a_1 = 2522.27752;
  float t_1 = -0.84633;
  float a_2 = 1159880;
  float t_2 = -0.17733;
  float a_3 = 333.08129;
  float t_3 = -6.37121;
  float volts = 0;

  set_adc_channel(5);
  delay_ms(5);
  average = 0;

  for(i=0;i<64;i++)
  {
    average += read_adc();
  }
  average = average/64;
  volts = (float){((average * resolucion) * 2)};

  return absorcion = (float){(a_1 * exp(volts/t_1)) + (a_2 * exp(volts/t_2)) +
    {a_3 * exp(volts/t_3)} + (y_0)};

}

```

```

//On valvula 1
void voltage_6(void)
{
  output_high(val_1);
  delay_ms(5);

  output_low(PIN_C7);
  output_high(PIN_D1);
  output_low(PIN_D0);
  delay_ms(5);

  if(input(neumatica))
  {
    comando[0] = 'T';
    comando[1] = 'r';
    comando[2] = 'u';
    comando[3] = 'e';
    comando[4] = 'T';
    comando[5] = 'r';
    comando[6] = 'u';
    comando[7] = 'e';
  }
  else {
    comando[0] = 'F';
    comando[1] = 'a';
    comando[2] = 'l';
    comando[3] = 's';
    comando[4] = 'F';
    comando[5] = 'a';
    comando[6] = 'l';
    comando[7] = 's';
  }
}

```

```

//On valvula 2
void voltage_7(void)
{
  output_high(val_2);
  delay_ms(5);

  output_low(PIN_C7);
  output_high(PIN_D1);
  output_high(PIN_D0);
  delay_ms(5);

  if(Input(neumatica))
  {
    comando[0] = 'T';
    comando[1] = 'r';
    comando[2] = 'u';
    comando[3] = 'e';
    comando[4] = 'T';
    comando[5] = 'r';
    comando[6] = 'u';
    comando[7] = 'e';
  }
  else {
    comando[0] = 'F';
    comando[1] = 'a';
    comando[2] = 'l';
    comando[3] = 's';
    comando[4] = 'F';
    comando[5] = 'a';
    comando[6] = 'l';
    comando[7] = 's';
  }
}

```

```

//On valvula 3
void voltage_8(void)
{
  output_high(val_3);
  delay_ms(5);

  output_high(PIN_C7);
  output_low(PIN_D1);
  output_low(PIN_D0);
  delay_ms(5);

  if(input(neumatica))
  {
    comando[0] = 'T';
    comando[1] = 'r';
    comando[2] = 'u';
    comando[3] = 'e';
    comando[4] = 'T';
    comando[5] = 'r';
    comando[6] = 'u';
    comando[7] = 'e';
  }
  else {
    comando[0] = 'F';
    comando[1] = 'a';
    comando[2] = 'l';
    comando[3] = 's';
    comando[4] = 'F';
    comando[5] = 'a';
    comando[6] = 'l';
    comando[7] = 's';
  }
}

```

```

//Open valvula 4
void voltage_9(void)
{
  output_high(val_4);
  delay_us(5);

  output_high(PIN_C7);
  output_low(PIN_D1);
  output_high(PIN_D0);
  delay_ms(5);

  if(input(neumatica))
  {
    comando[0] = 'T';
    comando[1] = 'r';
    comando[2] = 'u';
    comando[3] = 'e';
    comando[4] = 'T';
    comando[5] = 'r';
    comando[6] = 'u';
    comando[7] = 'e';
  }
  else {
    comando[0] = 'F';
    comando[1] = 'a';
    comando[2] = 'l';
    comando[3] = 's';
    comando[4] = 'F';
    comando[5] = 'a';
    comando[6] = 'l';
    comando[7] = 's';
  }
}

```



```

//Open valvula 5
void voltage_10(void)
{
  output_high(val_5);
  delay_ms(5);

  output_high(PIN_C7);
  output_high(PIN_D1);
  output_low(PIN_D0);
  delay_ms(5);

  if(input(neumatica))
  {
    comando[0] = 'T';
    comando[1] = 'r';
    comando[2] = 'u';
    comando[3] = 'e';
    comando[4] = 'T';
    comando[5] = 'r';
    comando[6] = 'u';
    comando[7] = 'e';
  }
  else {
    comando[0] = 'F';
    comando[1] = 'a';
    comando[2] = 'l';
    comando[3] = 's';
    comando[4] = 'F';
    comando[5] = 'a';
    comando[6] = 'l';
    comando[7] = 's';
  }
}

//On valvula 8
void voltage_11(void)
{
  output_high(val_8);
  delay_ms(5);

  output_high(PIN_C7);
  output_high(PIN_D1);
  output_high(PIN_D0);
  delay_ms(5);

  if(input(neumatica))
  {
    comando[0] = 'T';
    comando[1] = 'r';
    comando[2] = 'u';
    comando[3] = 'e';
    comando[4] = 'T';
    comando[5] = 'r';
    comando[6] = 'u';
    comando[7] = 'e';
  }
  else {
    comando[0] = 'F';
    comando[1] = 'a';
    comando[2] = 'l';
    comando[3] = 's';
    comando[4] = 'F';
    comando[5] = 'a';
    comando[6] = 'l';
    comando[7] = 's';
  }
}

```

```

// On purga
void voltage_12(void)
{
  output_high(purga);
  delay_ms(5);

  val_purga = PORTC;

  if(bit_test(val_purga,1))
  {
    comando[0] = 'T';
    comando[1] = 'r';
    comando[2] = 'u';
    comando[3] = 'e';
    comando[4] = 'T';
    comando[5] = 'r';
    comando[6] = 'u';
    comando[7] = 'e';
  }
  else {
    comando[0] = 'F';
    comando[1] = 'a';
    comando[2] = 'l';
    comando[3] = 's';
    comando[4] = 'F';
    comando[5] = 'a';
    comando[6] = 'l';
    comando[7] = 's';
  }
}

//Horno On
void voltage_13(void)
{
  output_high(horno);
  delay_rns(5);

  val_purga = PORTC;

  if(bit_test(val_purga,2))
  {
    comando[0] = 'T';
    comando[1] = 'r';
    comando[2] = 'u';
    comando[3] = 'e';
    comando[4] = 'T';
    comando[5] = 'r';
    comando[6] = 'u';
    comando[7] = 'e';
  }
  else {
    comando[0] = 'F';
    comando[1] = 'a';
    comando[2] = 'l';
    comando[3] = 's';
    comando[4] = 'F';
    comando[5] = 'a';
    comando[6] = 'l';
    comando[7] = 's';
  }
}

```

```

// On valvula UP1
void voltage_14(void)
{
  output_high(ups_senal);
  delay_ms(5);
  output_high(ups_cont);
  delay_ms(5);

  output_low(PIN_C7);
  output_low(PIN_D1);
  output_low(PIN_D0);
  delay_ms(5);

  if(input(neumatica))
  {
    comando[0] = 'T';
    comando[1] = 'r';
    comando[2] = 'u';
    comando[3] = 'e';
    comando[4] = 'T';
    comando[5] = 'r';
    comando[6] = 'u';
    comando[7] = 'e';
  }
  else {
    comando[0] = 'F';
    comando[1] = 'a';
    comando[2] = 'l';
    comando[3] = 's';
    comando[4] = 'F';
    comando[5] = 'a';
    comando[6] = 'l';
    comando[7] = 's';
  }
}

// On valvula UP2
void voltage_15(void)
{
  output_low(ups_senal);
  delay_ms(5);
  output_high(ups_cont);
  delay_ms(5);

  output_low(PIN_C7);
  output_low(PIN_D1);
  output_high(PIN_D0);
  delay_ms(5);

  if(input(neumatica))
  {
    comando[0] = 'T';
    comando[1] = 'r';
    comando[2] = 'u';
    comando[3] = 'e';
    comando[4] = 'T';
    comando[5] = 'r';
    comando[6] = 'u';
    comando[7] = 'e';
  }
  else {
    comando[0] = 'F';
    comando[1] = 'a';
    comando[2] = 'l';
    comando[3] = 's';
    comando[4] = 'F';
    comando[5] = 'a';
    comando[6] = 'l';
    comando[7] = 's';
  }
}

```

```

//Off valvula 1
void voltage_16(void)
{
  output_low(val_1);
  delay_ms(5);

  output_low(PIN_C7);
  output_high(PIN_D1);
  output_low(PIN_D0);
  delay_ms(5);

  if(!input(neumatica))
  {
    comando[0] = 'T';
    comando[1] = 'r';
    comando[2] = 'u';
    comando[3] = 'e';
    comando[4] = 'T';
    comando[5] = 'r';
    comando[6] = 'u';
    comando[7] = 'e';
  }
  else {
    comando[0] = 'F';
    comando[1] = 'a';
    comando[2] = 'l';
    comando[3] = 's';
    comando[4] = 'F';
    comando[5] = 'a';
    comando[6] = 'l';
    comando[7] = 's';
  }
}

//Off valvula 2
void voltage_17(void)
{
  output_low(val_2);
  delay_ms(5);

  output_low(PIN_C7);
  output_high(PIN_D1);
  output_high(PIN_D0);
  delay_ms(5);

  if(!input(neumatica))
  {
    comando[0] = 'T';
    comando[1] = 'r';
    comando[2] = 'u';
    comando[3] = 'e';
    comando[4] = 'T';
    comando[5] = 'r';
    comando[6] = 'u';
    comando[7] = 'e';
  }
  else {
    comando[0] = 'F';
    comando[1] = 'a';
    comando[2] = 'l';
    comando[3] = 's';
    comando[4] = 'F';
    comando[5] = 'a';
    comando[6] = 'l';
    comando[7] = 's';
  }
}

```

```

//Off valvula 3
void voltage_18(void)
{
  output_low(val_3);
  delay_ms(5);

  output_high(PIN_C7);
  output_low(PIN_D1);
  output_low(PIN_D0);
  delay_ms(5);

  if(!input(neumatica))
  {
    comando[0] = 'T';
    comando[1] = 'r';
    comando[2] = 'u';
    comando[3] = 'e';
    comando[4] = 'T';
    comando[5] = 'r';
    comando[6] = 'u';
    comando[7] = 'e';
  }
  else {
    comando[0] = 'F';
    comando[1] = 'a';
    comando[2] = 'l';
    comando[3] = 's';
    comando[4] = 'F';
    comando[5] = 'a';
    comando[6] = 'l';
    comando[7] = 's';
  }
}

```

```

//Off valvula 4
void voltage_19(void)
{
  output_low(val_4);
  delay_ms(5);

  output_high(PIN_C7);
  output_low(PIN_D1);
  output_high(PIN_D0);
  delay_ms(5);

  if(!input(neumatica))
  {
    comando[0] = 'T';
    comando[1] = 'r';
    comando[2] = 'u';
    comando[3] = 'e';
    comando[4] = 'T';
    comando[5] = 'r';
    comando[6] = 'u';
    comando[7] = 'e';
  }
  else {
    comando[0] = 'F';
    comando[1] = 'a';
    comando[2] = 'l';
    comando[3] = 's';
    comando[4] = 'F';
    comando[5] = 'a';
    comando[6] = 'l';
    comando[7] = 's';
  }
}

```

```

// Off valvula 5
void voltage_20(void)
{
  output_low(val_5);
  delay_ms(5);

  output_high(PIN_C7);
  output_high(PIN_D1);
  output_low(PIN_D0);
  delay_ms(5);

  if(!input(neumatica))
  {
    comando[0] = 'T';
    comando[1] = 'r';
    comando[2] = 'u';
    comando[3] = 'e';
    comando[4] = 'T';
    comando[5] = 'r';
    comando[6] = 'u';
    comando[7] = 'e';
  }
  else {
    comando[0] = 'F';
    comando[1] = 'a';
    comando[2] = 'l';
    comando[3] = 's';
    comando[4] = 'F';
    comando[5] = 'a';
    comando[6] = 'l';
    comando[7] = 's';
  }
}

```

```

//Off valvula 8
void voltage_21(void)
{
  output_low(val_8);
  delay_ms(5);

  output_high(PIN_C7);
  output_high(PIN_D1);
  output_high(PIN_D0);
  delay_ms(5);

  if(!input(neumatica))
  {
    comando[0] = 'T';
    comando[1] = 'r';
    comando[2] = 'u';
    comando[3] = 'e';
    comando[4] = 'T';
    comando[5] = 'r';
    comando[6] = 'u';
    comando[7] = 'e';
  }
  else {
    comando[0] = 'F';
    comando[1] = 'a';
    comando[2] = 'l';
    comando[3] = 's';
    comando[4] = 'F';
    comando[5] = 'a';
    comando[6] = 'l';
    comando[7] = 's';
  }
}

```

```

// Off switch de purga
void voltage_22(void)
{
    output_low(purga);
    delay_ms(5);

    val_purga = PORTC;

    if(!bit_test(val_purga,1))
    {
        comando[0] = 'T';
        comando[1] = 'r';
        comando[2] = 'u';
        comando[3] = 'e';
        comando[4] = 'T';
        comando[5] = 'r';
        comando[6] = 'u';
        comando[7] = 'e';
    }
    else {
        comando[0] = 'F';
        comando[1] = 'a';
        comando[2] = 'l';
        comando[3] = 's';
        comando[4] = 'F';
        comando[5] = 'a';
        comando[6] = 'l';
        comando[7] = 's';
    }
}

//Horno Off
void voltage_23(void)
{
    output_low(horno);
    delay_ms(5);

    val_purga = PORTC;

    if(!bit_test(val_purga,2))
    {
        comando[0] = 'T';
        comando[1] = 'r';
        comando[2] = 'u';
        comando[3] = 'e';
        comando[4] = 'T';
        comando[5] = 'r';
        comando[6] = 'u';
        comando[7] = 'e';
    }
    else {
        comando[0] = 'F';
        comando[1] = 'a';
        comando[2] = 'l';
        comando[3] = 's';
        comando[4] = 'F';
        comando[5] = 'a';
        comando[6] = 'l';
        comando[7] = 's';
    }
}

```

```

// Off valvula UP1
void voltage_24(void)
{
    output_low(ups_cont);
    delay_ms(5);

    output_low(ups_senal);
    delay_ms(5);

    output_low(PIN_C7);
    output_low(PIN_D1);
    output_low(PIN_D0);
    delay_ms(5);

    if(!input(neumatica))
    {
        comando[0] = 'T';
        comando[1] = 'r';
        comando[2] = 'u';
        comando[3] = 'e';
        comando[4] = 'T';
        comando[5] = 'r';
        comando[6] = 'u';
        comando[7] = 'e';
    }
    else {
        comando[0] = 'F';
        comando[1] = 'a';
        comando[2] = 'l';
        comando[3] = 's';
        comando[4] = 'F';
        comando[5] = 'a';
        comando[6] = 'l';
        comando[7] = 's';
    }
}

// Off valvula UP2
void voltage_25(void)
{
    output_low(ups_cont);
    delay_ms(5);

    output_low(ups_senal);
    delay_ms(5);

    output_low(PIN_C7);
    output_low(PIN_D1);
    output_high(PIN_D0);
    delay_ms(5);

    if(!input(neumatica))
    {
        comando[0] = 'T';
        comando[1] = 'r';
        comando[2] = 'u';
        comando[3] = 'e';
        comando[4] = 'T';
        comando[5] = 'r';
        comando[6] = 'u';
        comando[7] = 'e';
    }
    else {
        comando[0] = 'F';
        comando[1] = 'a';
        comando[2] = 'l';
        comando[3] = 's';
        comando[4] = 'F';
        comando[5] = 'a';
        comando[6] = 'l';
        comando[7] = 's';
    }
}

```

```

// Modo PC
void voltage_26(void)
{
    output_low(modos);
    delay_ms(5);

    val_purga = PORTC;

    if(!bit_test(val_purga,0))
    {
        comando[0] = 'T';
        comando[1] = 'r';
        comando[2] = 'u';
        comando[3] = 'e';
        comando[4] = 'T';
        comando[5] = 'r';
        comando[6] = 'u';
        comando[7] = 'e';
    }
    else {
        comando[0] = 'F';
        comando[1] = 'a';
        comando[2] = 'l';
        comando[3] = 's';
        comando[4] = 'F';
        comando[5] = 'a';
        comando[6] = 'l';
        comando[7] = 's';
    }
}
void main(void)
{

```

```

    int8 usb_indata[3];          //declaramos variables

```

```

    setup_adc_ports(NO_ANALOGS|VSS_VDD);
    setup_adc(ADC_OFF);
    setup_psp(PSP_DISABLED);
    setup_spi(SPI_SS_DISABLED);
    setup_wdt(WDT_OFF);
    setup_timer_0(RTCC_INTERNAL);
    setup_timer_1(T1_DISABLED);
    setup_timer_2(T2_DISABLED,0,1);
    setup_timer_3(T3_DISABLED|T3_DIV_BY_1);
    setup_comparator(NC_NC_NC_NC);
    setup_vref(FALSE);

```

```

    inicializar();
    output_high(led);

```

```

    usb_init();                //Se inicializa USB y espera que sea enumerado por la PC
    usb_task();
    usb_wait_for_enumeration();

```

```

// Modo Manual
void voltage_27(void)
{
    output_high(modos);
    delay_ms(5);

    val_purga = PORTC;

    if(bit_test(val_purga,0))
    {
        comando[0] = 'T';
        comando[1] = 'r';
        comando[2] = 'u';
        comando[3] = 'e';
        comando[4] = 'T';
        comando[5] = 'r';
        comando[6] = 'u';
        comando[7] = 'e';
    }
    else {
        comando[0] = 'F';
        comando[1] = 'a';
        comando[2] = 'l';
        comando[3] = 's';
        comando[4] = 'F';
        comando[5] = 'a';
        comando[6] = 'l';
        comando[7] = 's';
    }
}

```

```
while (TRUE) {  
  
  if(usb_enumerated())  
  {  
    if(usb_kbhit(1))  
    {  
      usb_get_packet(1,usb_indata,3);  
  
      switch (comand)  
      {  
        case 0x00://Temperatura externa  
  
          {  
            voltage_0();  
            sprintf(Datos,"%0.1f",celsius);  
            Datos[7] = 'A';  
            usb_puts(1,Datos,8,50);  
  
            break;  
          }  
  
        case 0x01://Humedad externa  
  
          {  
            voltage_1();  
            sprintf(Datos,"%0.1f",humedad);  
            Datos[7] = 'B';  
            usb_puts(1,Datos,8,50);  
  
            break;  
          }  
  
        case 0x02://Humedad interna  
  
          {  
            voltage_2();  
            sprintf(Datos,"%0.1f",interna);  
            Datos[7] = 'C';  
            usb_puts(1,Datos,8,50);  
  
            break;  
          }  
  
        case 0x03://Flujo de hidrogeno  
  
          {  
            voltage_3();  
            sprintf(Datos,"%0.1f",flujo);  
            Datos[7] = 'D';  
            usb_puts(1,Datos,8,50);  
  
            break;  
          }  
  
        case 0x04://Vacio jet  
  
          {  
            voltage_4();  
            sprintf(Datos,"%0.1f",vacio);  
            Datos[7] = 'E';  
            usb_puts(1,Datos,8,50);  
  
            break;  
          }  
  
        case 0x05://Valor absorcion  
  
          {  
            voltage_5();  
            sprintf(Datos,"%0.1f",absorcion);  
            Datos[7] = 'F';  
            usb_puts(1,Datos,8,50);  
  
            break;  
          }  
  
      }  
    }  
  }  
}
```

```
case 0x06://Valvula 1 On
```

```
{  
  voltage_6();  
  usb_puts(1,comando,8,50);  
  
  break;  
}
```

```
case 0x07://valvula 2 On
```

```
{  
  voltage_7();  
  usb_puts(1,comando,8,50);  
  
  break;  
}
```

```
case 0x08: //valvula 3 On
```

```
{  
  voltage_8();  
  usb_puts(1,comando,8,50);  
  
  break;  
}
```

```
case 0x09://valvula 4 On
```

```
{  
  voltage_9();  
  usb_puts(1,comando,8,50);  
  
  break;  
}
```

```
case 0x10://valvula 6 On
```

```
{  
  voltage_10();  
  usb_puts(1,comando,8,50);  
  
  break;  
}
```

```
case 0x11://Valvula 7 On
```

```
{  
  voltage_11();  
  usb_puts(1,comando,8,50);  
  
  break;  
}
```

```
case 0x12: //Purga On
```

```
{  
  voltage_12();  
  usb_puts(1,comando,8,50);  
  
  break;  
}
```

```
case 0x13: //Horno On
```

```
{  
  voltage_13();  
  usb_puts(1,comando,8,50);  
  
  break;  
}
```

```
case 0x14: //UP1 On
```

```
{  
  voltage_14();  
  usb_puts(1,comando,8,50);  
  
  break;  
}
```

```
case 0x15: //UP2 On
```

```
{  
  
  voltage_15();  
  usb_puts(1,comando,8,50);  
  
  break;  
}
```

```
case 0x16://valvula 1 Off
```

```
{  
  voltage_16();  
  usb_puts(1,comando,8,50);  
  
  break;  
}
```

```
case 0x17://valvula 2 Off
```

```
{  
  voltage_17();  
  usb_puts(1,comando,8,50);  
  
  break;  
}
```

```
case 0x18: //valvula 3 Off
```

```
{  
  voltage_18();  
  usb_puts(1,comando,8,50);  
  
  break;  
}
```

```
case 0x19: //valvula 4 Off
```

```
{  
  voltage_19();  
  usb_puts(1,comando,8,50);  
  
  break;  
}
```

```
case 0x20://Valvula 6 Off
```

```
{  
  voltage_20();  
  usb_puts(1,comando,8,50);  
  
  break;  
}
```

```
case 0x21://Valvula 7 Off
```

```
{  
  voltage_21();  
  usb_puts(1,comando,8,50);  
  
  break;  
}
```

```
case 0x22://Purga Off
```

```
{  
  voltage_22();  
  usb_puts(1,comando,8,50);  
  
  break;  
}
```

```
case 0x23: //Horno Off
```

```
{  
  voltage_23();  
  usb_puts(1,comando,8,50);  
  
  break;  
}
```

```
case 0x24://UP1 Off
```

```
{  
  voltage_24();  
  usb_puts(1,comando,8,50);  
  
  break;  
}
```

```
case 0x25: //UP2 Off
```

```
{  
  voltage_25();  
  usb_puts(1,comando,8,50);  
  
  break;  
}
```



```

case 0x26: //Modo PC

{
  voltage_26();
  usb_puts(1,comando,8,50);

  break;
}

case 0x27: //Modo manual

{
  voltage_27();
  usb_puts(1,comando,8,50);

  break;
}

case 0x28://Porcenta de Flujo de Hidrogeno

{
  output_b(data);
  delay_ms(20);

  comando[0] = 'T';
  comando[1] = 'r';
  comando[2] = 'u';
  comando[3] = 'e';
  comando[4] = 'T';
  comando[5] = 'r';
  comando[6] = 'u';
  comando[7] = 'e';

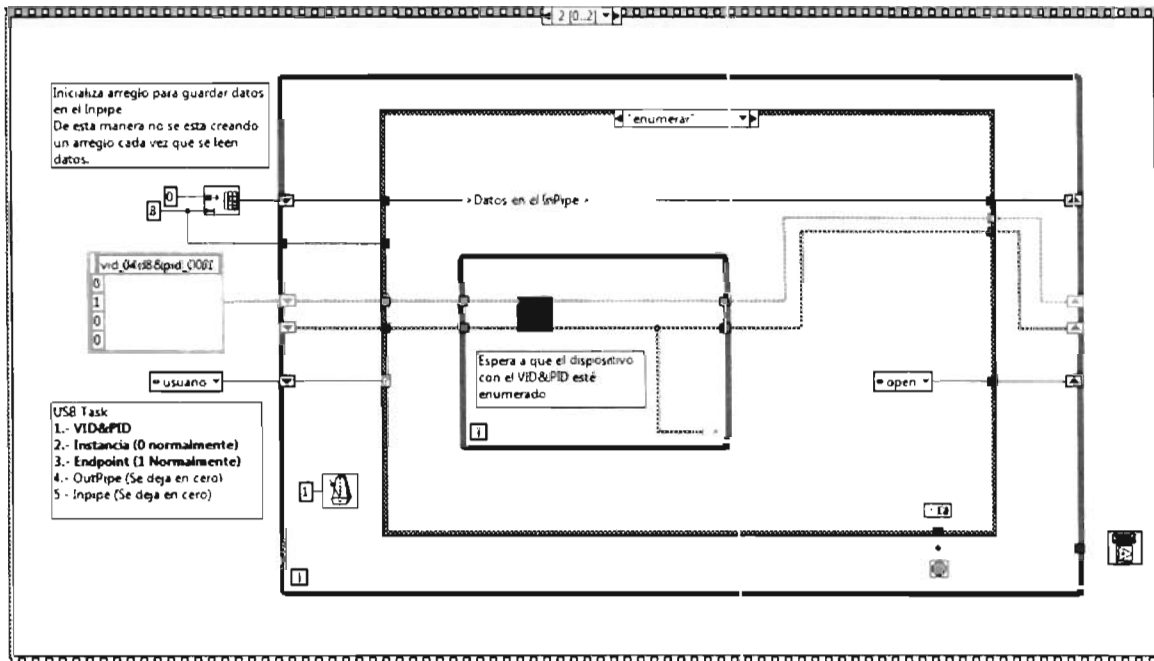
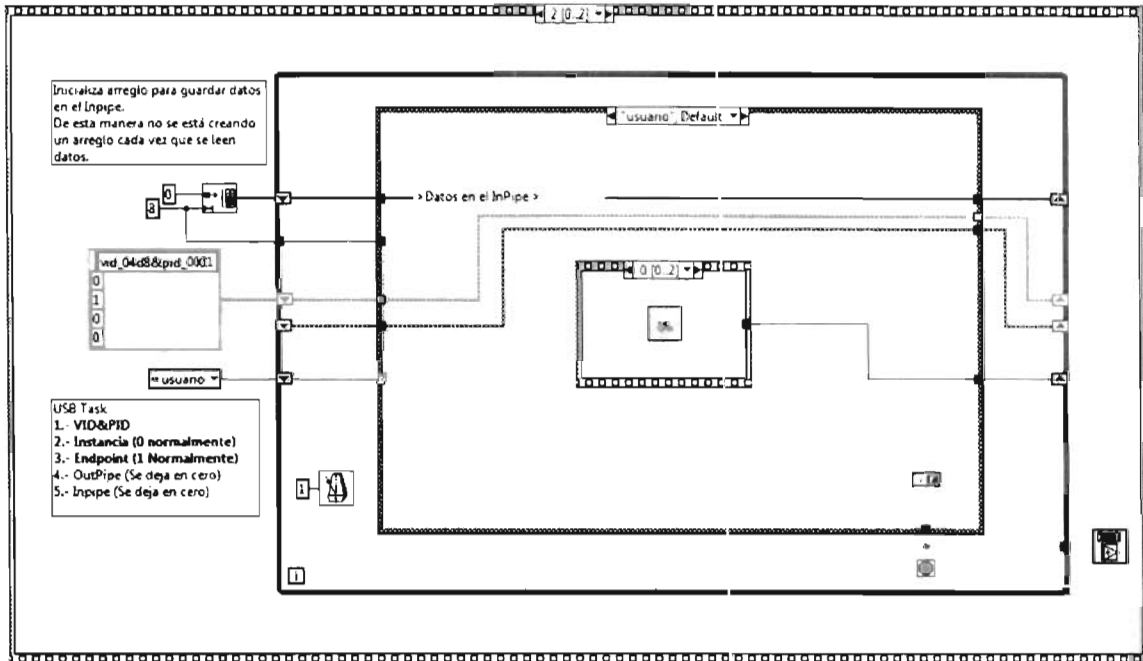
  usb_puts(1,comando,8,50);

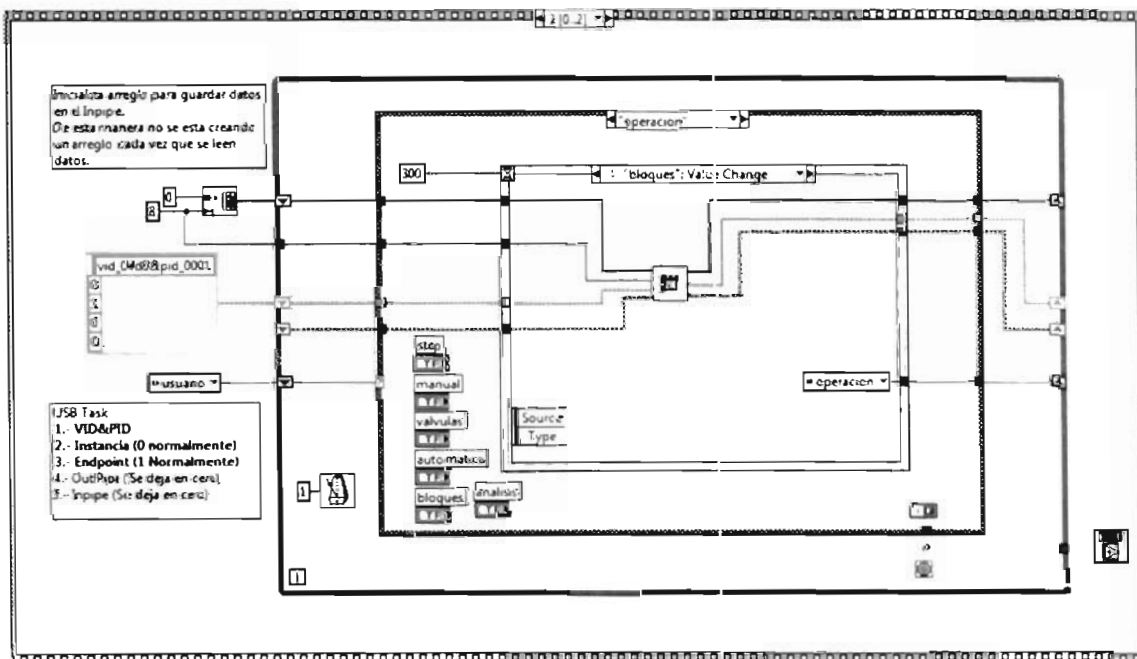
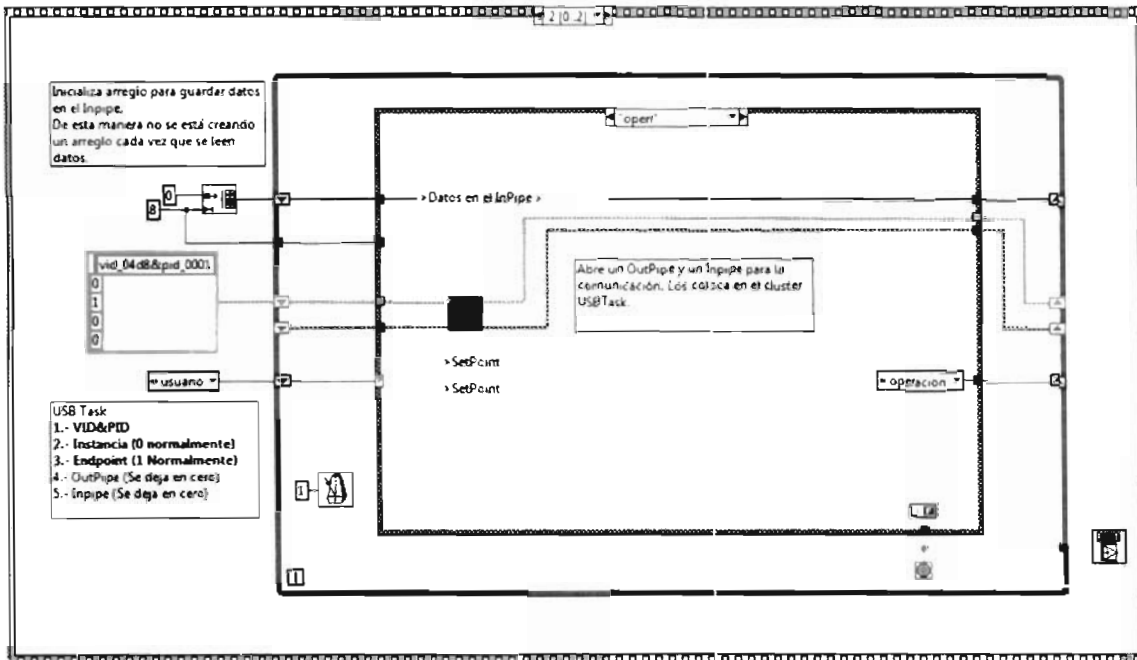
  break;
}

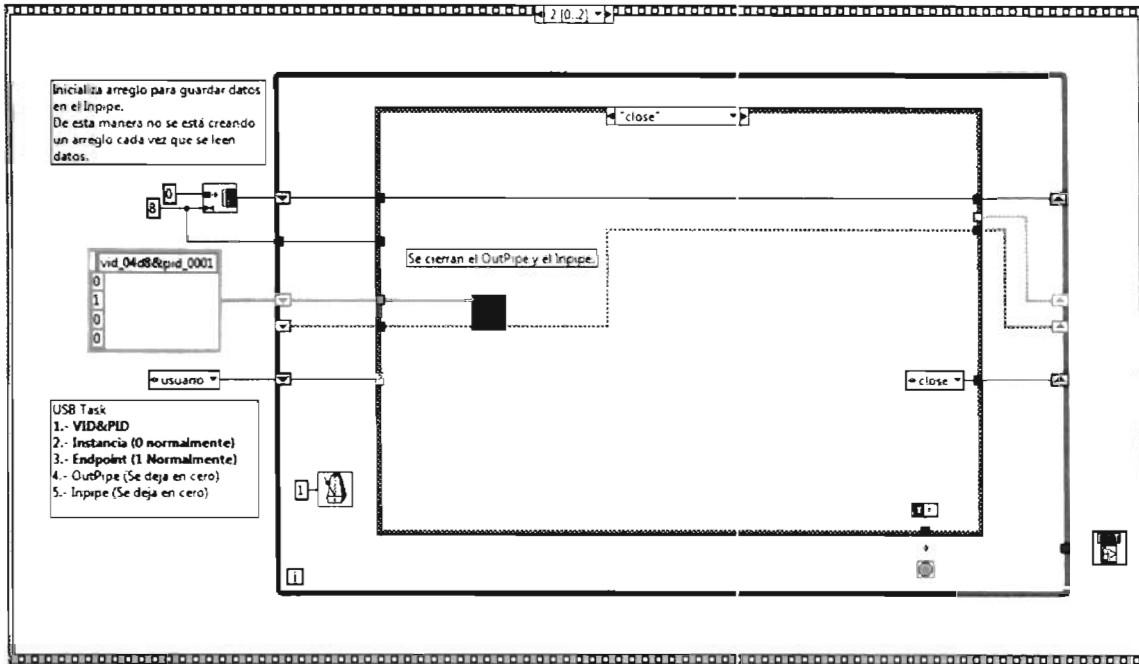
//Cierra switch
//Espera bytes en el buffer del usb
//Espera a que sea enumerado
//Cierra while
//Fin main

```

Programa principal de control de la PC implementado en LabVIEW.

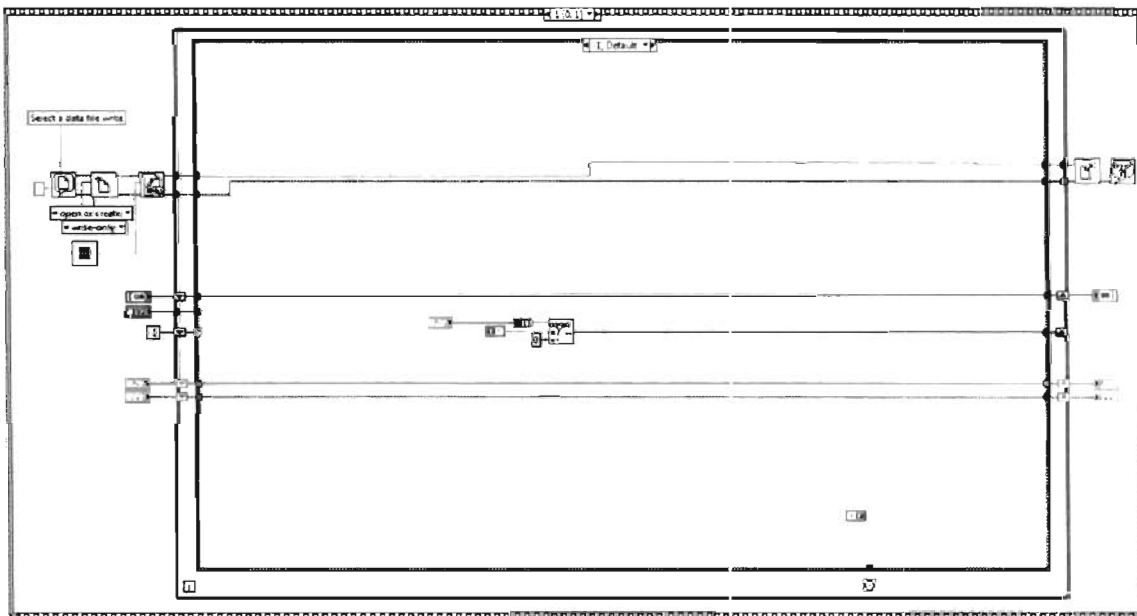


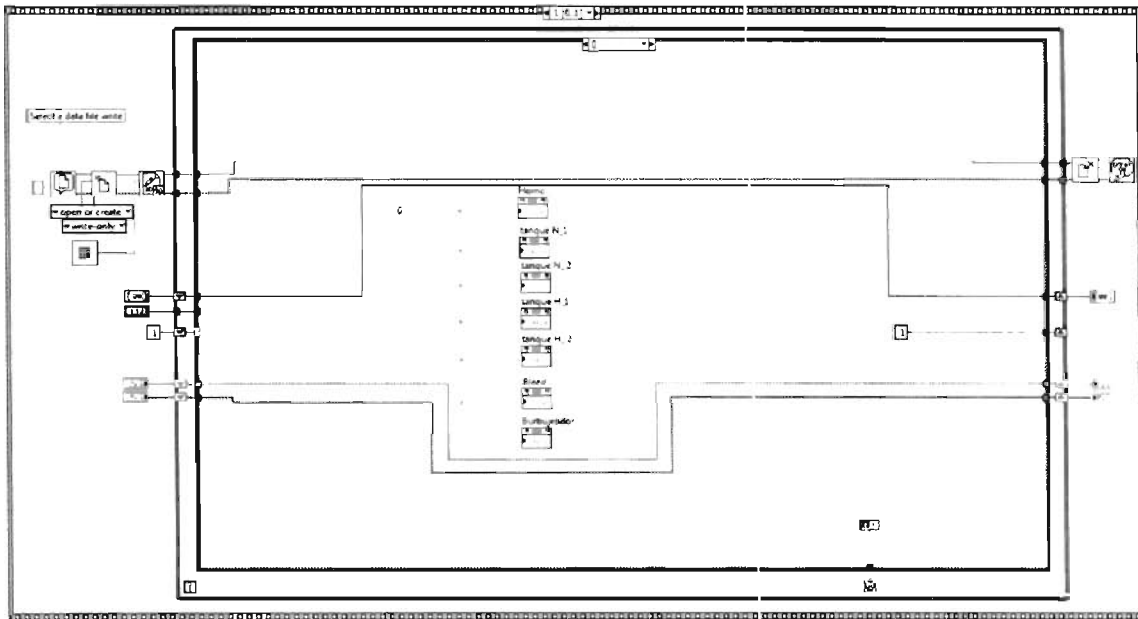
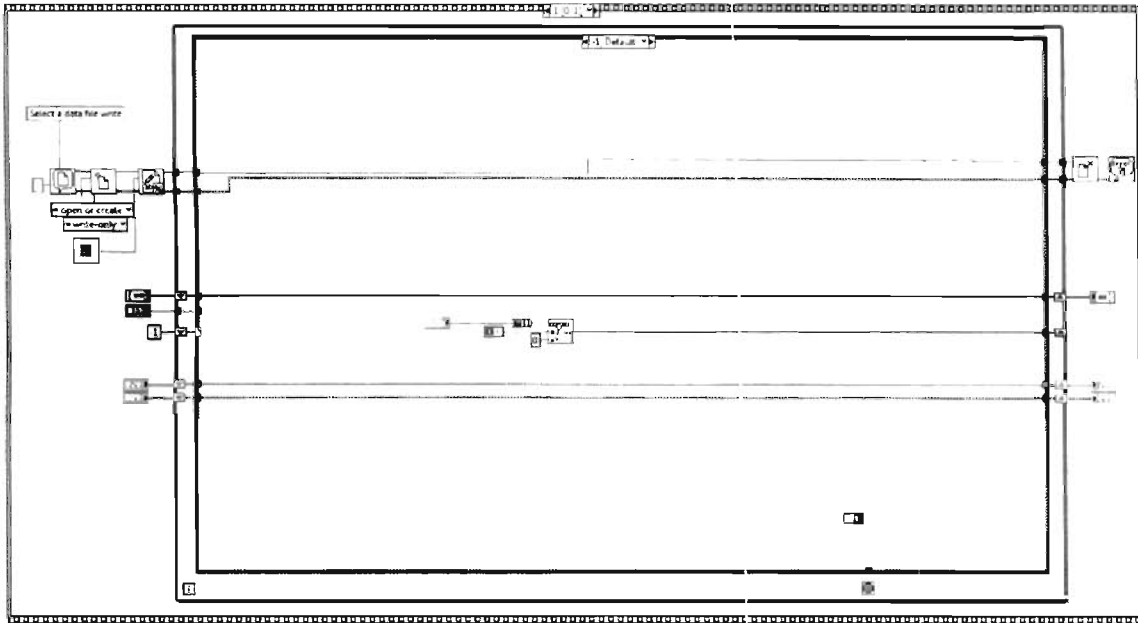


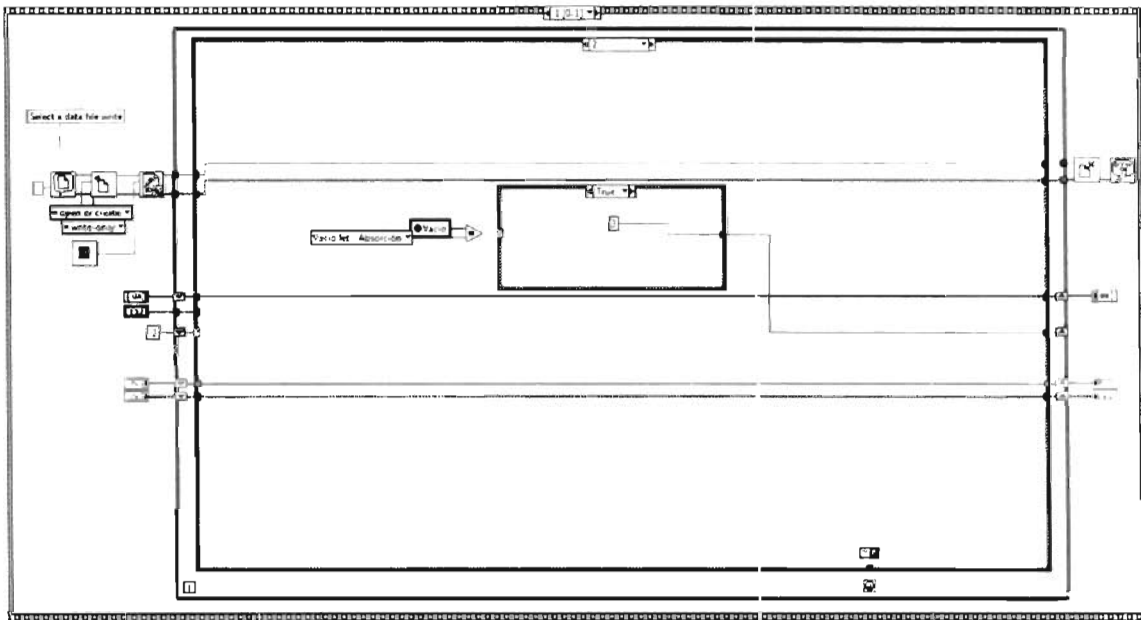
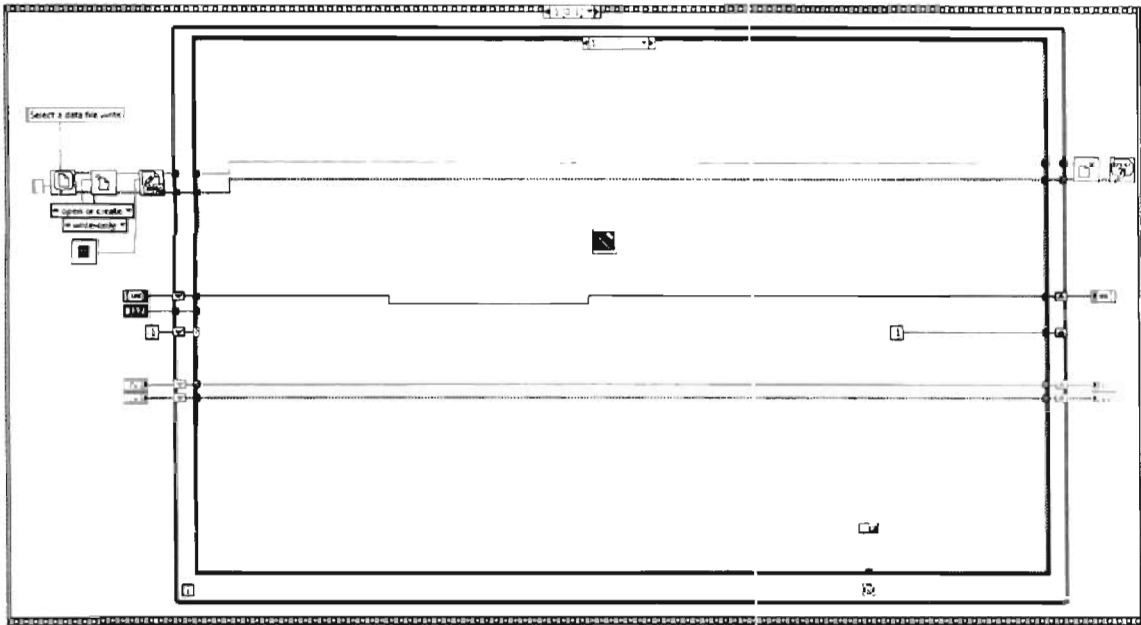


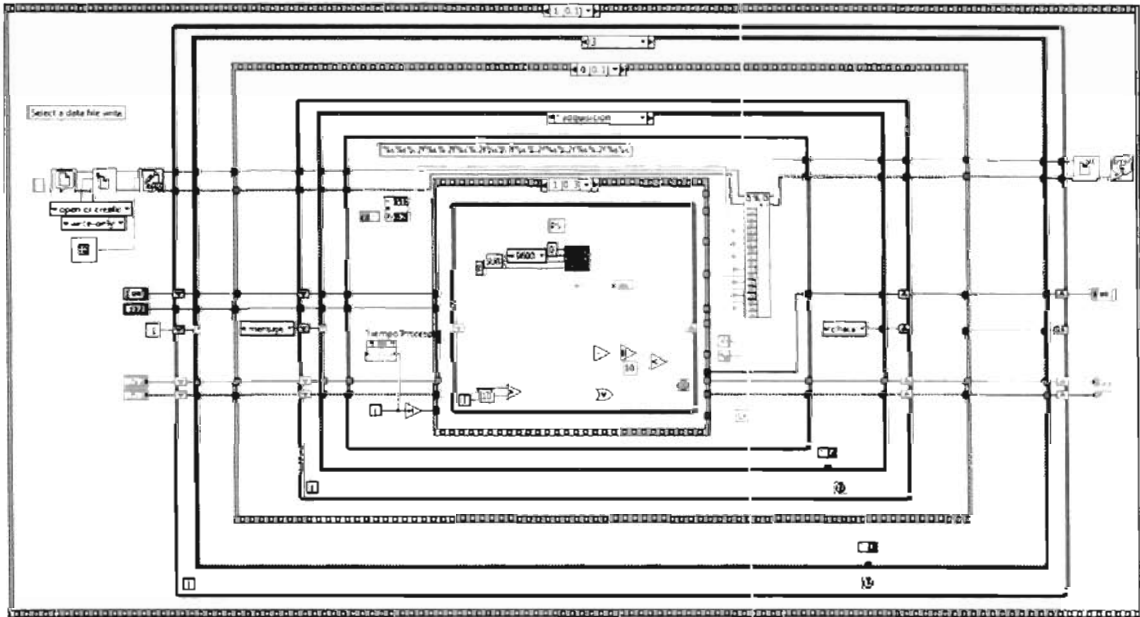
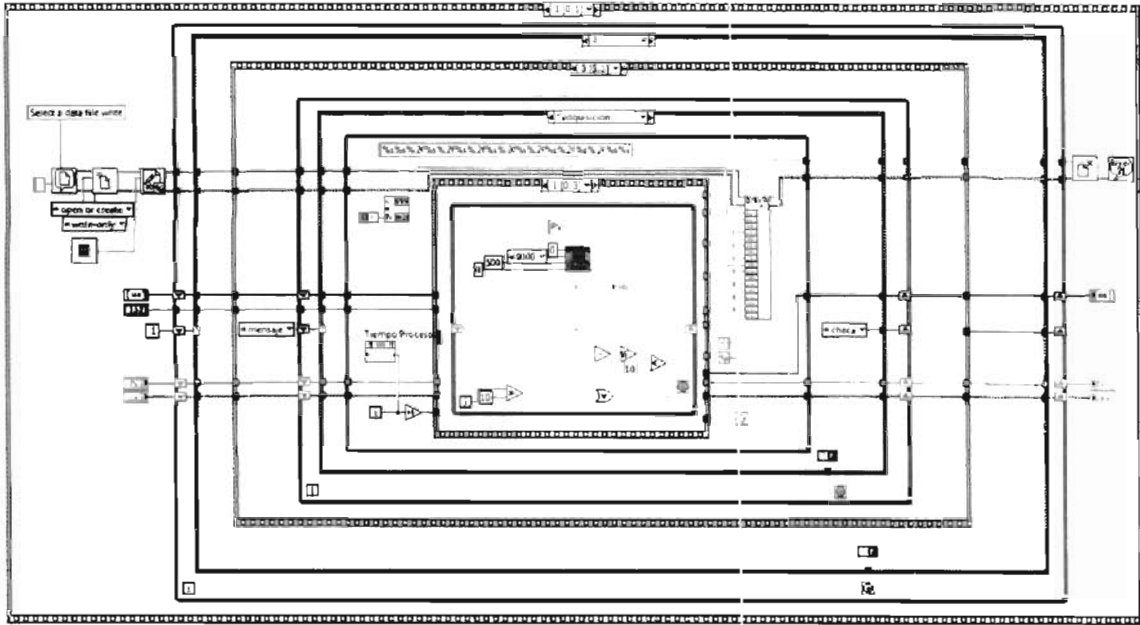
## Apéndice B.1

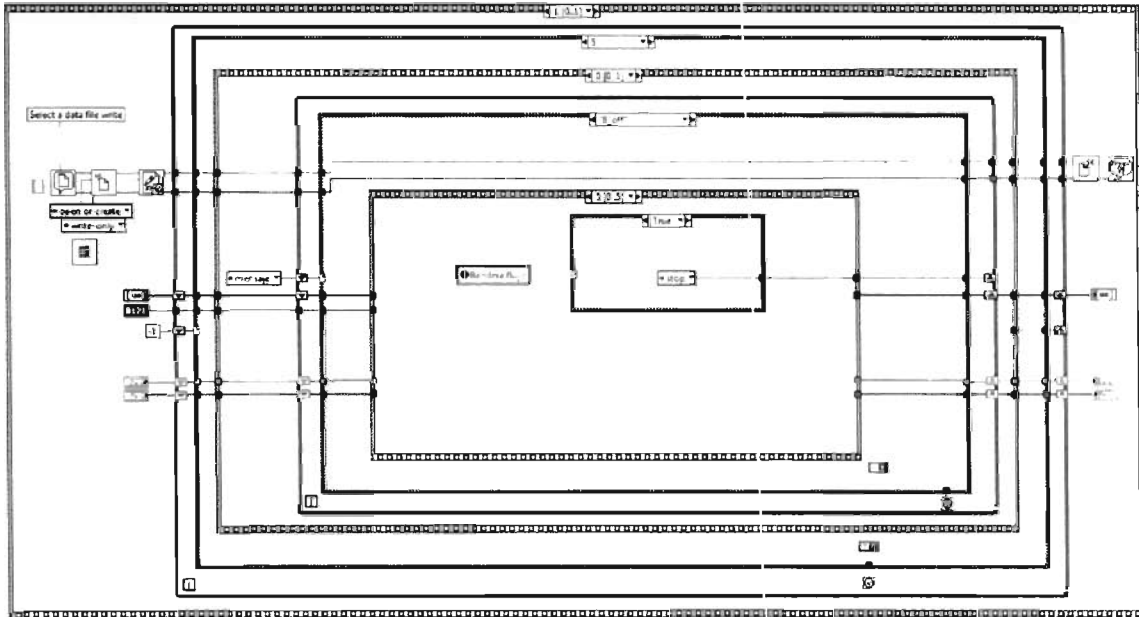
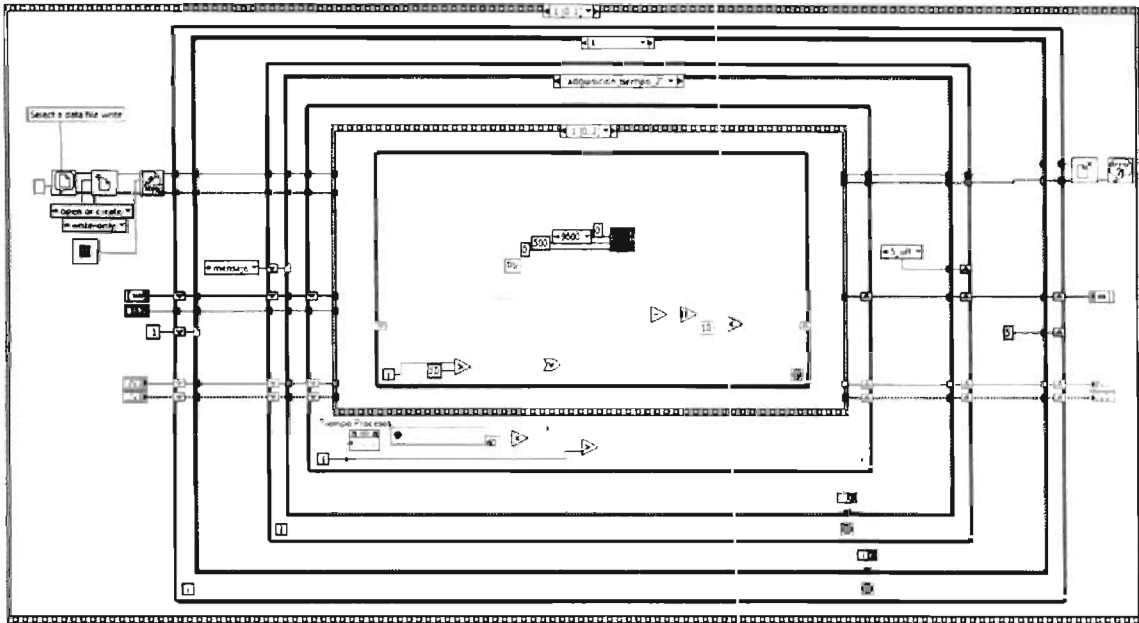
### Subprograma modo Automático.





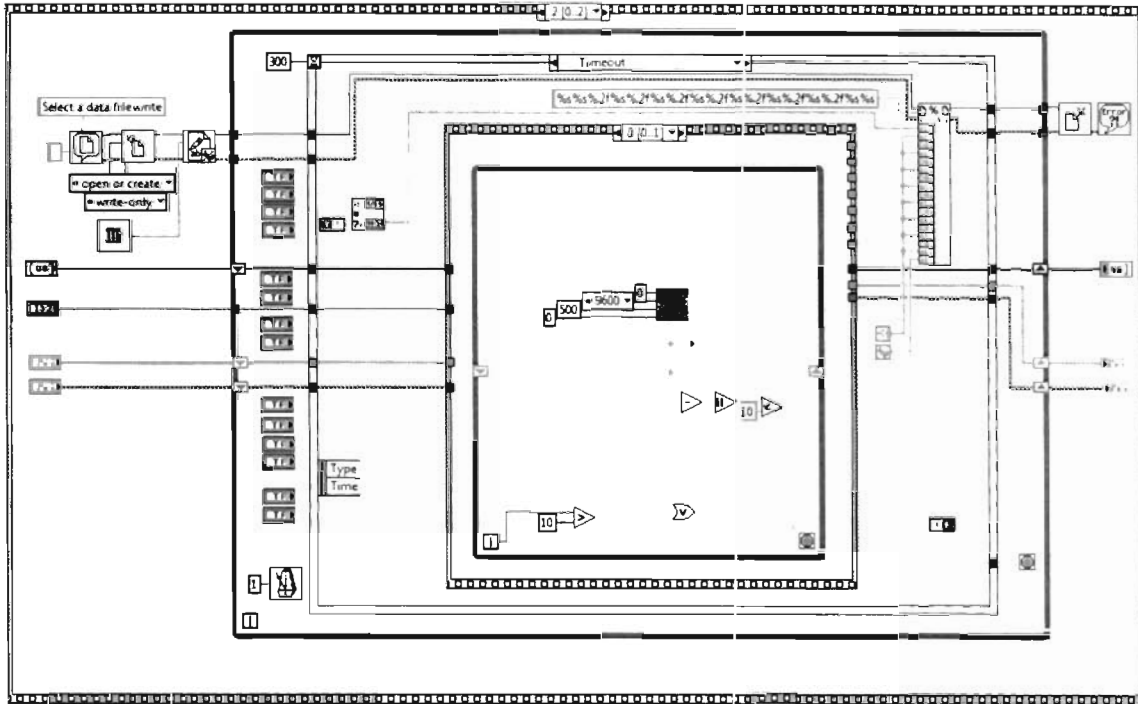




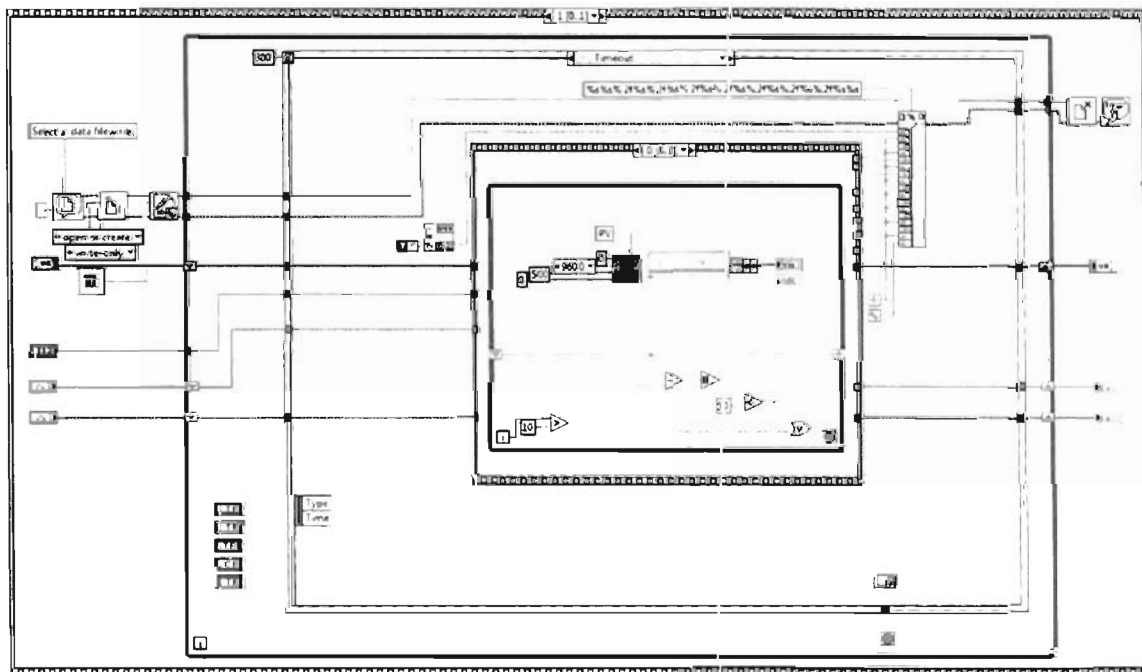




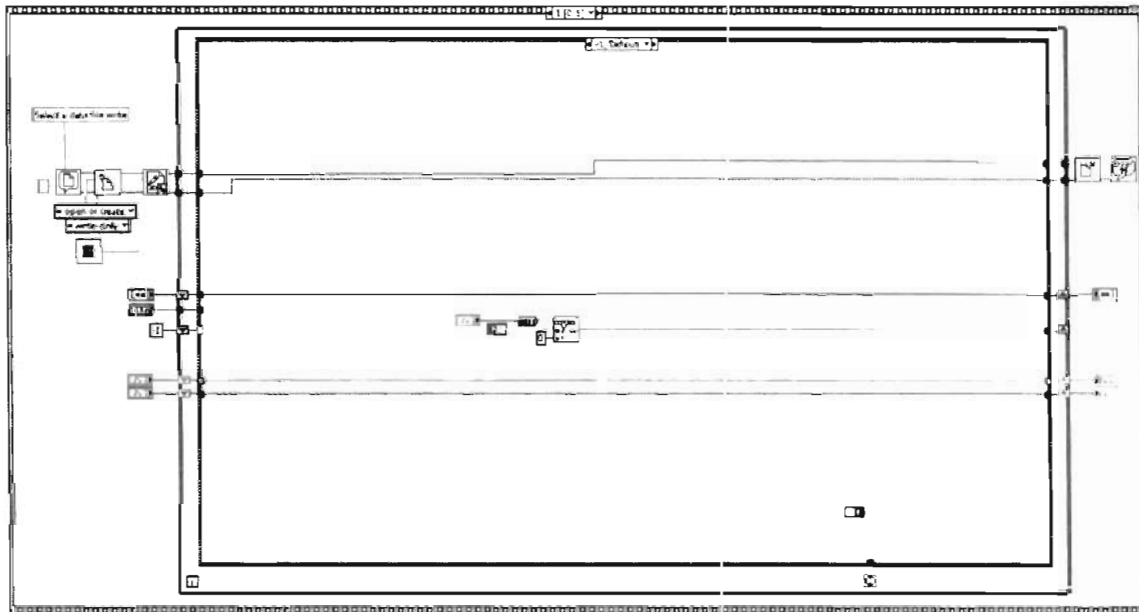
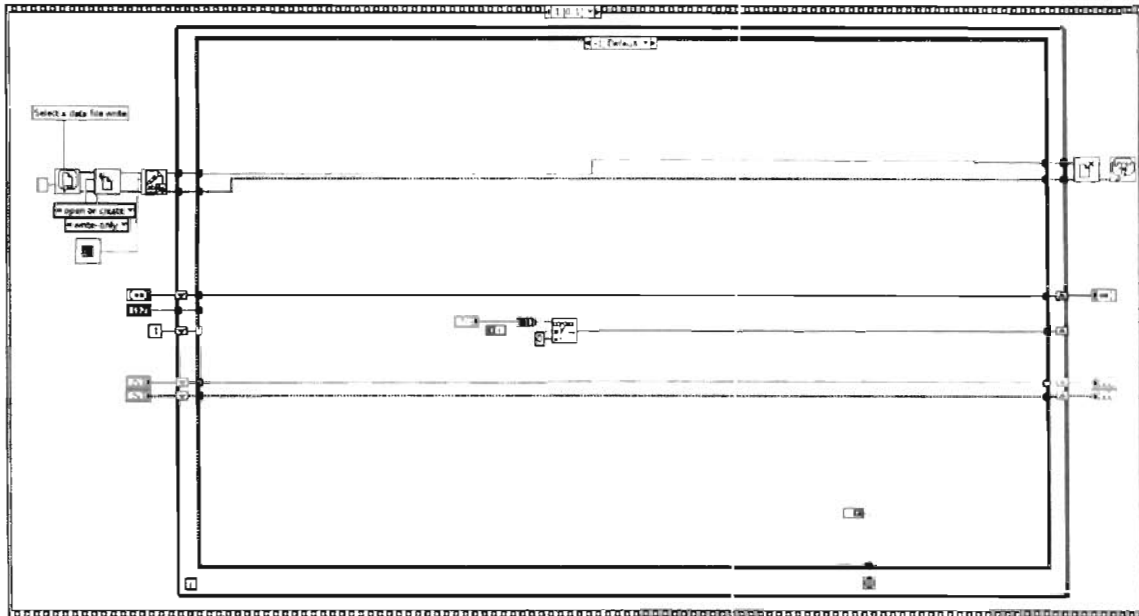
Subprograma modo Válvulas.

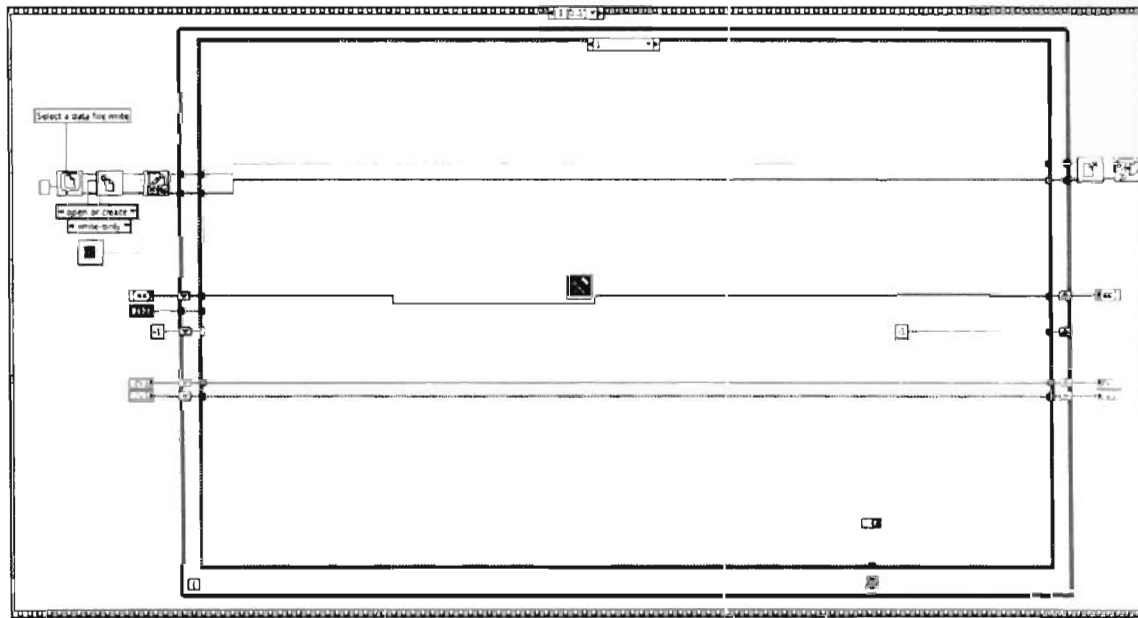
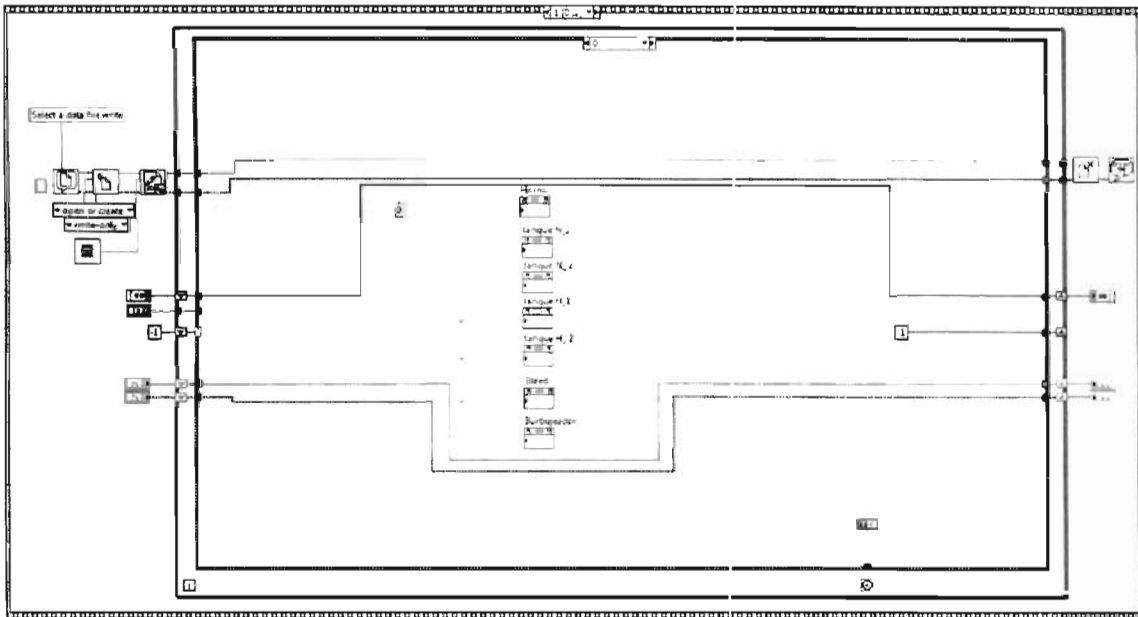


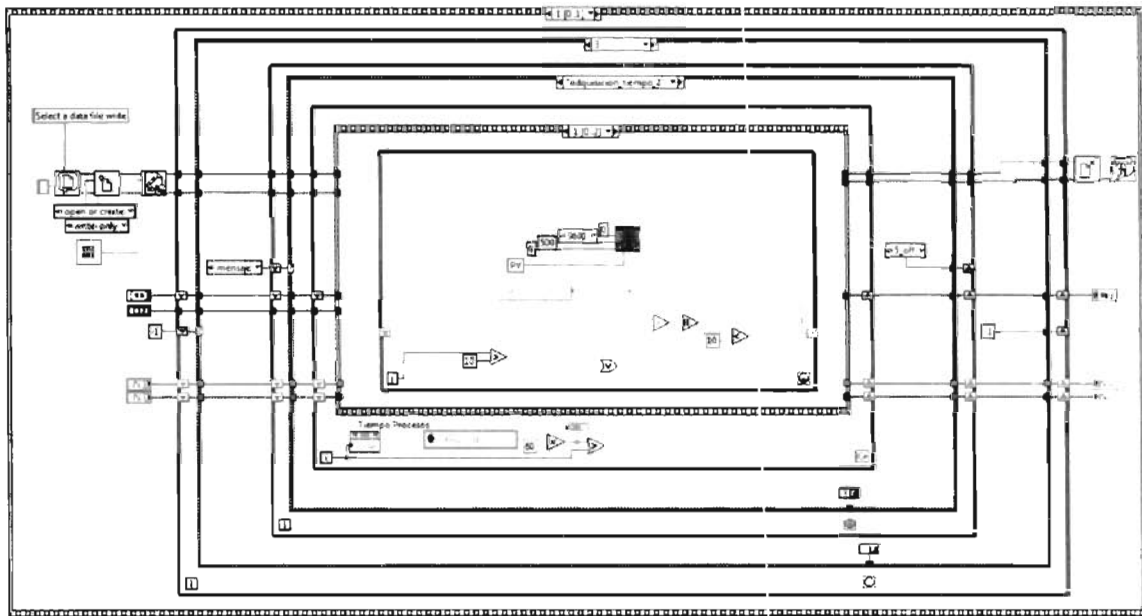
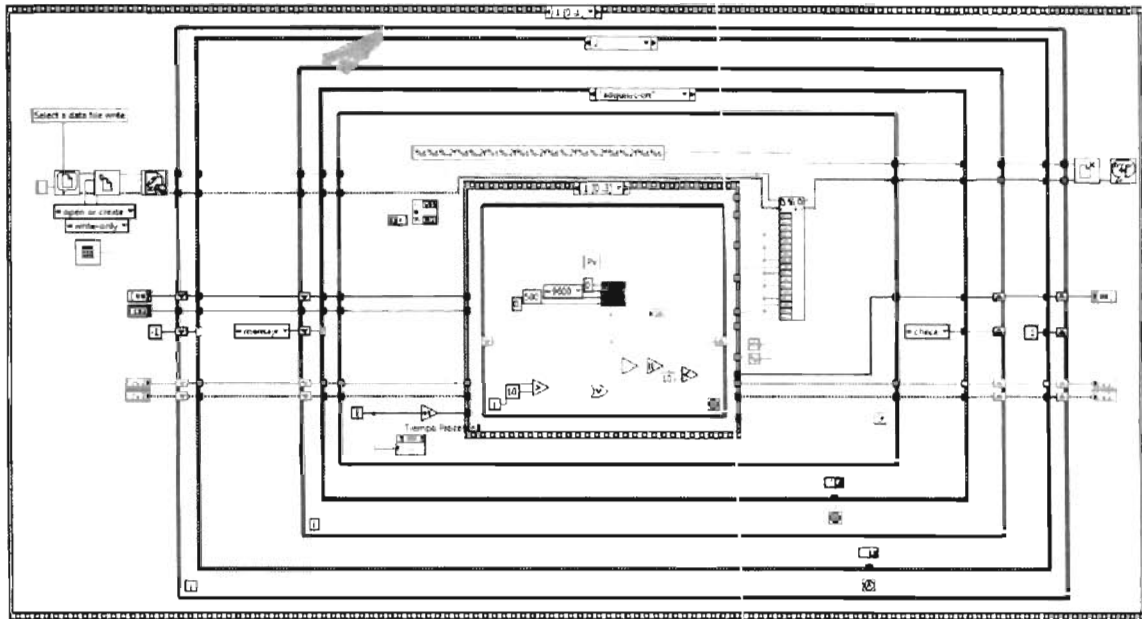
Subprograma modo Manual.

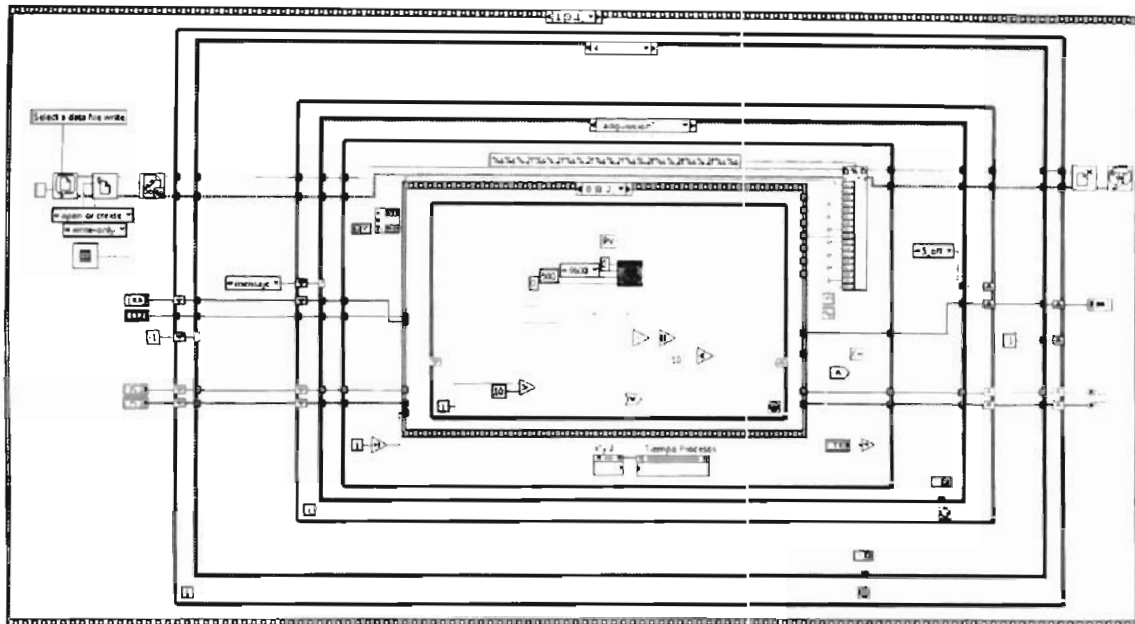


Subprograma modo Bloques.









## BIBLIOGRAFÍA

- LabVIEW 6i Programación Gráfica para el Control de Instrumentación, Antonio Manuel Lázaro, Paraninfo.
- Applying PIC18 Microcontrollers Architecture, Programming, And Interfacing Using C And Assembly, Barry B. Brey, Prentice Hall.
- C/C++ y Java Cómo Programar, Deitel Deitel, Prentice Hall.
- The Art of Electronics, Paul Horowitz, Cambridge University Press.
- Tecnologías epitaxiales de crecimiento de cristales semiconductores, V. A., Mishournyi, I. C Hernández del Castillo, A. Yu. Gorvatchev y A. Lastras Martínez, Avance y Perspectiva volumen 21.
- Fabricación y Caracterización Óptica de Materiales Semiconductores Para Aplicaciones en Optoelectrónica, Hernández Ariza Calderón, Rev. Acad. Colomb. Cienc. Volumen XXVII, Número 104 Septiembre de 2003.
- Crecimiento y caracterización de capas epitaxiales para aplicaciones láser, Ana Aznar Ecija, Departament de Química Física i Inorgànica Universitat Rovira i Virgili.