

UNIVERSIDAD AUTÓNOMA AGRARIA  
“ANTONIO NARRO”

DIVISION DE INGENIERIA

Desarrollo de un sistema de simulación de dosis  
variable de semillas.

Por:

**Joel Patricio José López**

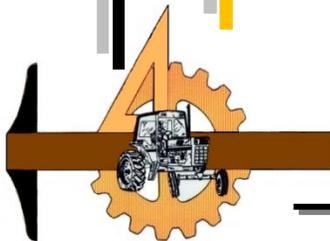
Tesis

Presentada como requisito parcial para obtener el título  
de:

INGENIERO MECÁNICO AGRÍCOLA

Buenavista, Saltillo, Coahuila, México

Mayo de 2010



**UNIVERSIDAD AUTONOMA AGRARIA  
"ANTONIO NARRO"**

**DIVISION DE INGENIERIA**

Desarrollo de un sistema de simulación de dosis variable de semillas.

**Por:**

**Joel Patricio José López**

**Tesis**

Que somete a consideración del H. Jurado Examinador como requisito parcial  
para obtener el título de:

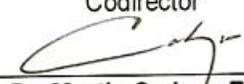
**INGENIERO MECÁNICO AGRÍCOLA**

Aprobado por el Comité de Tesis

Director de Tesis

  
\_\_\_\_\_  
**Dr. Santos G. Campos Magaña**

Codirector

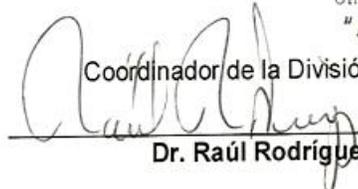
  
\_\_\_\_\_  
**Dr. Martín Cadera Zapata**

Sinodal

  
\_\_\_\_\_  
**Ing. Juan Arredondo Valdez**

Universidad Autónoma Agraria  
"ANTONIO NARRO"

Coordinador de la División de Ingeniería

  
\_\_\_\_\_  
**Dr. Raúl Rodríguez García**

Buenavista, Saltillo, Coahuila, México

  
Coordinación de  
Ingeniería

Mayo de 2010

## AGRADECIMIENTOS

Agradezco a mi **DIOS PADRE** por haberme dado la vida, por darme la oportunidad de seguir aprendiendo, y sobre todo por cuidarme en los momentos más difíciles, por ponerme en el lugar donde debo de estar.

A la UAAAN por haberme abierto sus puertas para poder terminar una carrera, por permitirme estar en sus aulas y ser uno de los pocos afortunados de tener una educación Universitaria... Gracias mi Alma máter

Al **Dr. Santos G. Campos Magaña**, por su gran apoyo y confianza, por el tiempo dedicado en el desarrollo de este proyecto, por su excelente dirección y asesoría técnica en este trabajo de tesis.

Al **Dr. Martín Cadena Zapata** por darnos la oportunidad y confianza de trabajar en el Departamento de Maquinaria Agrícola.

A los diversos maestros del departamento de Maquinaria Agrícola en especial al, **Ing. Juan Arredondo Valdez, M. C. Juan Antonio Guerrero Hernández, M. C. Blanca Elizabeth de la Peña Casas, M. C. Héctor Uriel Serna Fernández, Ing. Tomás Gaytan Muñiz, M. C. Jesús Valenzuela García, Ing. Rosendo González Garza, Ing. Jorge Flores Berrueto** y todos aquellos maestros que aportaron sus conocimientos y experiencias durante el desarrollo de mi carrera.

Al **Ing. Gerardo Sánchez Martínez** por su apoyo y enseñanza que siempre me brindo de manera desinteresada.

A mis **Compañeros de Generación CVIII** de la carrera de Ing. Mecánico Agrícola, en especial a **Oscar Nahúm, Samuel, Ignacio, Jorge Luis, Josafat, Cándido, Obet, Melbin, Michel, Daniel, Víctor, Nelson, Cesar, Deyanira** gracias a todos ellos por su amistad. Y los que no pueden faltar **Luis Alberto y Javier**, por brindarme su amistad sincera y desinteresada.

## DEDICATORIA

Este trabajo fue producto del esfuerzo y sacrificios de muchas personas por tal motivo es para ustedes.

A mis padres:

Sr. Cirilo José  
Sra. Clara López Sánchez

Por todo el apoyo que me han brindado durante toda mi vida, a mi padre por ser un gran ser humano, por sus consejos que me han alentado durante todo este camino, a ti con todo el respeto que te mereces Papá. A mi madre quien siempre me apoyado en las decisiones más difíciles, por brindarme todo ese cariño que no tiene precio, muchas gracias Mamá. Este sueño también es de ustedes.

A mis hermanas

Adriana José López  
Clara Yesenia José López  
Deysi Ruth José López  
Cecilia José López (†)

A todas ellas por brindarme todo su cariño, porque siempre las llevo presente en donde quiera que esté, a Ceci a quien no olvidamos.

A mi esposa e hijo

Elizabeth Camacho Hernández  
Lenin Eduardo José Camacho

Por todo este tiempo de espera, gracias Elí, a Lenin por todo es tiempo que has prescindido de mi ausencia, con mucho cariño y amor mí querido hijo

A Roberto además de ser mi Primo también es mi amigo, gracias por tu amistad.

A mis amigos Arturo e Hilario por todos estos años de amistad.

# INDICE

PAGINA

## RESUMEN

## I INTRODUCCION

1.1 Objetivo General .....	3
1.2 Objetivos Específicos.....	3
1.3 Hipótesis .....	3

## II REVISIÓN DE LITERATURA

2.1 Agricultura de precisión .....	4
2.2 Variabilidad .....	8
2.2.1 Medición de la variabilidad. ....	10
2.2.2 Mapa de rendimiento .....	12
2.3 Dosis variable.....	15
2.4 Microcontroladores .....	17
2.4.1 PIC16F84A .....	17
2.5 LabView .....	18

## III MATERIALES, EQUIPO Y METODOS

3.1 Materiales .....	22
3.2 Metodología.....	26
3.2.1 Familiarización con la elaboración de software. ....	26
3.2.2. Tablero Virtual.....	26
3.2.3. Acoplamiento del IV con los microcontroladores al simulador de semilla.....	28

## IV RESULTADOS

4.1 Programas para desarrollo del software .....	30
4.1.2 Desarrollo de los circuitos .....	33
4.3. Desarrollo del programa en LabView. ....	35
4.4 Funcionamiento de componentes .....	37
4.4.1. Programas para simulación de dosificación. ....	37
4.4.2. Programa para simulación de frecuencia .....	39

4.4.3 Funcionamiento del simulador .....	39
V CONCLUSIONES Y RECOMENDACIONES	
VI. LITERATURA CITADA	
ANEXOS	
Anexo A.....	47
A.1 Generalidades del PIC16F84A .....	47
A.2 Características de la estructura RISC del CPU.....	56
A.3 Características de los periféricos .....	57
A.4 Arquitectura interna .....	57
Anexo B.....	63
B.1 Uso del MPLAB y grabado del microcontrolador .....	63
B.2 Grabado del microcontrolador.....	65
Anexo C .....	71
C.1 Poner en servicio el convertidor MICROMASTER 420.....	71
C.2 Poner en servicio el convertidor MICROMASTER 420.....	71
C.3. Ajustes por defecto.....	73
C.4. Poner en servicio con el panel SDP .....	74
C.5 Poner en servicio con el panel BOP .....	75
C.6. Cambiar parámetros y ajustes usando un panel 'BOP' / 'AOP' .....	77
Anexo D .....	78
D.1. Códigos y diagramas de los programas implementados .....	78
Anexo E.....	86
E.1 Simulación con MPLAB & Proteus 7.0 .....	86

## INDICE DE FIGURAS

Figura 2.1. Interacción entre varios elementos que integran la AP .....	4
Figura 2.2. Las tres etapas de la agricultura de precisión, Fuente: AGCO .....	6
Figura. 2.3. Mapa de preproducción.....	10
Figura. 2.4. Mapas de rendimiento de diferentes años .....	11
Figura.2.5. Variabilidad de la estimación del rendimiento. ....	12
Figura. 2.6. A la izquierda, representación de los datos de posición y rendimiento, sin tratamiento con SIG. A la derecha, la misma parcela tras tratar los datos con un SIG. ....	12
Figura 2.7. Manejo de sitio específico .....	15
Figura 2.8. Diagrama de pines del PIC16F84A .....	18
Figura 2.9. Panel Frontal, aquí se muestra el resultado de la programación	19
Figura 2.10. Diagrama de bloques, aquí se realiza la programación. ....	20
Figura 3.1 Microcontrolador PIC16fF844 .....	22
Figura 3.2. Quemador PICSTART Plus .....	23
Figura 3.3 Tarjeta de adquisición de datos. ....	24
Figura 3.4. Tarjeta de conexiones externas de LabView .....	24
Figura 3.4 Banco de pruebas de PIC16F84A en la tarjeta TES1-3 .....	25
Figura 3.6 Optoelectrónico de barrera H21A1 .....	25
Figura 3.7.Desarrollo del software.....	27
Figura 3.8 Conocer como se usa LabView .....	28
Figura 3.9 Acoplamiento de los componentes .....	29
Figura 4.1. Diagrama de flujo .....	31
Figura 4.2. Tarjeta de Pruebas. ....	32
Figura 4.3 Simulación en PMLAB & PROTEUS .....	32
Figura 4.4 Tarjeta con circuitos implementados .....	34
Figura 4.5 Circuito terminado. ....	34
Figura 4.6. While Loop .....	35
Figura 4.7 DAQ Assistan .....	36
Figura 4.2. Instrumento Virtual .....	36

Figura 4.3 Conexión a la tarjeta de LabView con la tarjeta de pruebas del PIC16F84.....	37
Figura 4.4. Frecuencia en la pantalla de cristal e Instrumento virtual. ....	39
Figura 4.5 Simulador de dosis variable de semilla. ....	40
Figura 4.6. Dosificación en forma manual.....	41
Figura 4.7. Conexiones con el variador de frecuencia. ....	41
Figura A.1 Diagrama de pines del PIC16F84 .....	50
Figura A.2 Los puertos no utilizados se deben conectar a la fuente. ....	52
Figura A.3 Conexión de un oscilador de cristal. ....	54
Figura A.4 Conexión de un oscilador RC. ....	55
Figura A.5. Conexión del botón de reset.....	56
Figura A.6. Arquitectura interna del PIC16F84. ....	58
Figura A.7. Mapa de la memoria de programa. ....	59
Figura A.8. Registros del PIC16F84.....	61
Figura A.9. Contador de programa de (13 bits). ....	62
Figura B.1. Selección del programa “asm” .....	64
Figura. B.2. Procedimiento para el ensamblado.....	64
Figura B.3. USB Adapter Keyspan .....	65
Figura B.5. PICSTART Plus.....	66
Figura B.6. Habilidad del PICSTART Plus. ....	67
Figura B.7 Selección de puertos.....	68
Figura B.8. Para leer el contenido .....	68
Figura B.9. Verificación de memoria .....	69
Figura 2.10. Borrar la memoria del microcontrolador .....	69
Figura 2.11 Conexión del PICSTART Plus .....	70
Figura. C.1 Diagrama para controlar en variador de manera externa. ....	73
Figura C.2. Diagrama para el funcionamiento del variador .....	74
Figura C.3. Esquema del variador de frecuencia.....	76
Figura C.3. Descripción cambiar el parámetro P1082.....	77
Figura D.1 Cuerpo del programa del Pulsador_05B.....	78
Figura D.3. Cuerpo del Programa Pulsador_057.....	79

Figura D.4 Cuerpo del programa Rebotes_Final .....	80
Figura. D.5. Circuito para el Pulsador_Final. ....	81
Figura D.6 Circuito para frecuencias del programa Int_Cuadradas_01 .....	85
Figura E.1. Ensamblar programas.....	86
Figura E.2 Habilitar Proteus para simulación.....	87
Figura E.3. Ejecución del programa ISIS en MPLAB. ....	87
Figura E.4. Habilitar circuito en ISIS.....	88
Figura E.5. Programa circuitos para simulación. ....	88
Figura E.6. Inicio de simulación.....	89
Figura E.7. Terminar simulación.....	89

## INDICE DE CUADROS

Cuadro 2.1 Etapas para la aplicación de agricultura de precisión (Adaptado de USDA, 1998). .....	8
Cuadro 4.2 Pruebas de conteo el LabView y la tarjeta de pruebas del Rebotas_Final. ....	38
Cuadro A.1 Lista de características del PIC16F8x .....	49
Cuadro A.2 Descripción de Pines .....	51
Cuadro C.1. Explicación de conexiones del Diagrama.....	73

## RESUMEN

La agricultura de precisión requiere del uso de las tecnologías que existen para adaptarlas a las necesidades que requiere el campo, ya en otros países como Argentina y EUA se han desarrollado estos trabajos, sin embargo en nuestro país es urgente la necesidad de adaptar y crear tecnología que se requiere en este campo de la agricultura moderna. Esta tesis forma parte del proyecto: **DESARROLLO DE EQUIPOS, SENSORES E INSTRUMENTOS PARA AGRICULTURA DE PRECISION Y LABRANZA DE CONSERVACION.**

Para el desarrollo del proyecto en el presente trabajo se presentan los avances para crear un sistema de simulación que integre la digitalización de dosificación variable de semilla con el control automático de velocidad. Los avances que se obtuvieron en este trabajo fueron muy positivos ya que fue posible crear un sistema que permite ir contabilizando la semilla del dosificador, sin tener ningún rebote, con ayuda de un circuito antirrebotes y la programación de un microcontrolador PIC 16F84A de gama media en el lenguaje ensamblador, lo anterior monitoreado en un Panel de control virtual que fue desarrollado mediante el uso de instrumentos virtuales. Se desarrolló también un programa que nos permite enviar una señal de diferentes frecuencias, que nos permitirá desarrollar la simulación de velocidad de siembra. Así como control de velocidad del motor a través del variador de frecuencia, donde se implementó un circuito que nos permite hacerlo de manera manual. Finalmente se dejó desarrollado un tablero virtual que nos dé la dosificación que permite: contabilizar la semilla a través del dosificador con un 100% de eficiencia, envío de frecuencias, comunicación con el variador de frecuencia para controlar la velocidad del motor.

Palabras clave:

Agricultura de Precisión, Dosificación, Simulación, Control de velocidad, Programación, Instrumento virtual.

## I INTRODUCCION

La Agricultura de precisión o manejo de sitio específico es la utilización de herramientas que permiten la obtención y análisis de datos geo-referenciados, mejorando el diagnóstico, la toma de decisiones y la eficiencia en el uso de insumos así como una disminución sustantiva en la contaminación.

A nivel mundial las oportunidades y riesgos actuales en la producción de alimentos está vinculada con: alta demanda mundial de alimentos (granos, carne, aceite, proteína, bioenergía); alta demanda y costo de petróleo y gas natural en todo el mundo (el gas natural es 90% del costo de la producción de amoníaco); aumento del área fertilizada en el mundo; deficiencias de nutrientes que limitan la producción de cultivos y forrajes así como altos índices de contaminación ambiental entre otros.

Una forma de contrarrestar estos efectos y corregir algunas de sus causas es mediante el manejo óptimo en la aplicación de insumos en la agricultura, para la aplicación de esta tecnología se requiere de una fase de diagnóstico previo de rendimiento, combinado con muestreo de ambientes a nivel de predio para determinar los factores limitantes de la producción así como su localización precisa para poder de ahí realizar las prescripciones en tiempo real y sitio específico de insumos.

Se requiere para lo anterior contar con sistemas que se integren entre otros por AgDGPS (Sistema de Posicionamiento Global Diferencial), Sensores, SIG (Sistema de Información Geográfica), así como equipos de dosificación variables de insumos. A nivel mundial como un indicador existen alrededor de 20 países que han incorporado estos sistemas inteligentes y automatizados en la aplicación de insumos agrícolas, entre los que destacan Estados Unidos con 30 000 unidades de producción, Argentina con 1200, Brasil con 250, Reino Unido con 400, Paraguay con 4 y México con Cero unidades. Actualmente en la UAAAN (Universidad Autónoma Agraria Antonio Narro) no se cuenta con maquinaria ni equipos e instrumentos automatizados para docencia e investigación en mecanización para la agricultura de precisión.

Por tal motivo durante la convocatoria interna 2009, para proyectos de Investigación de la Universidad fue presentado y autorizado un proyecto de investigación denominado:

“DESARROLLO DE EQUIPOS, SENSORES E INSTRUMENTOS PARA AGRICULTURA DE PRECISION Y LABRANZA DE CONSERVACION”. Cuyos objetivos planteados fueron:

1. Fortalecer la enseñanza e investigación en agricultura de precisión,
2. Evaluación de calidad de equipos agrícolas de labranza de conservación.
3. Desarrollo de un Sistema de simulación de dosis variable de semillas.
4. Sistema de Dosificación Variable de Semillas en tiempo real.
5. Desarrollo de equipo para la Generación de Mapas de Conductividad Eléctrica, resistencia al corte y penetración geo-referenciados.
6. Desarrollo de un Sistema Integral para el Monitoreo de Fuerzas en Equipos de Labranza.

“El presente trabajo de tesis se circunscribe en el objetivo número 3; con los siguientes objetivos:”

## 1.1 Objetivo General

1. El objetivo general es el generar un sistema de simulación que integre la digitalización de dosificación variable de semilla con el control automático de velocidad, desplegando en tiempo real la dosificación. Para contribuir a alcanzar este objetivo se plantearon los siguientes objetivos específicos:

## 1.2 Objetivos Específicos

1. Desarrollo de un tablero Virtual que permita monitorear la velocidad de dosificación de semilla.
2. Desarrollo de un programa para la simulación de velocidad de avance.
3. Desarrollo de un programa para evitar revotes en el conteo de semillas.
4. Construcción de un simulador de dosificación.

## 1.3 Hipótesis

Es factible la evaluación de un sistema de dosificación variable mediante el uso de mapas digitales de prescripción basados en mapas de rendimiento y de diagnóstico reales.

## II REVISIÓN DE LITERATURA

### 2.1 Agricultura de precisión

Las características del suelo y del cultivo varían en el espacio (distancia y tiempo) y en el tiempo. La agricultura de precisión es un conjunto de técnicas orientadas a optimizar el uso de los insumos agrícolas (semillas, agroquímicos y correctivos) en función de la cuantificación de la variabilidad espacial y temporal de la producción agrícola. Esta optimización se logra con la distribución de la cantidad correcta de esos insumos, dependiendo del potencial y la necesidad de cada punto de las áreas de manejo.(Figura 2.1)

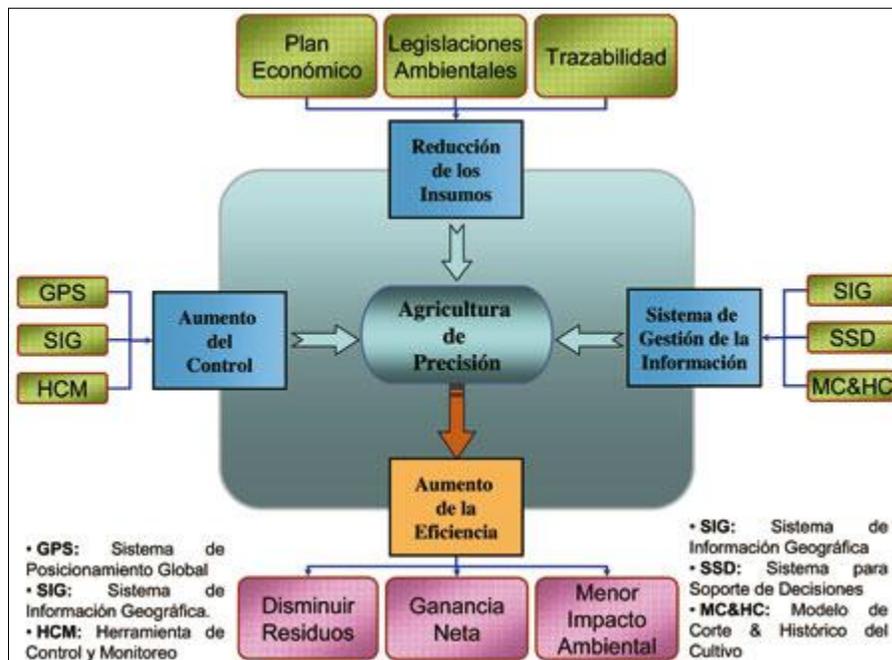


FIGURA 2.1. Interacción entre varios elementos que integran la AP, Fuente: Universidad del País Vasco, 2000

La agricultura de precisión no consiste solamente en medir la variabilidad que existen en el área sino también en la adopción de prácticas administrativas que se realizan en función de esa variabilidad. De acuerdo con Robert (1999), la observación de la existencia de la variabilidad en las propiedades o factores determinantes de la producción en los agroecosistemas no es una novedad. Lo que es diferente, en realidad, es la posibilidad de identificar, cuantificar y mapear esa variabilidad. Más aún, es posible georeferenciar y aplicar los insumos con dosis variables en puntos o áreas de coordenadas geográficas conocidas.

Los agentes involucrados en el desarrollo y adopción de las prácticas de agricultura de precisión suelen dividir este conjunto de tecnologías en tres etapas diferentes AGCO 2005 (Allis Corporación Gleaner), (Figura 2.2): 1) Recolección de datos 2) Procesamiento e interpretación de la información 3) Aplicación de insumos

En base de ello, se define un ciclo de prácticas agrícolas orientado a sustituir la recomendación habitual de insumos en base a valores promedio, como ocurre en la agricultura tradicional, por una más precisa, con manejo localizado, considerando las variaciones tradicionales en toda el área. Es una optimización del uso de los insumos porque deposita en el suelo la cantidad de semilla que cada punto soporta, la cantidad de nutrientes requerida, y además el control de malezas, plagas y enfermedades se hace solamente en los puntos que demanden tal control.

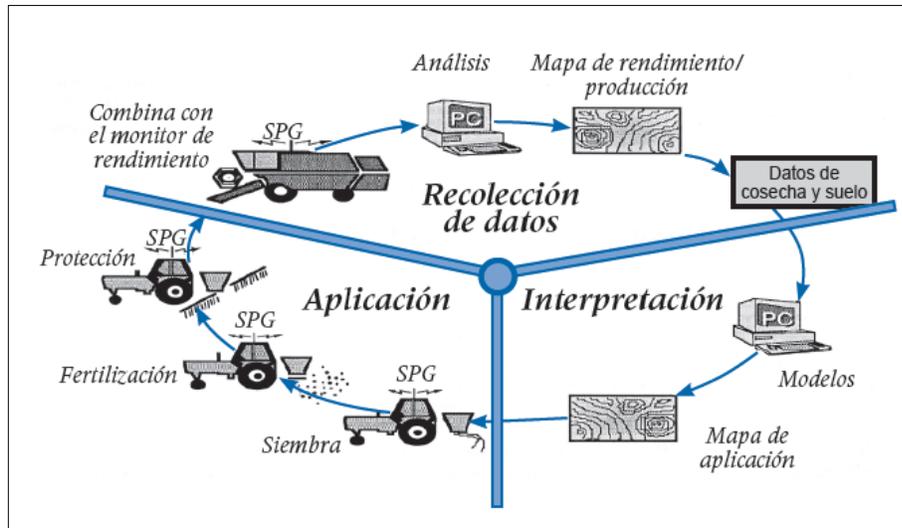


Figura 2.2. Las tres etapas de la agricultura de precisión, Fuente: AGCO

La agricultura de precisión podría definirse como “...un conjunto de prácticas agrícolas utilizadas para identificar y registrar la variabilidad de ciertos parámetros agronómicos dentro de un mismo lote. Sobre la base de dicha información se aplican dosis variables de insumos posicionándolos en el lote acorde a la necesidad de un nivel de productividad prefijado”. Definición propuesta por Maroni y Gargicevich (1996).

La agricultura de precisión (AP) es una estrategia de manejo que utiliza la tecnología de la información para captar datos de múltiples fuentes para generar decisiones asociadas a la producción de cultivos (US National Research Council, 1997).

Otra definición de la AP es un sistema de producción que se basa en la integración de la información y la producción. Está diseñado para sitios específicos y la unidad completa de producción, a objeto de incrementar en el largo plazo, la eficiencia, productividad y margen de ganancia, minimizando los indeseados impactos sobre el ambiente (US Farm Bill, 1996 ).

La Agricultura de Precisión es una aproximación a un sistema para manejar cultivos y suelos en forma selectiva de acuerdo con sus necesidades. Utiliza la experiencia de muchas disciplinas e integra las últimas herramientas y tecnológicas de información que permiten a los administradores de fincas tener una mejor

*comprensión y control de sus lotes. La administración es el factor esencial para alcanzar el resultado indicado en la finca, por lo cual los administradores deberán identificar sus propias estrategias y prácticas que permitan un manejo efectivo de la variabilidad encontrada en la finca.*

*Tres tipos de variabilidad han sido identificados. El primer tipo es la variabilidad espacial, la cual puede ser vista como los cambios a través del lote. Un ejemplo, puede ser cuando un sector del lote genera rendimientos superiores frente a otro sector del mismo lote, inclusive frente a uno vecino. La segunda variabilidad es la temporal que se refiere a cuando los parámetros cambian a través del tiempo. Puede verse cuando un cultivo tiene una buena fase de desarrollo pero posteriormente arroja bajos rendimientos. El tercer tipo es la variabilidad predictiva que explica la diferencia entre lo que el productor predice que a suceder y lo que realmente ocurre.*

*El ejemplo clásico de esta variabilidad predictiva es cuando el productor predice que una cierta productividad puede ser lograda si determinada cantidad de fertilizante es aplicada, pero el cultivo no alcanza esto por causa del cambio climático u otro factor. Cada tipo de variabilidad puede ser medida, evaluada y posiblemente influenciada, de acuerdo con cuan significativa es.*

En el Cuadro 2.1 se presentan las principales etapas o pasos para la aplicación de AP (Agricultura de Precisión), las tecnologías involucradas y las actividades realizadas.

En AP existen dos aproximaciones para la aplicación variable de insumos. La primera de ellas se basa en el muestreo y mapeo de los factores de producción a ser manejados en forma diferencial (fertilidad del suelo, malezas, etc.) y la posterior elaboración de mapas de prescripción para la aplicación variable de los insumos (fertilizantes, herbicidas, etc.). La segunda aproximación es el sensoriamiento directo del suelo y/o el cultivo para la aplicación inmediata de los insumos en forma variable. El uso de una u otra dependerá del nivel tecnológico disponible y del costo de operación involucrado.

Cuadro 2.1 Etapas para la aplicación de agricultura de precisión (Adaptado de USDA, 1998).

ETAPA	TECNOLOGÍA INVOLUCRADA	ACTIVIDADES
Recolección e ingreso de datos.	<ul style="list-style-type: none"> <li>• Sistemas de posicionamiento global (GPS).</li> <li>• Sistemas de información geográfica (SIG).</li> <li>• Instrumentos topográficos.</li> <li>• Sensores remotos.</li> <li>• Sensores directos.</li> </ul>	<ul style="list-style-type: none"> <li>• Medición de la topografía del suelo.</li> <li>• Muestreo de suelos en grilla.</li> <li>• Recorrido de los cultivos para la detección de plagas y enfermedades.</li> <li>• Monitoreo de rendimientos.</li> <li>• Medición directa de propiedades del suelo y cultivos.</li> <li>• Sensoriamiento remoto de suelos y cultivos.</li> <li>• Digitalización de mapas.</li> </ul>
Análisis, procesamiento e interpretación de la información.	<ul style="list-style-type: none"> <li>• Programas de SIG.</li> <li>• Sistemas expertos.</li> <li>• Programas estadísticos.</li> <li>• Experiencia del operador.</li> </ul>	<ul style="list-style-type: none"> <li>• Análisis de dependencia espacial.</li> <li>• Confección de mapas de evaluación.</li> <li>• Confección de mapas de prescripción.</li> <li>• Otras.</li> </ul>
Aplicación diferencial de insumos.	<ul style="list-style-type: none"> <li>• Tecnología de dosis variables.</li> <li>• Pulverización asistida por GPS.</li> <li>• Programas computacionales.</li> </ul>	<ul style="list-style-type: none"> <li>• Aplicación variable de nutrientes.</li> <li>• Aplicación variable de plaguicidas.</li> <li>• Siembra diferencial de variedades y aplicación variable de semillas.</li> <li>• Otras.</li> </ul>

## 2.2 Variabilidad

Los mapas de rendimiento permiten cuantificar la variabilidad de rendimiento existente durante la cosecha de un cultivo dentro del lote, quedando grabada espacialmente.

La variabilidad de los lotes es uno de los factores que pueden justificar la realización de dosis variable en la siembra, fertilización, pulverización, etc y gracias al GPS y algunas herramientas de la Agricultura de Precisión, entre ellas el monitor de rendimiento, podemos conocer el área que ocupa cada sitio y cuantificar sus diferencias de rendimiento.

Comprender la variabilidad es una de las claves de la Agricultura de Precisión. Se han identificado tres formas principales de variabilidad a tener en cuenta: Espacial, temporal y en la estimación.

- ✓ La variabilidad espacial, es la que se da a lo largo y ancho del terreno. Esta variabilidad se puede reflejar mediante mapas de suelo. Si esta variabilidad es significativa, es el agricultor el que debe tomar las decisiones, aunque lo normal es contrastar este mapa con los de rendimiento.
- ✓ La variabilidad temporal se pone de manifiesto al comparar mapas de rendimientos de un año a otro, y su significado debe también ser deducido, aunque se pueden obtener mapas de tendencias que muestren las características fundamentales.
- ✓ La variabilidad en la estimación, es la diferencia de lo estimado, con respecto a lo obtenido. Para un correcto aporte de inputs, se deben hacer ciertas aseveraciones, para obtener el rendimiento estimado. Si ese rendimiento no se obtiene, las aseveraciones que se hicieron en un primer momento, deberán ser mejoradas.

Antes de entrar a tomar decisiones respecto a estas variabilidades, se deben comprender, y por lo tanto previamente medirlas.

### 2.2.1 Medición de la variabilidad.

La variabilidad espacial: Esta se puede medir mediante la obtención de datos sobre factores con localización precisa. Los mapas de rendimiento se producen realizando un seguimiento del rendimiento a la cosechadora, o maquina recolectora, tal que se conoce la cantidad cosechada en un determinado tiempo. Un DGPS (GPS Diferencial) se utiliza para obtener la localización de la posición de los rendimientos que se están obteniendo. Un sistema similar se utiliza para el muestreo del suelo, excepto si las muestras se recogen a mano, y se envían posteriormente a laboratorio para su análisis, en este último caso se utiliza un GPS para el posicionamiento del lugar de toma de muestras (Figura 2.3).

Estos datos se filtran y se representan como mapas que muestran la variabilidad espacial.

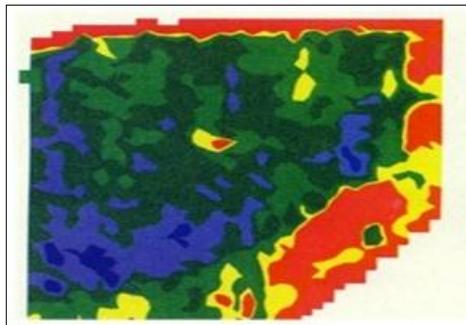


Figura. 2.3. Mapa de producción.

La variabilidad temporal: Se obtiene mediante la observación y comparación de mapas que han sido obtenidos a lo largo del tiempo. (Figura 2.4).

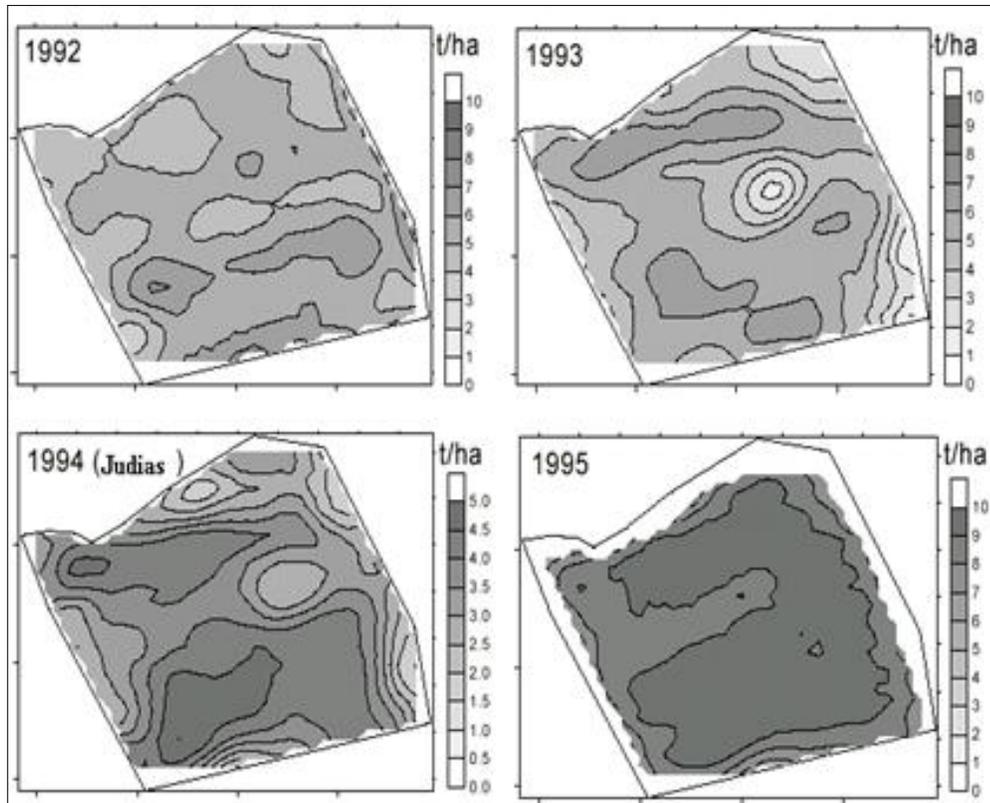


Figura. 2.4. Mapas de rendimiento de diferentes años

La variabilidad en la estimación: Esta variabilidad se obtiene midiendo la diferencia entre los valores para el cultivo, y los valores conseguidos. Los valores que mas se utilizan son los de área de producción, el rendimiento y calidad. La calidad es difícil de predecir, hasta los estadios tardíos de la producción, mediante la medida de la clorofila, y la realización de un mapa de clorofila. (Figura 2.5).

La mayoría de los trabajos publicados relacionados a sistemas de determinación de producción de productos agrícolas para la elaboración de mapas de rendimiento están en el área de granos, donde empresas como JOHN DEERE (1998), CASE (1999) y AGCO (1999) desarrollaron sensores de flujo de masa específicos para determinar la producción y la humedad de granos y software para manejar estos datos y generar los mapas de rendimiento.

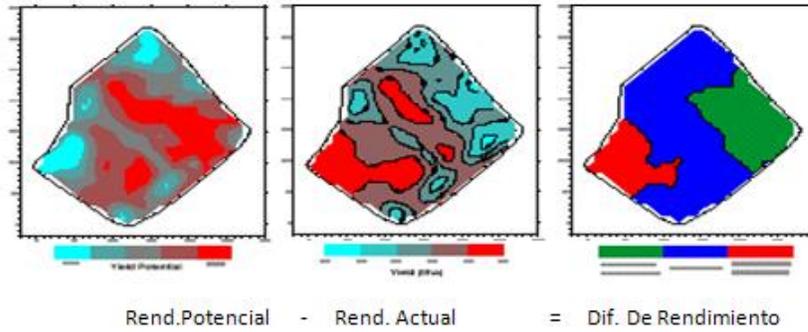


Figura.2.5. Variabilidad de la estimación del rendimiento.

### 2.2.2 Mapa de rendimiento

La práctica más importante, quizá, son los mapas de rendimiento. Para ello son necesarias una serie de tecnologías, tomemos como ejemplo una cosechadora de cereal:

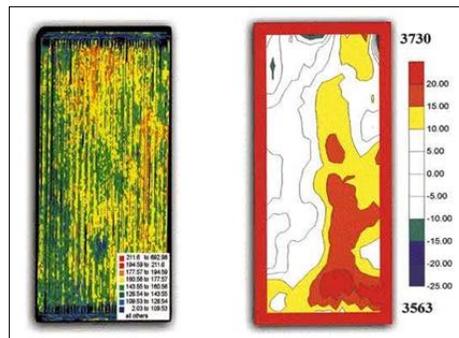


Figura. 2.6. A la izquierda, representación de los datos de posición y rendimiento, sin tratamiento con SIG. A la derecha, la misma parcela tras tratar los datos con un SIG.

Será necesario un receptor GPS montado sobre la cosechadora, el cual obtendrá segundo a segundo la información de la posición en el campo, a medida que la cosechadora avanza por el campo. Además, esta cosechadora debe tener 2 sensores, uno para medir el flujo de grano, y otro para medir la humedad del grano cosechado. A medida que la maquina

avanza, un ordenador recoge la información que le llega del GPS, con un intervalo de 1.2 segundos registrando la latitud y la longitud, y de los dos sensores.

Normalmente la información se almacena en memoria PCMCIA, la cual puede extraerse de este ordenador, y pasar su información posteriormente al ordenador de la oficina.

Si se realizase una representación de esta información, se obtendría la parte izquierda de la figura 2.6, la cual como se ve es difícil de interpretar. Por lo tanto es necesario un software que permita una mejor lectura o interpretación de los datos, estos programas informáticos son los SIG (Sistemas de Información Geográfica). Así la figura antes mencionada, aparecerá suavizada, como en la parte derecha de la figura 2.6.

ELIAS (1997) que relato el desarrollo, construcción y uso a campo, de un sistema de instrumental y adquisición de datos para el mapeo de producción en granos. El sistema consistía en la medición automática de peso de los granos cosechados utilizando una sub-tolva granelera apoyada sobre cuatro células de carga dentro de la tolva de la cosechadora, sin ninguna modificación estructural de la misma, obteniendo un error máximo de 3,99 N para una capacidad de la sub-tolva de 4.905 N. La principal ventaja del sistema de pesaje automático, desarrollado por ELIAS, está en permitir la obtención directa del peso de los granos cosechados, sin necesidad de sensores para la medición del flujo de granos y calibración para cada cultivo.

BALASTREIRE (1998) que presenta el potencial de utilización de los conceptos de AP en el cultivo de la caña de azúcar. En su artículo el autor resalta entre los principales beneficios que la aplicación de este concepto puede ofrecer, como: a) posibilidad de reducción de polución ambiental; b) beneficios económicos, donde la reducción del costo final del producto puede ser obtenida reduciendo los insumos en los puntos de bajo potencial de

producción y redireccionandolo a los puntos de mayor potencial; c) ganancia de productividad, con aumento de producción y un aumento de la cantidad de azúcar de caña disponible. Además según BALASTREIRE el potencial de aplicar los conceptos de AP en el cultivo de la caña de azúcar se divide en cuatro puntos:

Mapa de suelo – este mostrará el potencial de fertilidad del suelo, y permitirá establecer las estrategias de aplicación de fertilizantes.

Mapa de cultivo – permitirá obtener mapas georeferenciados de los tablones del cultivo en sus diferentes estadios.

Mapa de cosecha – se puede obtener un mapa de producción de cada celda dentro de un tablón. Con la Georeferenciación se tiene la localización exacta de cada celda, entonces, se puede verificar en el lugar cual fue el motivo de la baja de producción.

SCHNEIDER *et al.*, (1996) desarrollaron un sistema de monitoreo de producción de papa que consistía en un acondicionador de señales y una unidad de control (SCCU), sensor para medir la velocidad de la correa transportadora de las celdas de carga, DGPS, computadora a bordo y un modem con frecuencia de radio (RF). El sensor de masa (celda de carga) fue montado en una barra con una escala de fondo de 225 kg. El sensor de velocidad de la correa transportadora fue fijado al lado del eje en el cual fue montado un magneto. Los tiempos registrados en un archivo fueron ajustados para corregir la diferencia de tiempo entre el momento de cosecha y el momento de pesaje, "lag time". Fue usado un algoritmo para determinar la media de las medidas de peso dentro de una distancia fija de 15 m. En este método es preferible usar un número de fijo de puntos, considerando la posibilidad de inicio y fin de las operaciones de la máquina.



En el caso que haya variabilidad de rendimientos en el campo, es posible conocer la superficie que ocupa el sitio con mayor o menor potencialidad de rendimiento y se puede caracterizar por medio de análisis físicos o químicos de suelo. Luego comienza el proceso de saber que dosis de fertilización remendar, por ejemplo, y existen varias maneras de realizar recomendaciones. Una de ellas, es el método de balance (que es simple pero más preciso si varían mucho las condiciones de los ambientes), o haciendo correr modelos de simulación para cada sitio (que le da más precisión al dato). Luego, se aplica la cantidad de fertilizante que los modelos indiquen con la maquinaria de dosis variable y en cada sitio específico. El objetivo es apuntar a un manejo más eficiente de los insumos, dejando de lado la dosis uniforme. Por ejemplo, donde hay recomendaciones zonales de aplicar 100 kg de nitrógeno/ha para el cultivo de maíz, ahora se puede determinar en qué parte del lote se fertiliza con 30 o 150 kg de N/ha y, de esta manera, se maneja la variabilidad del campo, “muchas personas hacen dosis variable aplicando menos fertilizante en algún lugar y más en otro sin tener datos precisos sobre el potencial de rendimiento del cultivo u lo que puede aportar el suelo al cultivo. Es decir que realizan dosis variable de manera imprecisa”.

Los pasos a seguir para llegar a buenos resultados en dosis variable de semilla y fertilizantes. Contar con información georreferenciada es un paso importante en la aplicación variable de insumos, nos permite determinar la variabilidad presente en el lote. Para ello, existen diferentes herramientas que nos pueden llevar a una mejor definición de los ambientes homogéneos presentes en el lote como ser: carta de suelos, fotografías aéreas, imágenes satelitales, entre otras. Una de las herramientas más destacadas es el mapa de rendimiento, debido a que el mismo, nos permite saber, no solo la variabilidad presente en el lote, sino también su cuantificación. En forma metafórica, es como una radiografía del lote. Esta información tan valiosa se

puede conseguir mediante un monitor de rendimientos montado en una cosechadora y el mismo conectado a un GPS

## 2.4 Microcontroladores

Un microcontrolador es un circuito integrado que contiene una Unidad Central de Proceso (CPU) y una serie de recursos internos en un solo encapsulado. El CPU permite que el microcontrolador pueda ejecutar instrucciones almacenadas en una memoria. Los recursos internos son memoria RAM, memoria ROM, memoria EEPROM, puerto serie, puertos de entrada/salida, temporizadores, comparadores, etc. Se puede decir que es una evolución del microprocesador, al añadirle a este último las funciones que antes era necesario situar externamente con otros circuitos. El ejemplo típico está en los puertos de entrada/salida y en la memoria RAM, en los sistemas con microprocesadores es necesario desarrollar una lógica de control y unos circuitos para implementar las funciones anteriores, con un microcontrolador no hace falta porque lo lleva todo incorporado, además en el caso de tener que ampliar el sistema ya ofrece recursos que facilitan esto. En resumen, un microcontrolador es un circuito integrado independiente, que no necesita memoria ni puertos externos pues los lleva en su interior, que facilita la tarea de diseño y reduce el espacio, traduciéndose todo a una aplicación final más económica y fiable.

### 2.4.1 PIC16F84A

Es el PIC con memoria Flash más popular. Cuenta con una Memoria de Programación serial de 1024 instrucciones y 64 localidades de memoria RAM. La memoria de programación es eléctricamente borrable ya que no se requiere borrarlo con luz ultravioleta como las versiones EPROM.

Internamente cuenta con un contador de tiempo. Este PIC se presenta en un Chip de dieciocho pines de los cuales 13 están disponibles como Entradas / salidas. Figura 2.8

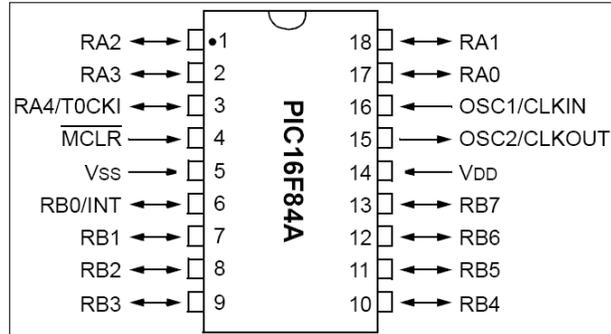


Figura 2.8. Diagrama de pines del PIC16F84A

El PIC16F84 (Anexo A) es un microcontrolador de Microchip Technology, su consumo de potencia es muy bajo y además es completamente estático, esto quiere decir que el reloj puede detenerse y los datos de la memoria no se pierden.

## 2.5 LabView

De acuerdo con la página de internet: <http://digital.ni.com/worldwide/spain.nsf/web/all/AF4C3DCD8ED8042C80256C68004330AB?OpenDocument>, LabView es un revolucionario entorno de desarrollo gráfico con funciones integradas para realizar adquisición de datos, control de instrumentos, análisis de medida y presentaciones de datos. LabView le da la flexibilidad de un potente ambiente de programación, pero mucho más sencillo que los entornos tradicionales.

Lenguaje Desarrollado para Medida, Control y Automatización a diferencia de los lenguajes de propósito general, LabView tiene funciones específicas para acelerar el desarrollo de aplicaciones de medida, control y automatización.

Los programas desarrollados en LabView llevan por nombre instrumento virtual que es un módulo software que simula el panel frontal de instrumentación que antes hemos comentado y, apoyándose en elementos hardware accesibles por el computador (tarjetas de adquisición, tarjetas DSP, instrumentos accesibles vía GPIB, RS-232, USB, ethernet).

Cuando trabajamos en el ambiente LabView trabajamos con dos ventanas: una en la que implementara el panel frontal (figura 2.9) y otra que soporta el nivel de programación (figura 2.10). Para la creación del panel frontal se dispone de una librería de controles e indicadores de todo tipo y la posibilidad de crear más, diseñados por el propio usuario.

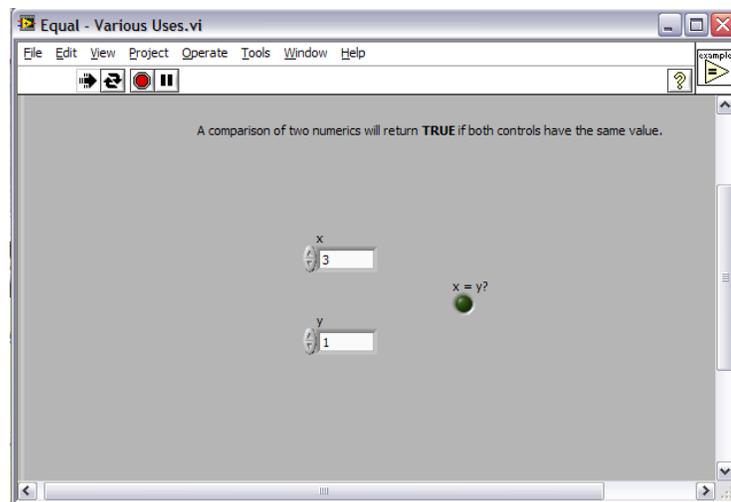


Figura 2.9. Panel Frontal, aquí se muestra el resultado de la programación

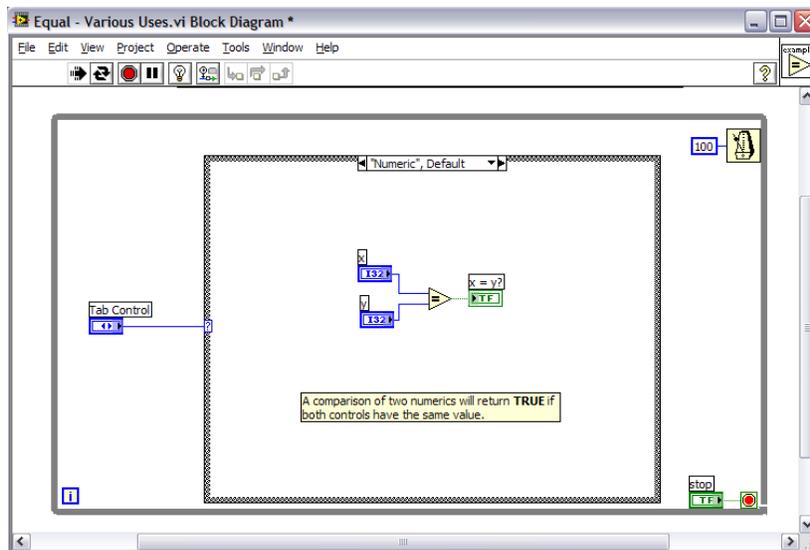


Figura 2.10. Diagrama de bloques, aquí se realiza la programación.

Anzures (2001) Diseño de un medidor de energía virtual, empleando LabView, el cual indica, de manera digital, en una computadora personal (PC), el consumo de ésta a nivel residencial. El instrumento virtual, sensa las señales de voltaje y corriente del sistema, para ser llevadas a la PC a través de una tarjeta de adquisición de datos. Además, indica el voltaje y corriente en ms; potencias activa, reactiva y aparente; demanda máxima; factor de potencia y genera un espectro de frecuencia de las señales. Actualmente, permite capturar eventos en sistemas monofásicos en intervalos de tiempo definidos por el usuario; por defecto el instrumento maneja 5 y 15 min de acuerdo a los estándares de Comisión Federal para sus medidores de estado sólido.

Burgos (2003) desarrollo un control de un horno microondas casero convencional que fue adaptado para ser usado como medio de calentamiento en un proceso de destilación de agua. En la automatización del proceso se uso LabView y una tarjeta de adquisición de datos de National Instruments con la que se tomó registro de la temperatura dentro del horno y se enviaba la señal de encendido y apagado del horno. Para el control de la

temperatura dentro del horno microondas se uso una termocupla, la cual era la responsable de realizar la realimentación del sistema para informar al programa y mantener el nivel de temperatura dentro de la cavidad en los niveles deseados para la destilación.

### III MATERIALES, EQUIPO Y METODOS

#### 3.1 Materiales

Para la realización de este trabajo fue necesario contar con distintos equipos y accesorios electrónicos los cuales se describen a continuación.

1. Un microcontrolador PIC16F84, memoria tipo flash lo que permite que sea reprogramado, compuesto de 18 pines, 13 pines de E/S 5 del puerto A y 8 del puerto B, utiliza un oscilador de cristal de 4MHz, Cuenta con una Memoria de Programación serial de 1024 registros, 128 posiciones de memoria RAM de 8 bits cada una y 64 posiciones de memoria EEPROM de 8 bits cada una, Figura 3.1.
2. Componentes electrónicos, y herramienta utilizados en la construcción del hardware, para el programa desarrollado.



Figura 3.1 Microcontrolador PIC16f844

3. Software MPLAB IDE Versión 8.1; como el editor del lenguaje ensamblador donde se desarrollaron y probaron los programas hasta obtener resultados positivos.



Figura. 3.2. Quemador PICSTART Plus

4. Un programador o quemador PICSTART Plus (Development programmer) de Microchip Technology, conectado al puerto serial de la computadora y a una fuente de alimentación de +9 V, utilizado para grabar el programa en la memoria del chip. Figura 3.2.
5. Software Proteus 7 Profesional, el cual aunado con el MPLAB, son utilizados para la simulación de los circuitos, antes de ser probados de manera física.
6. Software LabView 8.5 de National Instrument, el cual se desarrollo el programa que nos permite tener comunicación con la tarjeta de adquisición de datos.

7. Tarjeta de adquisición de datos PCI-NI-6221 de E/S para uso con dispositivos DAQ de las Series X, M y E (sólo NI-DAQmx), Figura 3.3



Figura 3.3 Tarjeta de adquisición de datos.

8. Tarjeta de conexión de señales necesaria para la comunicación externa.



Figura 3.4. Tarjeta de conexiones externas de LabView

9. Una computadora personal (Thosiba, con procesador centrino, con un procesador de memoria RAM de 1GB, con capacidad de disco duro de 120 Gb).

10. Una Tarjeta de pruebas de microcontroladores para probar los diferentes programas, que se fueron implementando para alcanzar el objetivo deseado. Figura 3.4.

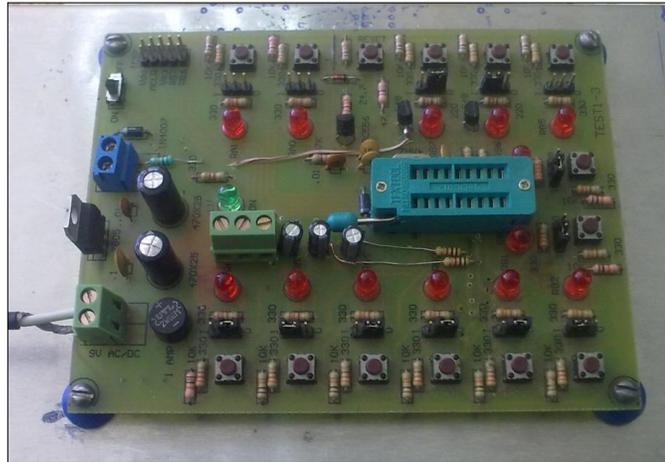


Figura 3.4 Tarjeta de pruebas modelo TES1-3 para PIC16F84A .

11. Un sensor optoelectrónico para medir pulsos, que detecta la señal analógica y la envía al microcontrolador

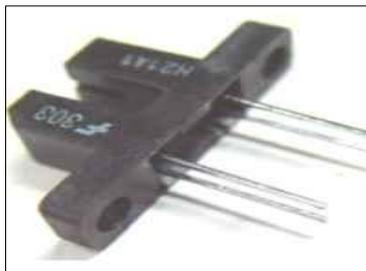


Figura 3.6 Optoelectrónico de barrera H21A1

12. Un motor trifásico de 0.25 kW, con un disco intercambiable de 20 0 mas perforaciones. Para la simulación de dosificación

## 3.2 Metodología

Para lograr los objetivos planteados, el método a seguir fue una familiarización con la programación de microcontroladores así como la programación en el ambiente de Instrumentos Virtuales, para posteriormente la elaboración de un programa relacionados con la simulación de velocidad; simulación de dosificación de semilla y la construcción del simulador en tiempo real, a continuación se describe el método para cada etapa.

### 3.2.1 Familiarización con la elaboración de software.

Para la programación en micro controladores se empleo lenguaje ensamblador con el programa MPLAB el cual es provisto por la empresa Microchip; así como el proceso de simulación usando el Proteus 7.0 en el ambiente de ISIS, una vez verificado su funcionamiento se realiza la grabación el microcontrolador (anexo B), para su posterior verificación en la tarjeta de pruebas, en la figura 3.7 se ilustra la metodología empleada.

### 3.2.2. Tablero Virtual

En esta parte se verán los resultados programación de los micro controladores, para poder llegar al desarrollo de instrumento virtual (IV), fue necesario realizar diferentes ejecuciones de programas de instrumentación virtual los cuales fueron hechos en LabView 8.5 de la empresa de National Instruments.

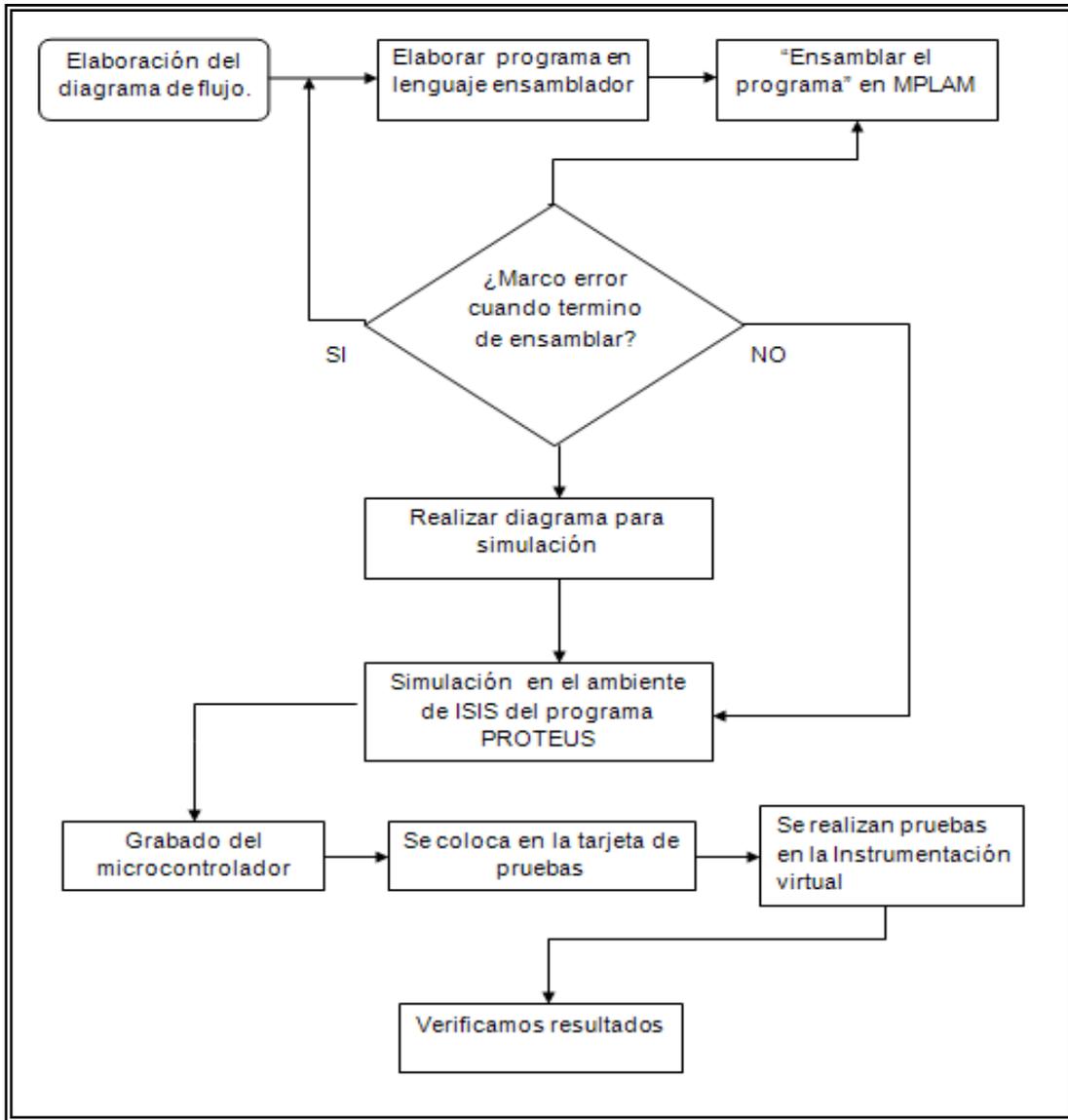


Figura 3.7.Desarrollo del software

Se realizaron diferentes programas desde los más elementales de sumas y resta para ambientarse en el lenguaje de programación a base de objetos, así como conocer la estructura de programación. En la figura 3.8 describimos la forma de conocer el ambiente LabView.

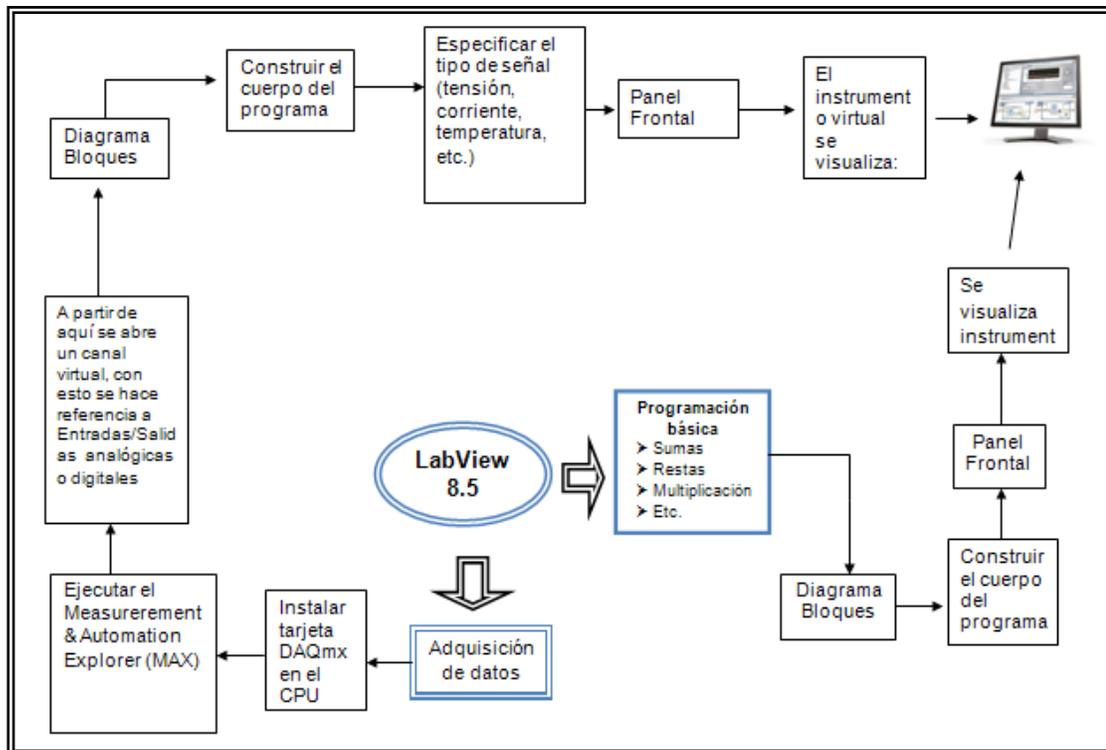


Figura 3.8 Conocer como se usa LabView

Siguiendo la metodología descrita se diseño un instrumento virtual el cual contiene:

- Registro de frecuencia
- Registro de la cantidad de semillas
- Control de la velocidad del motor a través del variador de frecuencia.

3.2.3. Acoplamiento del IV con los microcontroladores al simulador de semilla.

Una vez que se tengan desarrollados el software para el microcontrolador, y se hayan simulado de manera correcta los circuitos en Proteus, procedemos a la verificación de el funcionamiento a través la interacción con el programa desarrollado en LabView (instrumentación virtual).en la figura 3.9 se muestra el procedimiento de verificación.

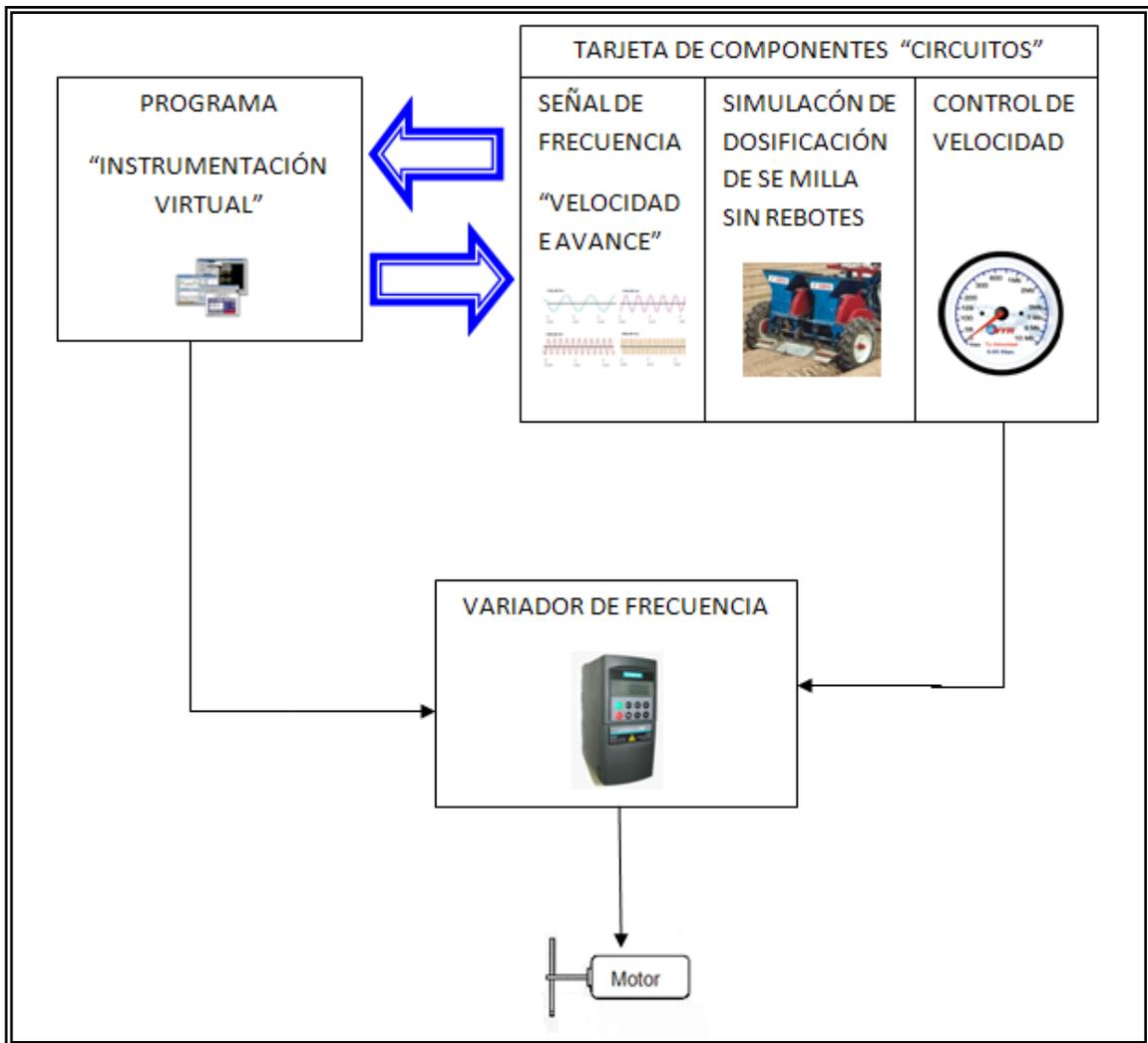


Figura 3.9 Acoplamiento de los componentes

Una vez realizado el procedimiento descrito en la figura 3.10 se verifico el funcionamiento de cada uno de los componentes, y fue realizado en el siguiente orden:

- Envío de señal del simulador de dosificación sin rebotes
- Verificar el envío de la frecuencia.
- Envío de señal al variador

## IV RESULTADOS

En esta sección se describen los resultados obtenidos de los programas desarrollados en el lenguaje ensamblador, del circuito y su funcionamiento cuando se colocan los microcontroladores ya grabados, el montaje de los componentes (hardware), la conexión del circuito con la tarjeta de adquisición de datos y su funcionamiento con el Instrumento Virtual (VI).

### 4.1 Programas para desarrollo del software

De acuerdo con el diagrama de flujo que se muestra en la figura 4.1 fue necesario realizar varias pruebas de simulación de los programas que fueron desarrollados en lenguaje ensamblador.

Con la ayuda del MPLAB se transcribió el cuerpo del programa del software, programando los pines del puerto A como salidas y los pines del puerto B como entradas. Debido a que el ensamblador exige cierta tabulación de sus elementos, para la escritura del programa pusimos en la primera columna la definición de variables, mientras que las instrucciones las colocamos en la segunda y tercera columna y los comentarios anteceditos de un (;) en la cuarta columna, el uso de mayúsculas y minúsculas generalmente no son obligatorias pero obedecen a una serie de reglas o normas de estilo que facilita la lectura del código fuente, se procedió al ensamblado para verificar los errores y corregirlos, en caso de que los hubiera.

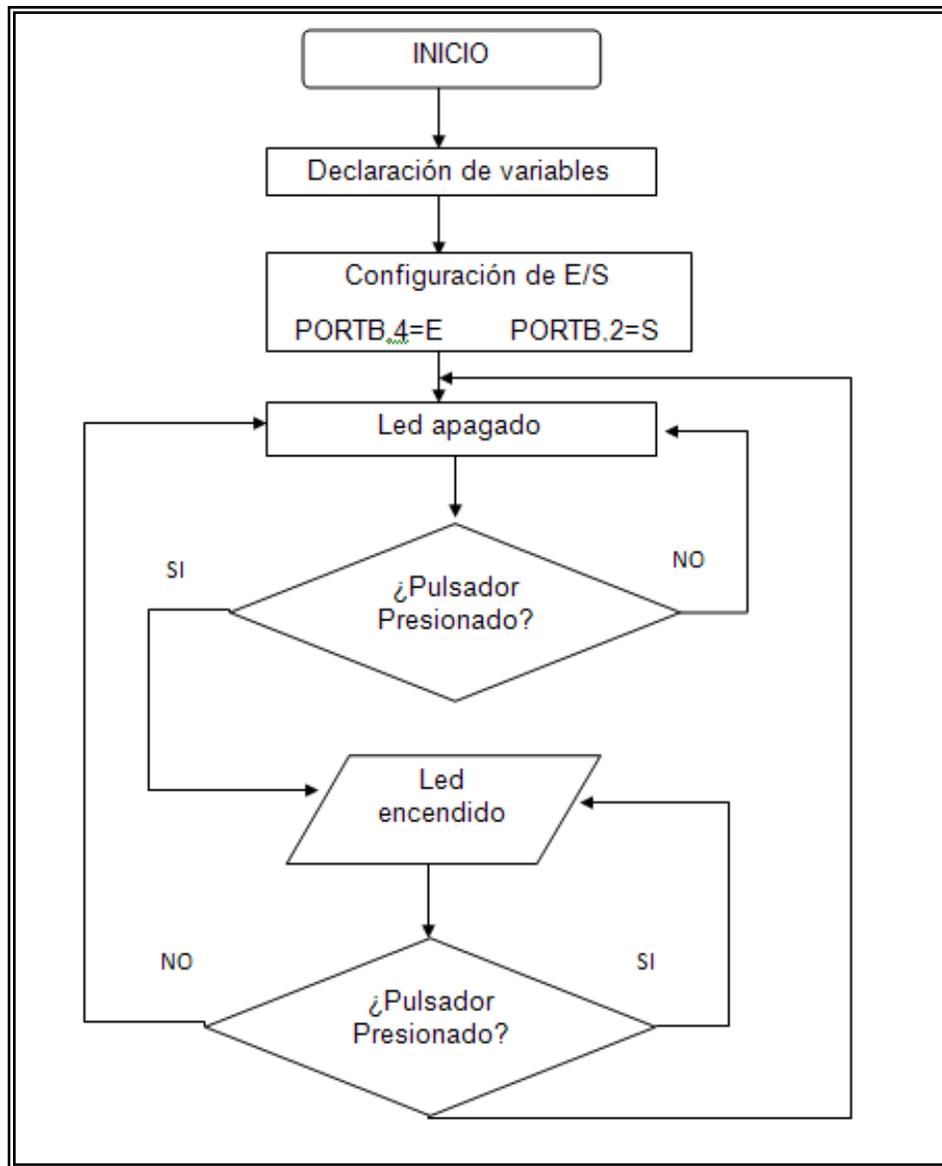


Figura. 4.1. Diagrama de flujo

Cuando hemos verificado que no hay errores, usando el PICSTART Plus procedimos al quemado o grabado del programa en la memoria del microcontrolador (Anexo II), y se monto en una tarjeta de prueba para ver su funcionamiento, Figura 4.2, después configuramos el puerto A como entrada y B como salida, se quemó y se probó nuevamente el programa, con lo que se comprobó el uso de los puertos como entradas ó como salidas digitales.

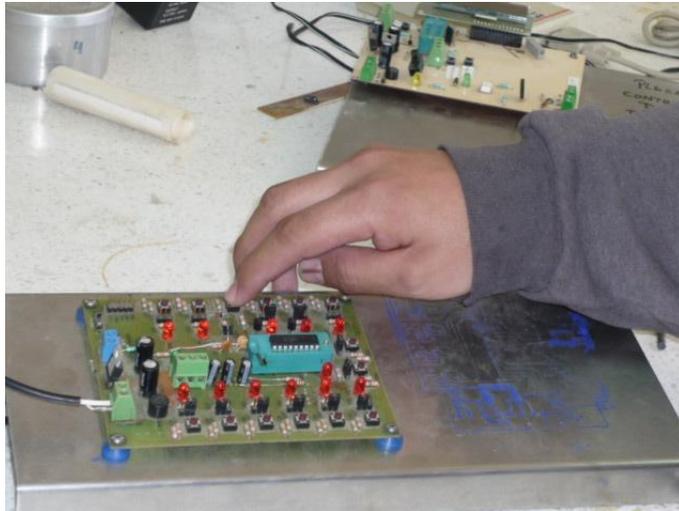


Figura 4.2. Tarjeta de Pruebas.

Una vez que verificamos el correcto funcionamiento del programa realizamos una simulación en el MPLAB, llamando a ejecutar el ISIS Profesional 7.0 (anexo E), que es el que nos permite cargar el circuito, de esta manera verificar su funcionamiento antes del desarrollo del equipo (software y hardware), Figura 4.3 se muestra la simulación.

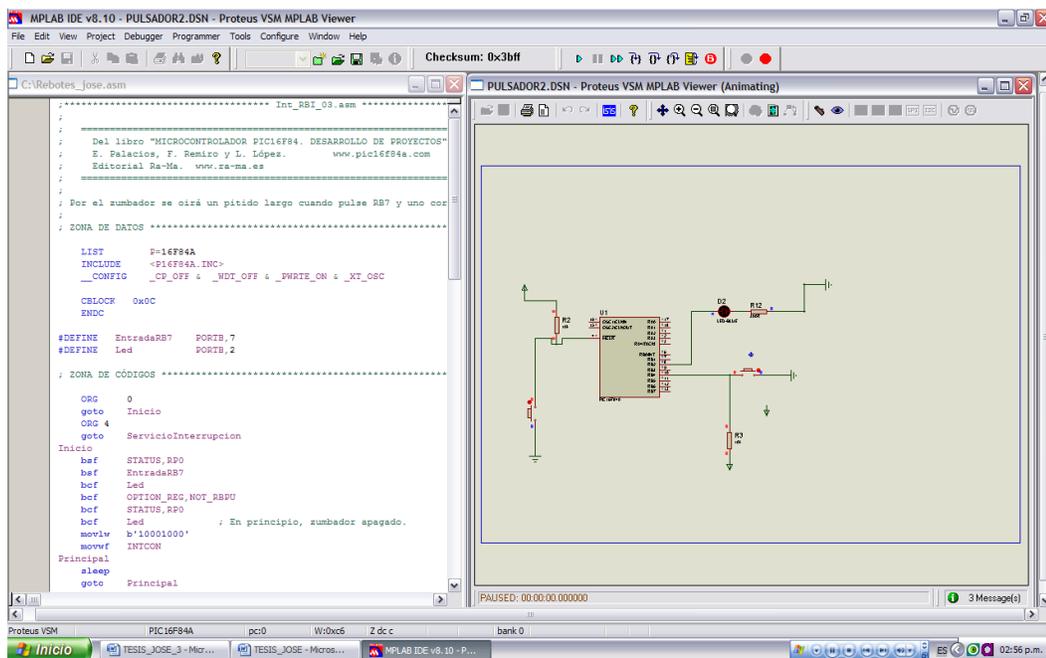


Figura 4.3 Simulación en PMLAB & PROTEUS

El anexo D, se muestran los proyectos implementados así como sus diagramas respectivos; los cuales fueron necesarios para finalmente obtener un pulsador que realizara las funciones requeridas para su implementación como un contador, el cual recibe la señal del optoelectrónico.

#### **4.1.2 Desarrollo de los circuitos**

El desarrollo de los circuitos fue diseñado en Proteus 7.0 Professional, en el ambiente de Ares 7 Professional. Figura 4.4

1. Se desarrollo un circuito que nos permita enviar una frecuencia.
2. Fue necesario implementar un circuito, el cual nos enciende y apaga un led, para lo cual se hicieron varios intentos tanto de circuitos así como programas de los microcontroladores.
3. El desarrollo de la acondicionador de señal la cual se realiza con una compuerta logia HD74LS02P
4. Así como un circuito que nos permite tener comunicación con el variador de frecuencia, el diagrama del circuito se encuentra en anexo III
5. Circuito para el sensor optoelectrónico.

Una vez definido cual sería el circuito a diseñar (hardware), se realizo el grabado del circuito en la placa fenólica, con la técnica de serigrafía.

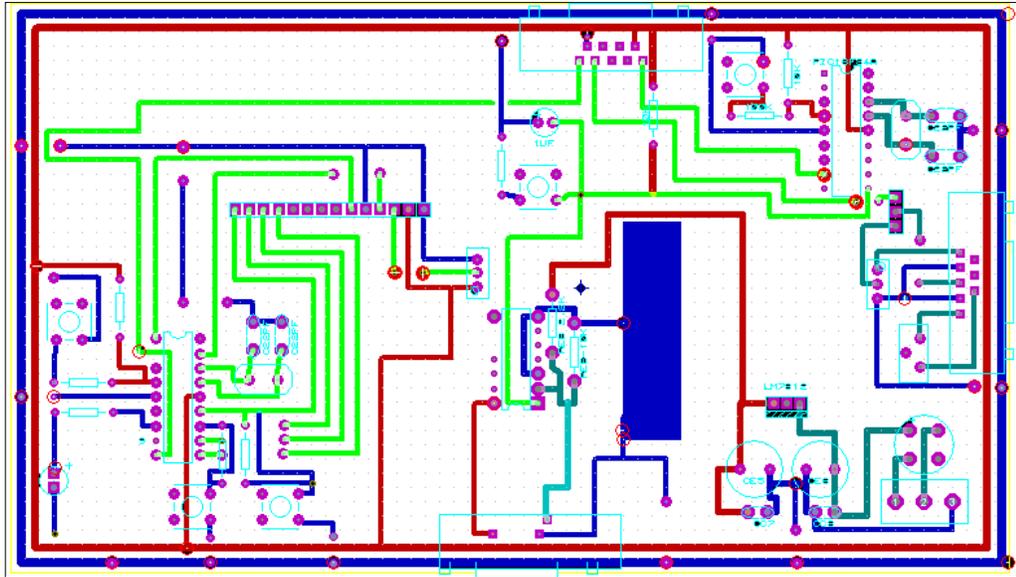


Figura 4.4 Tarjeta con circuitos implementados

El resultado final de nuestra tarjeta se muestra en la figura 4.5 la cual tiene ya montados todos componentes necesarios

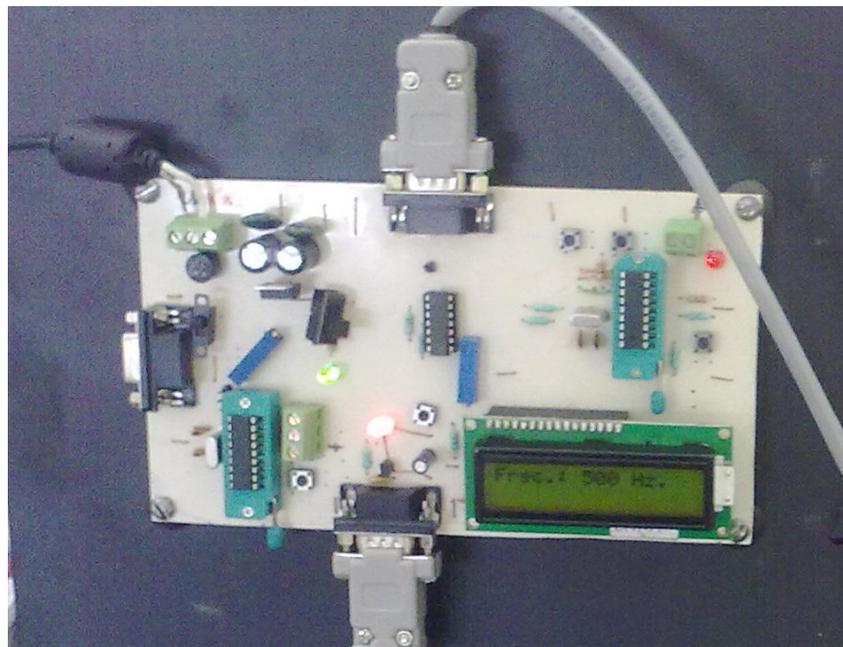


Figura 4.5 Circuito terminado.

### 4.3. Desarrollo del programa en LabView.

En LabView, se desarrollo un programa que nos permite recibir señales a través de una tarjeta de adquisición de datos, la cual es el medio de comunicación externa que tiene el ambiente de programación LabView, el programa o instrumento virtual (VI), como se le conoce en este lenguaje de programación tiene como finalidad mostrar la frecuencia que nos envía la tarjeta, mostrar en forma numérica el conteo, el cual es enviado por el sensor optoelectrónico, como una señal digital.

Como se menciona en la revisión de literatura los VI están conformados del panel frontal y diagrama de bloques; por tal motivo usamos el panel frontal para colocar los esquemas donde se visualizan los datos arrojados por el programa. En el diagrama de bloques es donde realizamos la programación. Para realizar el Vi se realizo los siguientes pasos:

1. Es necesario que nuestro programa se repita una o varias veces, por tal motivo fue necesario incluir un **while loop**, el cual nos permite hacer las iteraciones requeridas, Figura 4.6

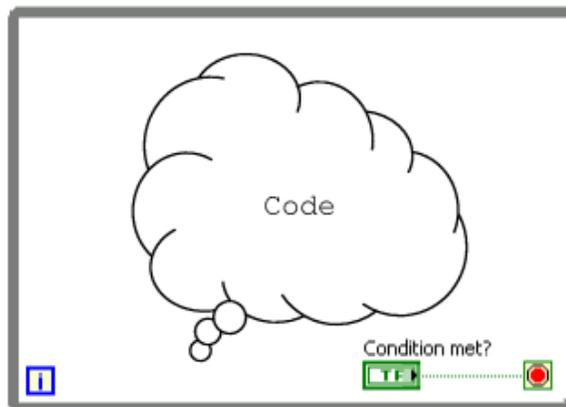


Figura 4.6. While Loop

2. Utilizamos un instrumento virtual de adquisición de datos, DAQ Assistant, el cual se coloca dentro del while loop, después lo configuramos para el tipo de señal requerida. Fugura 4.7
3. Para que nuestro caso utilizamos dos DAQ Assistant y sus respectivos indicadores.



Figura 4.7 DAQ Assistan

Para que nuestro programa funcione correctamente fue necesario instalar un software Measurement & Automation Explorer (MAX), el cual nos crea una canal virtual, de esta manera podremos realizar la adquisición de datos de manera correcta, todo lo dicho se ilustra en la siguiente figura

En la figura 4.2 se muestra el IV desarrollado, en el panel frontal a la izquierda, se visualiza los resultados el contador de semillas, identificador de señal así como el panel que grafica la simulación de la frecuencia.

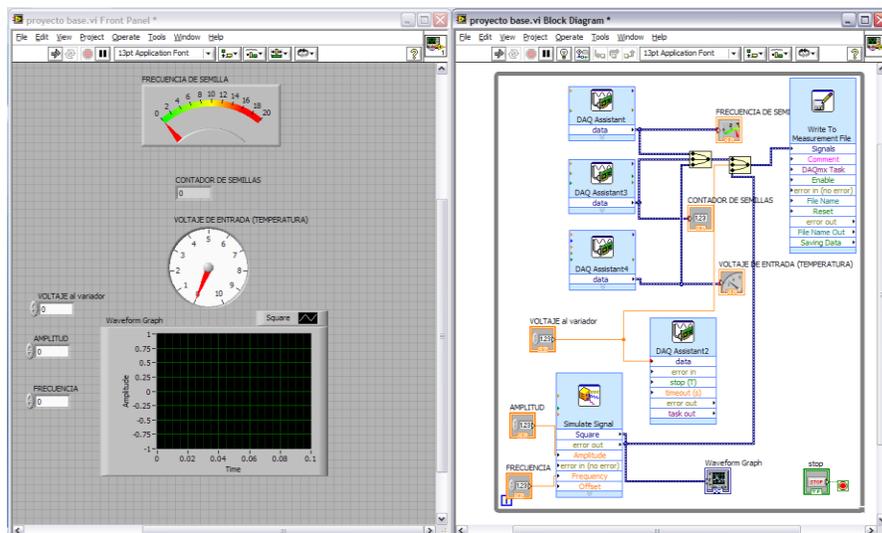


Figura 4.2. Instrumento Virtual Desarrollado.

## 4.4 Funcionamiento de componentes

### 4.4.1. Programas para simulación de dosificación.

Una vez que se que se tienen completados los componentes verificamos su funcionamiento. Lo primero que se verifico fue el programa que se utilizo para el contador de semilla en tiempo real. Su funcionamiento fue verificado en el panel de control Virtual, conectando la tarjeta de pruebas a la tarjeta de conexiones de LabView, tal y como lo muestra la figura 4.3

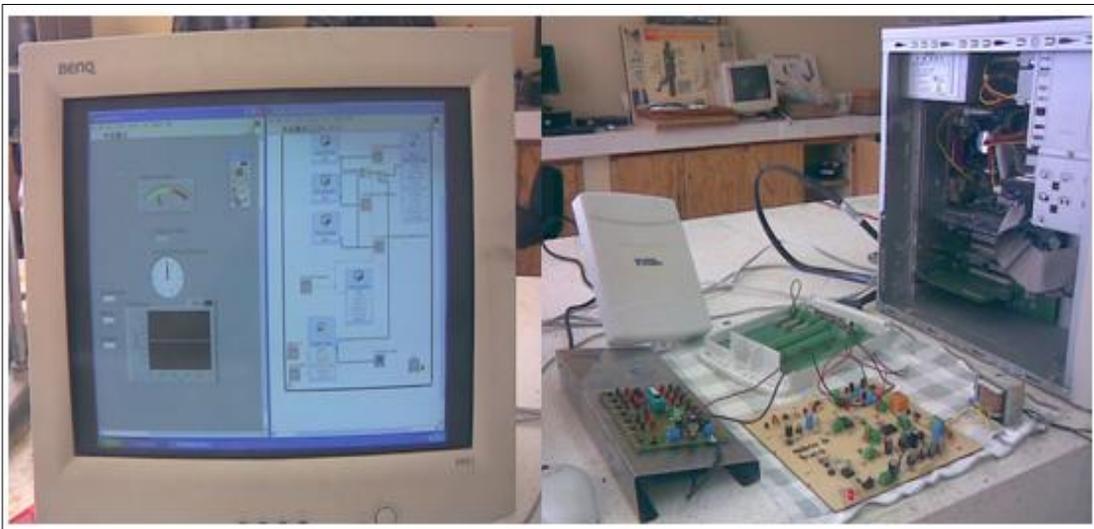


Figura 4.3 Conexión a la tarjeta de LabView con la tarjeta de pruebas del PIC16F84.

Una vez hechas la conexiones nos enfocamos a verificar que el conteo que se refleja en el monitor (Panel de frontal); lo que se hizo fue contar las pulsaciones en la tarjeta y constatar que también se contaban los mismos en el registro de semillas, los resultados se muestran en el cuadro 4.1.

**Cuadro 4.1.** Pruebas de conteo el LabView y la tarjeta de pruebas con el programa Pulsador\_05B.

<b>Pulsación en la tarjeta</b>	<b>Contador de semillas (LabView)</b>
8	4
15	7
20	10
25	13
30	15
45	23

En esta prueba verificamos que nuestro programa implementado en el microcontrolador aun no satisface nuestro propósito (por cada pulsación en la tarjeta debe contar una semilla y solo una), aquí el resultado fue que por cada pulsación par el contador de semilla nos cuenta solo la mitad y por un número impar nos cuenta una menos que la mitad.

El siguiente programa que fue el **Pulsador\_final** al cual se le adicono un circuito para evitar rebotes. Se realizo el mismo procedimiento que el anterior y se obtuvieron los siguientes resultados, los cuales se muestran en cuadro 4.2.

**Cuadro 4.2** Pruebas de conteo el LabView y la tarjeta de pruebas del Rebotes\_Final.

<b>Pulsación en la tarjeta</b>	<b>Contador de semillas (LabView)</b>
8	8
15	15
20	20
25	25
30	30

#### 4.4.2. Programa para simulación de frecuencia

El programa y el circuito que nos permite enviar la frecuencia al (IV), fue probado de manera satisfactoria ya que tanto en la pantalla de cristal líquido habilitado en nuestra tarjeta nos muestra la señal requerida al igual que el instrumento virtual, (Figura 4.4). La importancia de esta señal es que de acuerdo a la frecuencia seleccionada, esta indique cual será la velocidad de siembra.

El programa y el circuito de simulación de la frecuencia se encuentran en el anexo D, en el programa **Int\_Cuadradas\_01**.



Figura 4.4. Frecuencia en la pantalla de cristal e Instrumento virtual.

#### 4.4.3 Funcionamiento del simulador

Una vez que hemos verificado el funcionamiento de los microcontroladores armamos el simulador como lo muestra la figura 4.5.



Figura 4.5 Simulador de dosis variable de semilla.

Una vez que tenemos armado el simulador, lo primero a verificar es que el sensor optoelectrónico nos mande señal a la compuerta lógica y esta a su vez al microcontrolador que finalmente se verá reflejado en el (IV), en el contador de semillas.

La primera prueba se realizó en forma manual para constatar que el contador de semilla en el (IV), nos cuente las 20 perforaciones del disco implementado como dosificador. El resultado fue satisfactorio ya que el contador de semillas cuenta el 100% de las perforaciones, sin tener ningún porcentaje de error, al mismo tiempo se envía la frecuencia ya que de lo contrario nos marca un error el programa (IV), ya que se necesita de esta para la dosificación **Variable de Insumos.** (Figura 4.6)

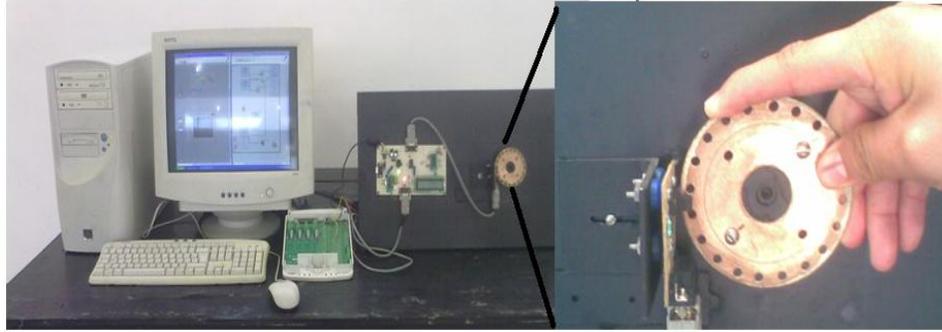


Figura 4.6. Dosificación en forma manual.

El siguiente paso fue probar el simulador con el variador de frecuencia para esto se conecto el motor trifásico de 0.25 kW al variador, el cual va a controlar la velocidad del motor. En la figura 4.7 se muestra la forma que se realizo la prueba con el variador de frecuencia.

El variador de frecuencia controla la velocidad del motor de manera satisfactoria, sin embargo cuando nos envía la señal al (IV), nos provoca ruido magnético, ya que el contador de semillas se altera arrojándonos datos de conteo erróneos.



Figura 4.7. Conexiones con el variador de frecuencia.

## V CONCLUSIONES Y RECOMENDACIONES

- Se contribuyo en el logro del objetivo general con el registro de las variables: conteo de semillas y velocidad de avance.
- Se desarrollo un software y hardware que nos permite, contabilizar al dosificador de semilla, y verificar que no existen rebotes.
- Se desarrollo un software y hardware que permite generar la frecuencia para simular la velocidad de avance de la sembradora.
- La comunicación con el variador de frecuencia quedo establecida y arrojando resultados positivos ya que se puede controlar al variador a través de un potenciómetro e ir variando la velocidad del motor.

Sin embargo se detecto ruido magnético cuando el motor fue conectado al variador de frecuencia, alterando los datos en el Instrumento Virtual. Siendo probable que esto sea producto de que los Pines de E/S del microcontrolador no fueron utilizados, es decir, quedan sin ninguna conexión, por lo que requieren que sean conectados a la fuente de alimentación de 5V.

Debido al problema presentado, de ruido magnético, se recomienda que los Pines que se encuentren sin utilizar se conecten a la fuente de alimentación de 5V.

## VI. LITERATURA CITADA

1. Angulo U. J y I. A. Martínez. 1999. Microcontroladores PIC, diseño practico de aplicaciones segunda edición, McGraw-Hill Interamericana de España, S.A.U
2. Palacios M. E, Remiro D. E y López P. L. 2004. Microcontrolador PIC16F84 Desarrollo de Proyectos. Primera Edición. Editorial Alfaomega, Mex.
3. Edison Duque C. 1998. Curso avanzado de Microcontroladores PIC, CEKIT (Compañía Editorial Tecnológica).
4. Lázaro A. M y Fernández J del Rio. 2006. LabView 7.1: Programación Grafica para el control de Instrumentos, Editorial Paraninfo, España
5. IICA. (2000) Agricultura de precisión: Integrado de conocimientos para la agricultura moderna y sustentable. Argentina
6. Anzures M J y Ruiz G. V. 2001. Diseño de un instrumento virtual para la medición de energía usando LabView. Instituto Tecnológico de Chihuahua. GIRIA (Grupo de investigación en Robótica e inteligencia Artificial) de la Universidad Industrial de Santander
7. Burgos R. C, Barrero P. J. y Correa R. C. (1999). Control de un horno microondas usando LabView.
8. CASE IH AGRICULTURE EQUIPMENT, Advanced Farming Systems, [www.casecorp.com/agricultural/afs/index.html](http://www.casecorp.com/agricultural/afs/index.html), Janeiro de 1999

9. <http://www.agriculturadeprecision.org/monred/SistemaMonitoreoCanizaAzucarParaMapas deRendimiento.htm>
10. Hancock, D. 2002. Site-Specific Farming on Small Farms. Topics in Precision Agriculture. University of Kentucky, College of Agriculture, Cooperative Extension Service. URL: [http://www.bae.uky.edu/~precag/PrecisionAg/Extension\\_pubs.htm](http://www.bae.uky.edu/~precag/PrecisionAg/Extension_pubs.htm)
11. JOHN DEERE, Precision Farming, [www.deere.com/greenstar](http://www.deere.com/greenstar), agosto de 1998.
12. Lowenberg-DeBoer, J. 1997. A bumpy road to the adoption of precision agriculture. Purdue Agricultural Economics Report. Nov. 1997. [http://www.agecon.purdue.edu/extension/pubs/paer/pre\\_98/paer1197.pdf](http://www.agecon.purdue.edu/extension/pubs/paer/pre_98/paer1197.pdf)
13. Norton, G.W. y Swinton S.M.. 2001. Precision Agriculture: Global Prospects and Environmental Implications. Forthcoming in G.H. Peters and P. Pingali, eds. Tomorrow's Agriculture: Incentives, Institutions, Infrastructure and Innovations: Proceedings of the 24th International Conference of Agricultural Economists, 2000. London: Ashgate.
14. Thompson, S. 1989. Control systems Engineering and Design. Longman Group UK, Limited.
15. US National Research Council, 1997. INTRODUCCIÓN A LA AGRICULTURA DE PRECISIÓN. Revista Digital CENIAP HOY N° 12 septiembre-diciembre 2006, Maracay, Aragua, Venezuela. ISSN: 1690-4117 Depósito Legal: pp.200302AR1449 Sitio: [www.ceniap.gov.ve](http://www.ceniap.gov.ve)
16. [http://www.pic16f84a.org/index.php?option=com\\_weblinks&view=category&id=47%3Aproteus&Itemid=104](http://www.pic16f84a.org/index.php?option=com_weblinks&view=category&id=47%3Aproteus&Itemid=104)

17. [www.redeya.com](http://www.redeya.com)
18. [www.lancos.com](http://www.lancos.com)
19. [www.microchip.com](http://www.microchip.com)
20. [www.mindspring.com/~covington/noppp](http://www.mindspring.com/~covington/noppp)

**ANEXOS**

## Anexo A

### A.1 Generalidades del PIC16F84A

El PIC16F84 al igual que la versión mejorada de mayor velocidad PIC16F84A pertenece a la familia de microcontroladores de 8 bits PIC16CXX, los cuales son de bajo costo, alto rendimiento y CMOS. Este grupo contiene los siguientes dispositivos:

- ✓ PIC16F83
- ✓ PIC16F84
- ✓ PIC16F84A
- ✓ PIC16CR83
- ✓ PIC16CR84

Todos los microcontroladores de PICmicro™ emplean una avanzada arquitectura RISC (Computadores de Juego de Instrucciones Reducido). Los dispositivos PIC16F8X han reforzado características principales, pila profunda de ocho niveles y múltiples fuentes de interrupción internas y externas.

La separación del bus de instrucciones y del bus de datos en la arquitectura Harvard da lugar a un ancho de la palabra de instrucción de 14 bits, con un bus de datos separado de 8 bits de ancho. Los dos conductos de fase de instrucción permite que todas las instrucciones se ejecuten en un solo ciclo, excepto por los saltos del programa, los cuales requieren dos ciclos.

Un total de 35 instrucciones (Juego de instrucciones reducido) están disponibles. Adicionalmente un set del registro largo es usado para lograr un

nivel de alto rendimiento. Los microcontroladores PIC16F8x típicamente logran una compresión de código de 2:1 y un incremento de 4:1 en la mejora de velocidad (a 20 MHz) sobre otros microcontroladores de 8 bits de su misma clase. En el cuadro 1.1 se muestran las características principales de la serie de microcontroladores PIC16F8x.

Los microcontroladores PIC16F8x poseen 68 bytes de RAM, 64 bytes de memoria de datos EEPROM y 13 pines de entrada / Salida. También está disponible un contador / temporizador.

La familia PIC16CXX tiene características especiales para reducir los componentes externos, reduciendo así el costo, reforzando la confiabilidad del sistema y reduciendo el consumo de energía.

Hay cuatro opciones del oscilador, de los cuales el oscilador RC mediante un solo pin, proporciona una solución de bajo costo, el oscilador LP reduce el consumo de energía, el oscilador XT es un cristal estándar y el oscilador HS se refiere a cristales de alta velocidad.

El modo SLEEP (Bajo consumo de corriente) ofrece un ahorro de energía. El usuario puede despertar al chip del modo SLEEP a través de varias interrupciones internas y externas para reestablecer su funcionamiento.

Un confiable temporizador de perro guardián (WDT) está integrado en el chip, así como su propio oscilador RC proporciona protección cuando el programa ha sido ciclado. Los dispositivos con memoria de programa Flash permite que el mismo dispositivo pueda ser usado como prototipo y como elemento de producción. La gran capacidad de reprogramar el código del circuito permite su actualización sin que el dispositivo sea removido de su aplicación final.

Cuadro A.1 Lista de características del PIC16F8x

	PIC16F83	PIC16CR83	PIC16F84	PIC16CR84	PIC16F84A
Máxima frecuencia de operación MHz	10	10	10	10	20
Memoria de programa Flash	512	-	1K	-	1K
Memoria de programa EEPROM	-	-	-	-	-
Memoria de programa ROM	-	512	-	1K	-
Memoria de datos (bytes)	36	36	68	68	68
EEPROM de datos (bytes)	64	64	64	64	64
Módulos Temporizadores	TMRO	TMRO	TMRO	TMRO	TMRO
Fuentes de interrupción	4	4	4	4	4
Pines de I/O	13	13	13	13	13
Rangos de Voltaje (Volts)	4.0-6.0	2.0-6.0	4.0-6.0	2.0-6.0	4.0-6.0
Empaques	Dip 18-pin, SOIC				

Esto es muy útil en situaciones, donde el dispositivo es difícilmente accesible. También es útil en aplicaciones remotas donde el código necesita ser actualizado. El PIC16F8x encaja perfectamente en rangos de aplicaciones donde se requiere controlar pequeños motores, en la utilización de sensores remotos de baja potencia, cerrojos electrónicos y pequeñas tarjetas con dispositivos de seguridad.

La tecnología Flash/EEPROM optimiza los programas de aplicación (transmisión de códigos, velocidad de motores, receptor de frecuencias, códigos de seguridad, etc. ) volviéndolos extremadamente rápidos y convenientes. Los pequeños empaques hacen perfectos a esta serie de microcontroladores para todas las aplicaciones con limitaciones de espacio.

Bajo costo, bajo consumo de energía, alto rendimiento, fácil de usar y flexibilidad en las entradas/salidas, hacen muy versátil al PIC16F8x en áreas donde no se habían considerado los microcontroladores (funciones de temporizador, comunicación serie, capturadores, comparadores, moduladores por ancho de pulso, y aplicaciones como procesador).

La medida de programación serial en el sistema (por medio de dos pines), ofrece flexibilidad en la optimización del producto después de haber sido ensamblado y probado. Esta característica puede ser usada para agregar un número de serie a un producto, una calibración o reestablecimiento de datos o programar el dispositivo con la firma de software correspondiente antes de enviarlo.

El diagrama de pines del microcontrolador PIC16F84 se puede apreciar en la figura A.1

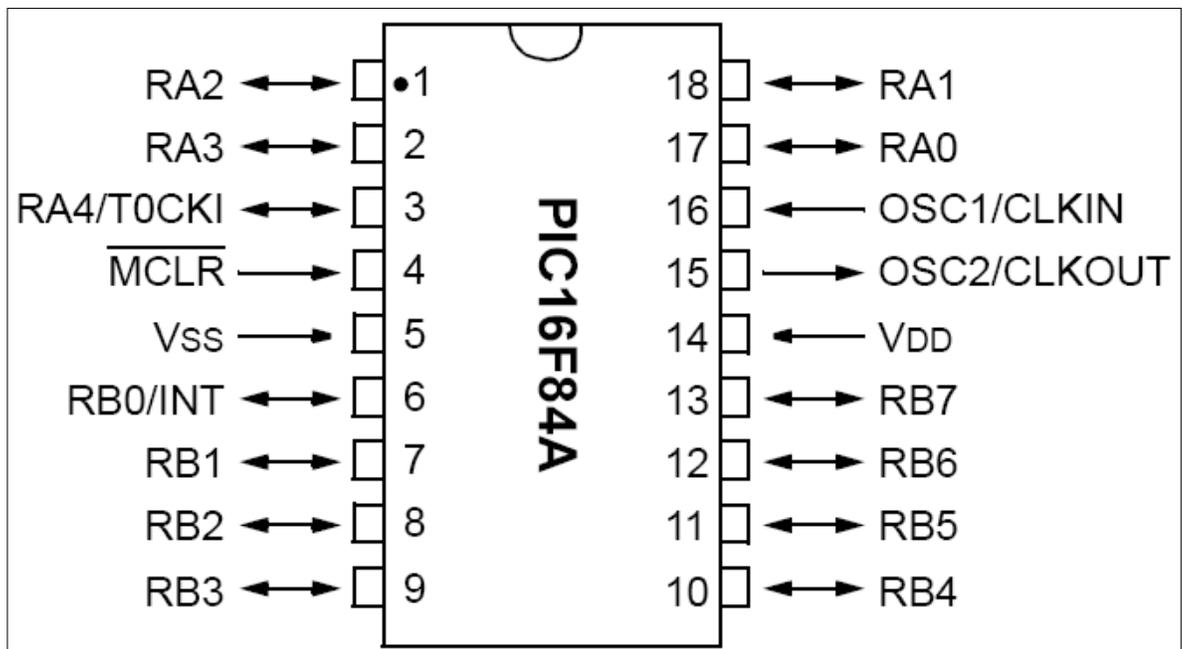


Figura A.1 Diagrama de pines del PIC16F84

La descripción de cada uno de los pines se muestra en el cuadro A.2

Cuadro A.2 Descripción de Pines

Nombre	Nº	Tipo	Descripción
OSC1/CLKIN	16	I	Entrada del oscilador a cristal/Entrada de la fuente de reloj externa
OSC2/CLKOUT	15	O	Salida del oscilador a cristal. En el modo RC, es una salida con una frecuencia de $\frac{1}{4}$ OSC1
MCLR	4	I/P	Reset/Entrada del voltaje de programación.
RA0	17	I/O	Puerto A bidireccional, bit 0
RA1	18	I/O	Puerto A bidireccional, bit 1
RA2	1	I/O	Puerto A bidireccional, bit 2
RA3	2	I/O	Puerto A bidireccional, bit 3
RA4/T0CKI	3	I/O	También se utiliza para la entra de reloj para el TMR0
RB0/INT	6	I/O	Puerto B bidireccional, bit 0 Puede seleccionarse para entrada de interrupción externa
RB1	7	I/O	Puerto B bidireccional, bit 1
RB2	8	I/O	Puerto B bidireccional, bit 2
RB3	9	I/O	Puerto B bidireccional, bit 3
RB4	10	I/O	Puerto B bidireccional, bit 4 Interrupción por cambio de estado
RB5	11	I/O	Puerto B bidireccional, bit 5 Interrupción por cambio de estado
RB6	12	I/O	Puerto B bidireccional, bit 6 Interrupción por cambio de estado
RB7	13	I/O	Puerto B bidireccional, bit 7 Interrupción por cambio de estado
Vss	5	P	Tierra de referencia
Vdd	14	P	Alimentación

El pin RA4/TOCKI del puerto A puede ser configurado como un pin de entrada/salida o como entrada del temporizador/contador. Cuando este pin se programa como entrada digital, funciona como un disparador de Schmitt (Schmitt trigger), puede reconocer señales un poco distorsionadas y llevarlas a niveles lógicos (cero y cinco voltios). Cuando se usa como salida digital se comporta como colector abierto, por lo tanto, se debe poner una resistencia de pull-up (resistencia externa conectada a un nivel de cinco voltios). Como salida, la lógica es inversa: un “0” escrito al pin del puerto entrega en el pin un “1” lógico.

Todos los pines deben estar conectados a alguna parte, nunca dejarlos al aire por qué se puede dañar el integrado. Los pines que no se estén usando se deben conectar a la fuente de alimentación de +5V, como se muestra en la Figura A.2

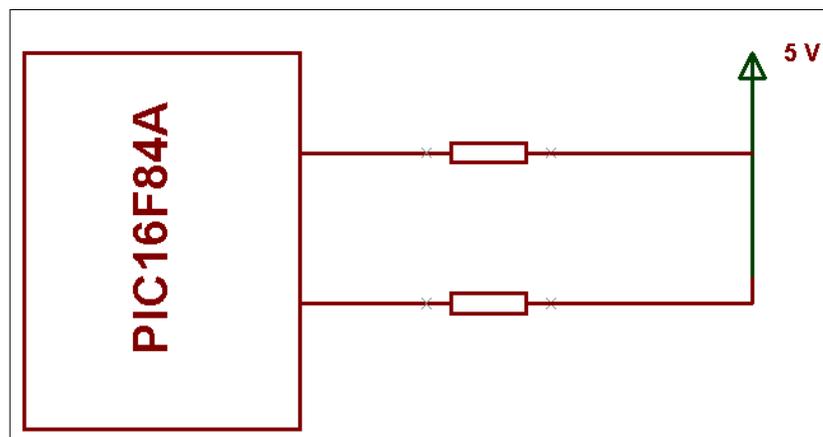


Figura A.2 Los puertos no utilizados se deben conectar a la fuente.

El consumo de corriente del microcontrolador para su funcionamiento depende del voltaje de operación, la frecuencia y de las cargas que tengan sus pines. Para un reloj de 4 MHz el consumo es de aproximadamente 2 mA; aunque este se puede reducir a 40 microamperios cuando se está en el modo sleep (en este

modo el micro se detiene y disminuye en consumo de potencia). Se sale de ese estado cuando se produce alguna condición especial

### **El oscilador externo**

Todo microcontrolador requiere un circuito externo que le indique la velocidad a la que debe trabajar, se les conoce como oscilador o reloj, es muy simple pero de vital importancia para el buen funcionamiento del sistema. El PIC16F84 puede utilizar 4 tipos de osciladores diferentes. Estos tipos son:

- **RC.** Oscilador con resistencia y condensador
- **XT.** Cristal
- **HS.** Cristal de alta velocidad
- **LP.** Cristal para baja frecuencia y bajo consumo de potencia

En el momento de programar o “quemar” el microcontrolador se debe especificar que tipo de oscilador se usa, esto se hace a través de los “fusibles de configuración”.

El tipo de oscilador que se sugiere es el de 4 MHz, por que garantiza mayor precisión y un buen arranque del microcontrolador. Internamente esta frecuencia es dividida por cuatro, lo que hace que la frecuencia efectiva de trabajo sea de 1 MHz, por lo que cada instrucción se ejecuta en un microsegundo. El cristal se conecta como se muestra en la Figura A.3.

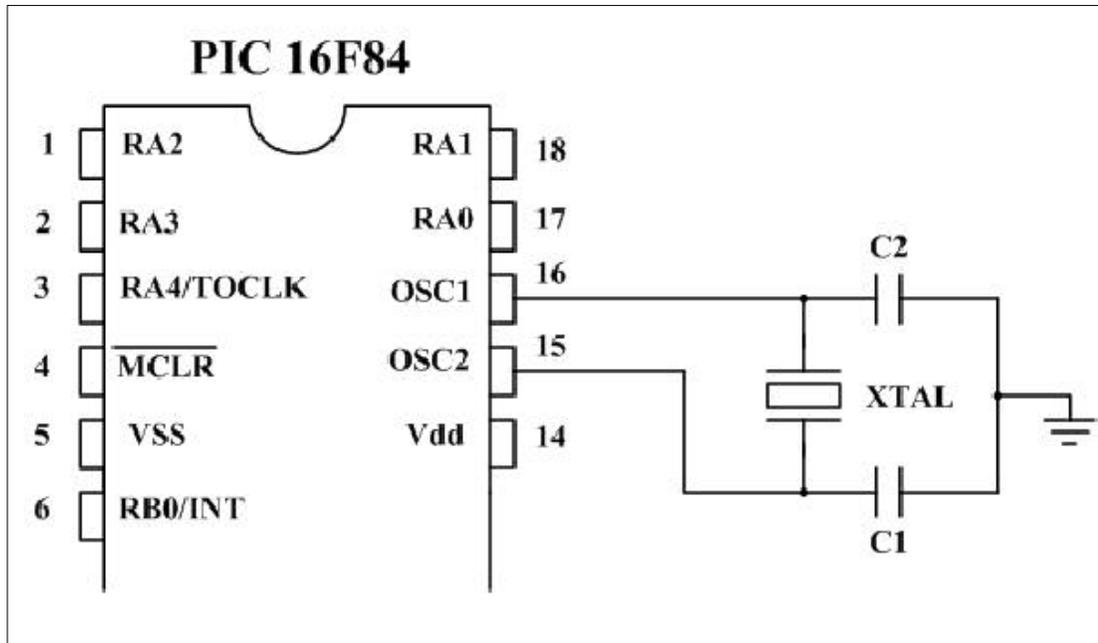


Figura A.3 Conexión de un oscilador de cristal.

Dependiendo de la aplicación, se puede utilizar cristales de otras frecuencias; por ejemplo el de 3.579545 MHz por que es muy económico, el de 32.768 kHz cuando se necesita crear bases de tiempo de un segundo muy precisas. El límite de velocidad en estos microcontroladores es de 10 MHz.

Si no se requiere mucha precisión en el oscilador y se requiere economizar dinero, se puede utilizar una resistencia y un condensador, como se muestra en la Figura A.4.

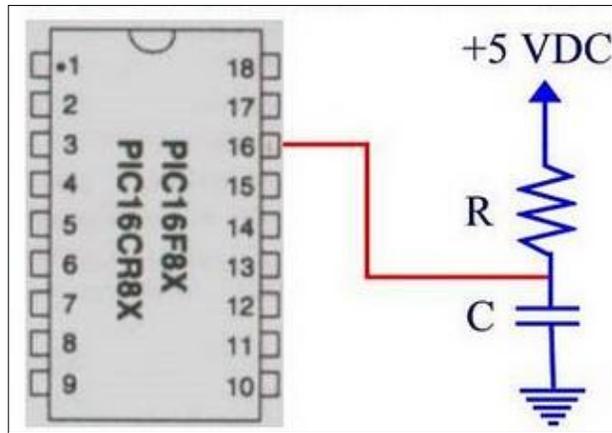


Figura A.4 Conexión de un oscilador RC.

## Reset

En los microcontroladores se requiere un pin de reset para reiniciar el funcionamiento del sistema cuando sea necesario, ya sea por una falla que se presente o por que así se halla diseñado el sistema. El pin de reset en los PIC es llamado MCLR (master clear). El PIC16F84 admite diferentes tipos de reset:

- Al encendido (Power On Reset)
- Pulso en el pin MCLR durante operación normal
- Pulso en el pin MCLR durante el modo de bajo consumo (modo sleep)
- El rebase del conteo del circuito de vigilancia (watchdog) durante operación normal
- El rebase del conteo del circuito de vigilancia (watchdog) durante el modo de bajo consumo (sleep)

El reset para el encendido se consigue con dos temporizadores. El primero de es el OST (Oscillator Star-Up Timer: Temporizador de encendido del oscilador), orientado a mantener el microcontrolador en reset hasta que el oscilador del cristal es estable. El segundo es el PWRT (Power-Up Timer: Temporizador de encendido), que provee un retardo fijo de 72 ms (nominal) en el encendido

únicamente, diseñado para mantener el dispositivo en reset mientras la fuente se estabiliza.

El reset por MCLR se consigue llevando momentáneamente este pin a un estado lógico bajo, mientras que el watchdog WDT produce el reset cuando su temporizador rebasa la cuenta, o sea que pasa de 0FFh a 00h. Cuando se quiere tener control sobre el reset del sistema se puede conectar un botón como se muestra en la Figura A.5.

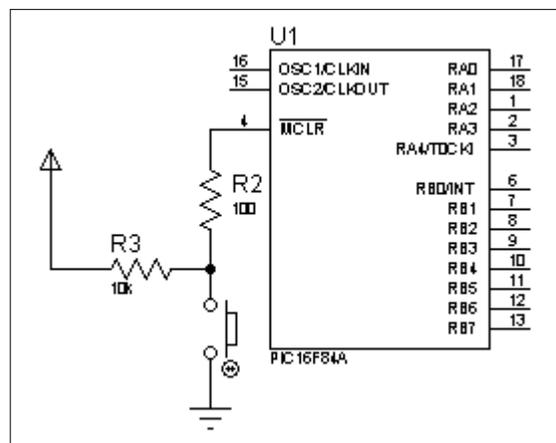


Figura A.5. Conexión del botón de reset.

## A.2 Características de la estructura RISC del CPU.

Las características de la estructura RISC del CPU son las siguientes:

- Solo 35 sencillas instrucciones para aprender.
- Todas las instrucciones son de un ciclo, excepto los saltos de programa que son de dos ciclos.
- Velocidad: 20MHz (0.2us por instrucción)
- Memoria de programa de 1024 palabras
- Memoria RAM de 68 bytes

- Memoria EEPROM de 64 Bytes
- Palabras de instrucción de 14 bits de ancho
- Bytes de datos de 8 bits de ancho
- 15 registros de función especial en el hardware
- Pila de hardware de 8 niveles de profundidad
- Modos de direccionamiento directo e indirecto
- Cuatro fuentes de interrupción:
- Pin externo de RB0/INT
- Temporizador TMR0 de sobreflujo
- Interrupción en cambio PORTB <RB7:RB4>
- Escritura completa de la memoria EEPROM

### **A.3 Características de los periféricos**

- 13 pines de entrada / salida con control individual de dirección
- Fuente / sumidero de alta corriente para controlador directo de LED.
- Sumidero de 25mA máximo por pin.

### **A.4 Arquitectura interna**

Este término se refiere a los bloques funcionales internos que conforman el microcontrolador y la forma en que están conectados, por ejemplo la memoria FLASH (de programa), la memoria RAM (de datos), los puertos, la lógica de control que permite que todo el conjunto funcione, etc.

La Figura A.6 muestra la arquitectura general del PIC16F84, en ella se pueden apreciar los diferentes bloques que lo componen y la forma en que se conectan.

Todos los elementos se conectan entre sí por medio de un conjunto de líneas que transportan información entre dos o más módulos. Vale la pena destacar

que el PIC16F84 tiene un bloque especial de memoria de datos de 64 bytes del tipo EEPROM, además de los dos bloques de memoria principales que son el de programa y el de datos o registros

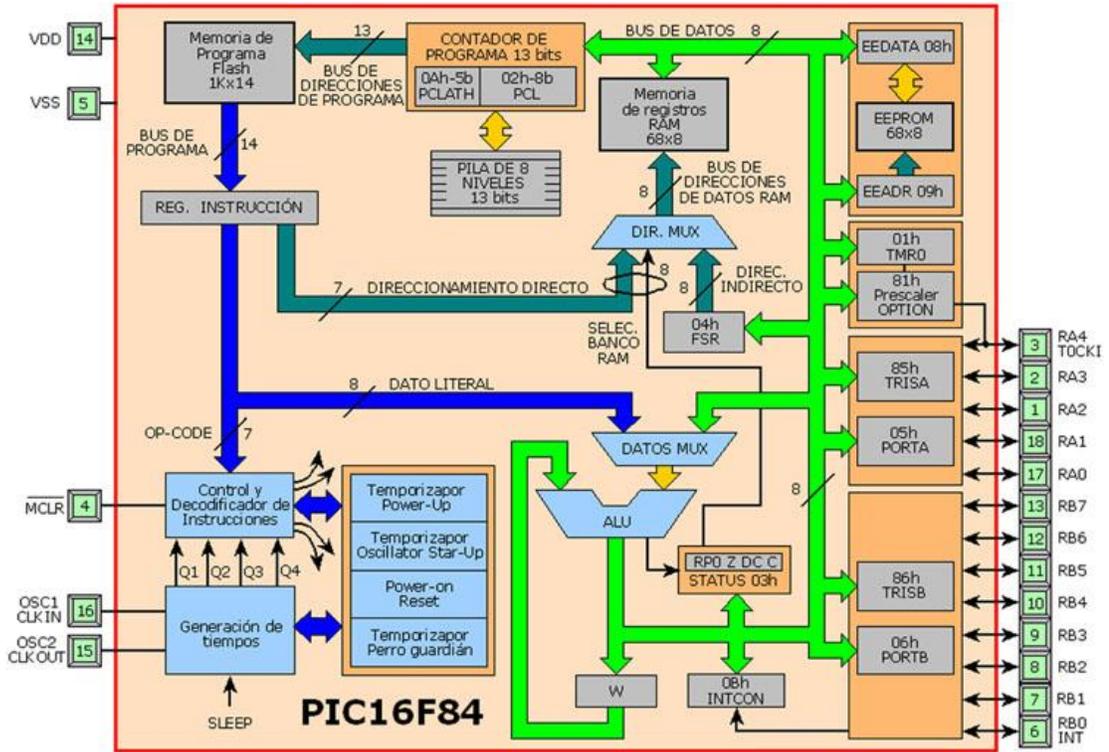


Figura A.6. Arquitectura interna del PIC16F84.

## Memoria de programa

Es una memoria de 1 Kbyte de longitud con palabras de 14 bits. En ella se graba, el programa que el microcontrolador debe ejecutar. El PIC16F84 tiene un contador de programa de 13 bits, y una capacidad de direccionamiento de 8K x 14, pero solamente tiene implementado el primer 1K x 14 (0000h hasta 03FFh). Si se direccionan posiciones de memoria mayores a 3FF se causará un solapamiento con el espacio del primer 1K. En la Figura A.7 se muestra el mapa de la memoria de programa.

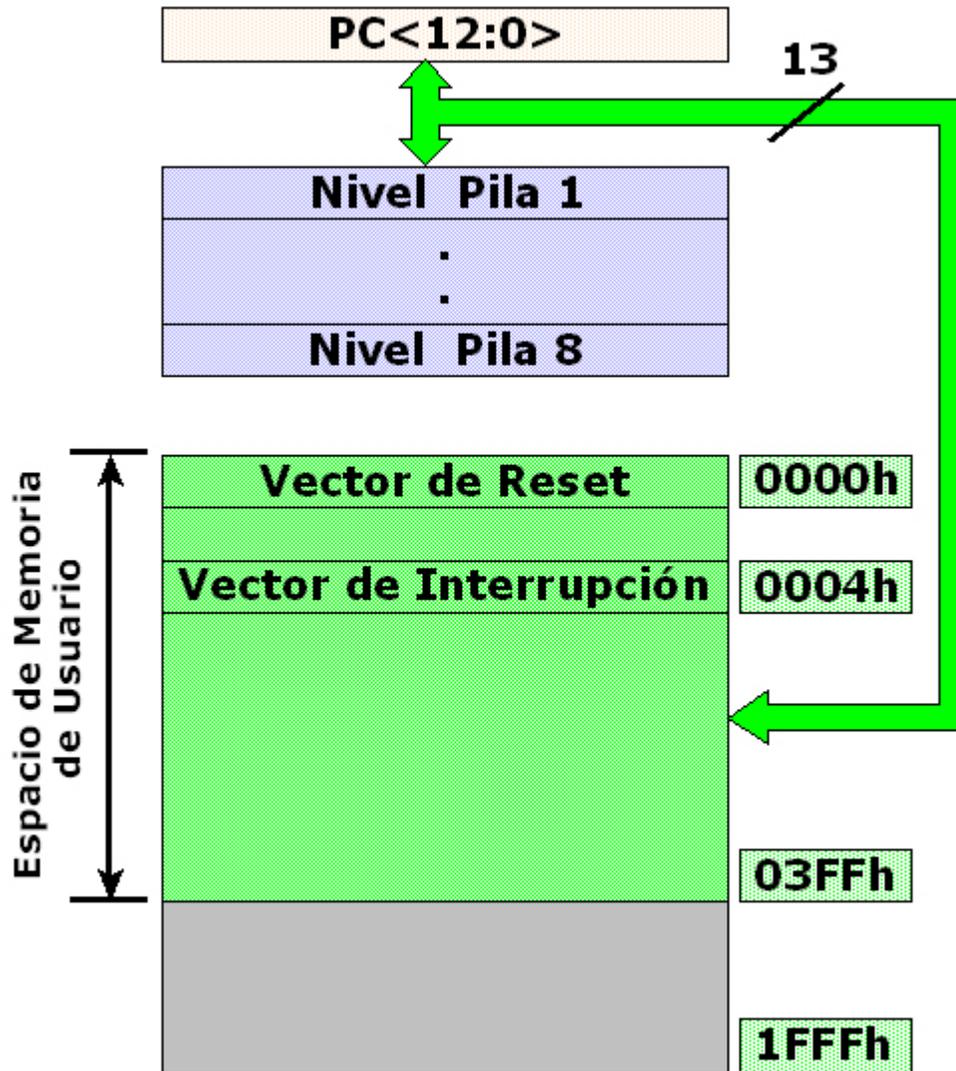


Figura A.7. Mapa de la memoria de programa.

Vector de reset. Cuando ocurre un reset al microcontrolador, el contador de programa se pone en ceros (000H), por esta razón, en la primera dirección del programa se debe escribir todo lo relacionado con la iniciación del mismo.

Vector de interrupción. Cuando el microcontrolador recibe una señal de interrupción, el contador de programa apunta a la dirección 04H de la memoria de programa, por eso, allí se debe escribir toda la programación necesaria para atender dicha interrupción.

## Registros (Memoria RAM)

El PIC16F84 puede direccionar 128 posiciones de memoria RAM, pero solo tiene implementados físicamente los primeros 80 (0-4F en hexadecimal). De estos los primeros 12 son registros que cumplen un propósito especial en el control del microcontrolador y los 68 siguientes son registros de uso general que se pueden usar para guardar los datos temporales de la tarea que se está ejecutando, Figura 1.8.

Los registros están organizados como dos arreglos (páginas) de 128 posiciones de 8 bits cada una (128 x 8); todas las posiciones se pueden acceder directa o indirectamente (esta última a través del registro selector FSR). Para seleccionar que página de registro se trabaja en un momento determinado se utiliza el bit RP0 del registro STATUS.

Descripción de los registros.

**00h o INDO: Registro para direccionamiento indirecto de datos.** Este no es un registro disponible físicamente; utiliza el contenido del FSR y el bit RP0 del registro STATUS para seleccionar indirectamente la memoria de datos o RAM del usuario; la instrucción determinara que se debe realizar con el registro señalado.

**01h o TMR0. Temporizador/contador de 8 bits.** Este se puede incrementar con una señal externa aplicada al pin RA4/TOCKI o de acuerdo a una señal interna proveniente del reloj de instrucciones del microcontrolador. La ruta de incremento del registro de puede determinar por medio de un preescalador, localizado en el registro OPTION. Como una mejora, se le ha agregado la generación de interrupción cuando se rebasa la cuenta (el paso de 0FFh a 00h).

Dir. de registro	BANCO 0	BANCO 1	Dir. de registro
00h	Dir. Ind. <sup>1</sup>	Dir. Ind. <sup>1</sup>	80h
01h	TMRO	OPTION	81h
02h	PCL	PCL	82h
03h	STATUS	STATUS	83h
04h	FSR	FSR	84h
05h	PORTA	TRISA	85h
06h	PORTB	TRISB	86h
07h	-	-	87h
08h	EEDATA	EECON1	88h
09h	EEADR	EECON2 <sup>1</sup>	89h
0Ah	PCLATH	PCLATH	8Ah
0Bh	INTCON	INTCON	8Bh
0Ch	68 REGISTROS DE PROPÓSITO GENERAL	MAPEADOS (ACCESO) EN EL BANCO 0	8Ch
4Fh			CFh
50h			D0h
7Fh			FFh

Localización de memoria no implementada, se lee como '0'

Nota 1: No es un registro físico

Figura A.8. Registros del PIC16F84.

**02h o PCL: Contador de programa.** Se utiliza para direccionar las palabras de 14 bits del programa del usuario que se encuentra almacenado en la memoria ROM; este contador de programa es de 13 bits de ancho, Figura A.9. Sobre el byte bajo, se puede escribir o leer directamente, mientras que sobre el byte alto, no. El byte alto se maneja mediante el registro PCLATH (0Ah). A diferencia de los PIC de primera generación, el 16F84 ante una condición de reset inicia el contador de programa con todos sus bits en “cero”.

Durante la ejecución normal del programa, y dado que todas las instrucciones ocupan sólo una posición de memoria, el contador se incrementa en uno con cada instrucción, a menos que se trate de alguna instrucción de salto.

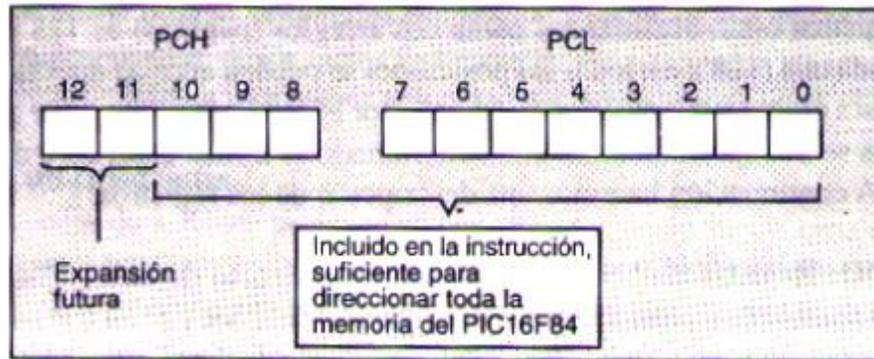


Figura A.9. Contador de programa de (13 bits).

## Anexo B

### B.1 Uso del MPLAB y grabado del microcontrolador

El MPLAB es un entorno de desarrollo integrado que le permite escribir y codificar los microcontroladores PIC de Microchip para ejecutarlos. El MPLAB incluye un editor de texto, funciones para el manejo de proyectos, un simulador interno y una variedad de herramientas que lo ayudarán a mantener y ejecutar su aplicación. También provee una interfase de usuario para todos los productos con lenguaje Microchip, programadores de dispositivos, sistemas emuladores y herramientas de tercer orden.

El MPLAB está diseñado para ser ejecutado bajo Windows 3.11, y puede operar con Windows 95 y 98. Asume que el usuario ya conoce el entorno de Windows y sabe manejarlo.

El código del programa puede estar escrito el blog de notas, tomando en consideración que nuestro archivo debe tener siempre la extensión “.asm”, ya que nuestro programa los reconoce de esta manera, para su posterior ejecución.

Es recomendable que nuestros archivos en “asm” estén guardados en **C**, para que el programa evite una búsqueda por todos los sectores; lo cual nos ocasionara un error cuando se esté ensamblando el programa.

Nuestro siguiente paso es ejecutar el archivo “asm”, de la siguiente manera: nos dirigimos al menú de herramientas damos click en File, y seleccionamos Open buscamos nuestro archivo deseado y lo abrimos, figura

B.1

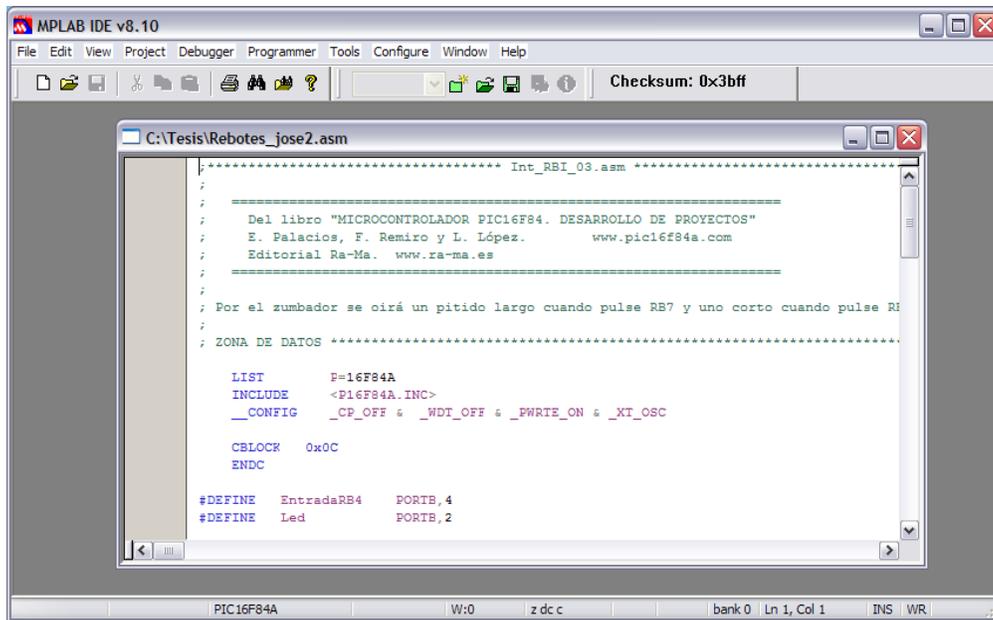


Figura B.1. Selección del programa “asm”

Una vez que hemos abierto el archivo, el siguiente paso es ensamblarlo, es decir, correr el programa para verificar su correcta escritura y ejecución, esto se hace como lo indica la figura B.2

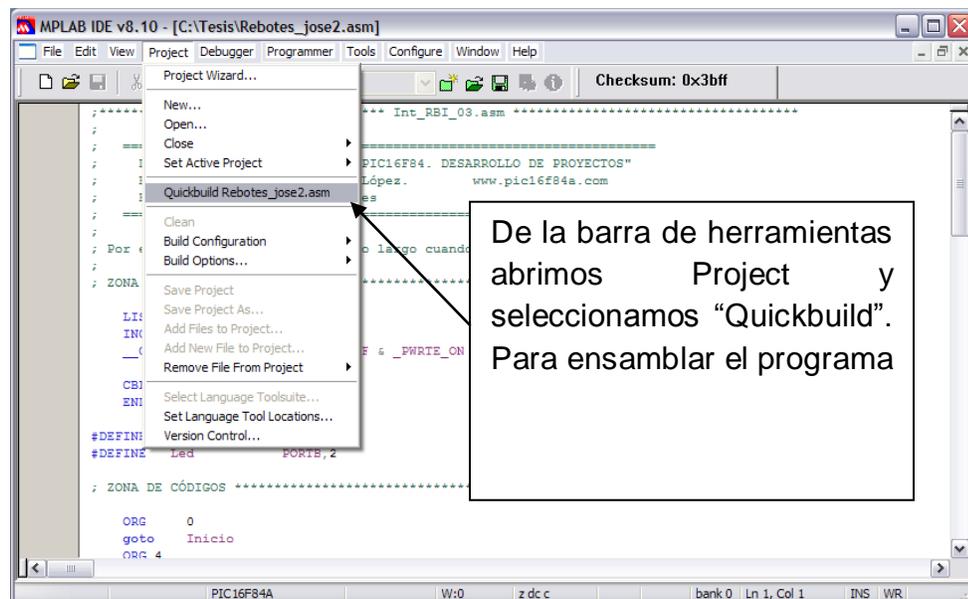


Figura. B.2. Procedimiento para el ensamblado

Una vez realizado esto, el programa nos enviara un mensaje como este: **BUILD SUCCEEDED**, lo que significa que ha sido ejecutado correctamente, lo que nos indica que podemos continuar con el grabado del micro controlador. De lo contrario nos enviara este otro mensaje: **BUILD FAILED**, lo cual nos indica que existe algún error y debemos verificar nuevamente nuestro programa.

## B.2 Grabado del microcontrolador

Para realizar este procedimiento es necesario contar con el siguiente equipo:

1. High-Speed USB Serial Adapter Keyspan by Tripp Lite, figura B.3

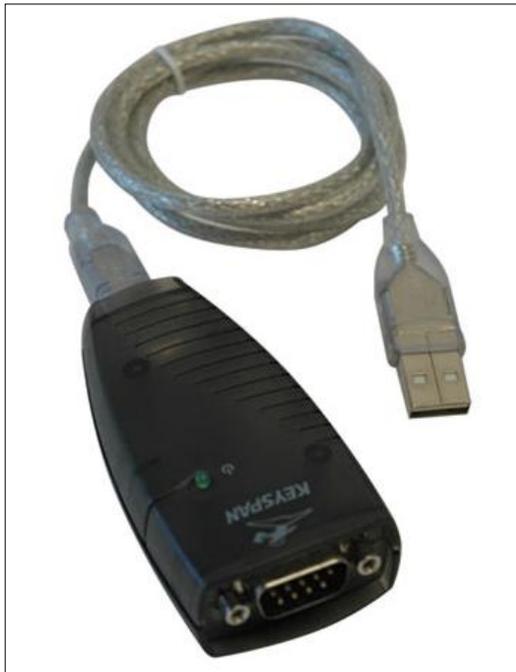


Figura B.3. USB Adapter Keyspan

2. El quemador o grabador PICSTART Plus, figura B.5

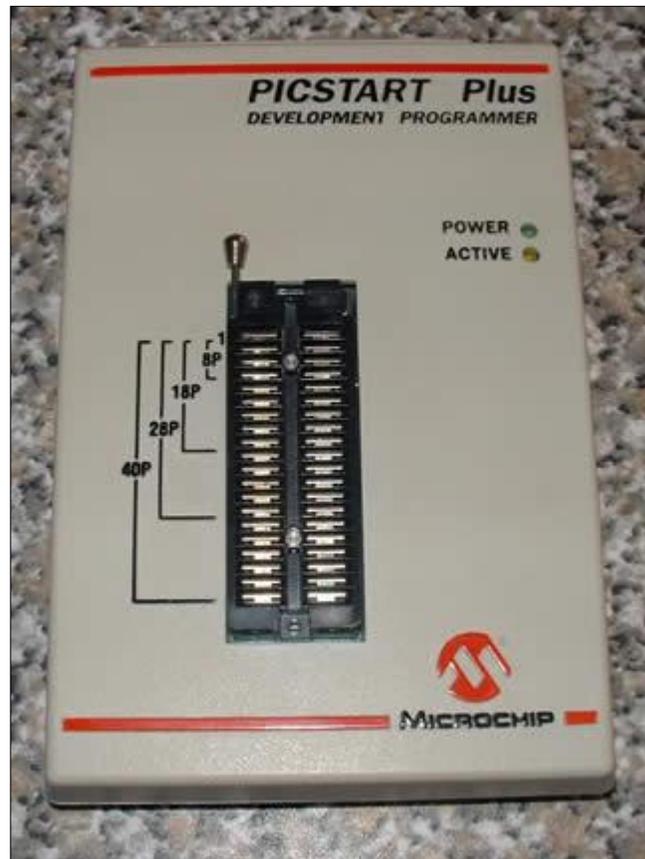


Figura B.5. PICSTART Plus.

Cuando hemos verificado que contamos con el equipo necesario volvemos a la parte donde se quedo ensamblado y verificado el buen funcionamiento del programa. Lo que prosigue es habilitar el PICSTART Plus desde el MPLAB lo cual se hace de la siguiente manera:

Nos dirigimos al menú de herramientas damos click en Programmer, seleccionamos 1PICSTART Plus, de esta manera seleccionamos nuestra programa externo, figura B.6; después volvemos a seleccionar Programmer y seleccionamos Enable Programmer para que así quede habilitado nuestro programa.

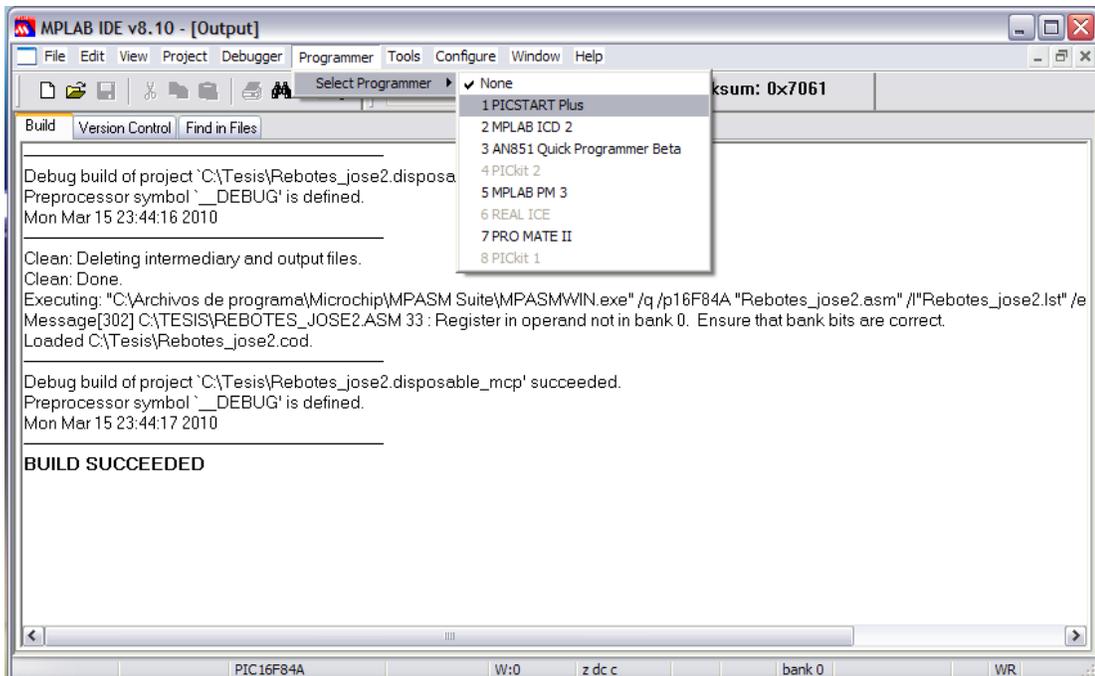


Figura B.6. Habilitación del PICSTART Plus.

Nota: si no es habilitado nuestro grabador es debido a que tenemos mal configurado nuestro puerto COM, de ser así lo habilitamos de la siguiente manera:

Volvemos al menú herramientas, abrimos el Programmer y escogemos settings, el cual abrirá una ventana que nos permitirá habilitar el puerto que se encuentre habilitado, figura B.7

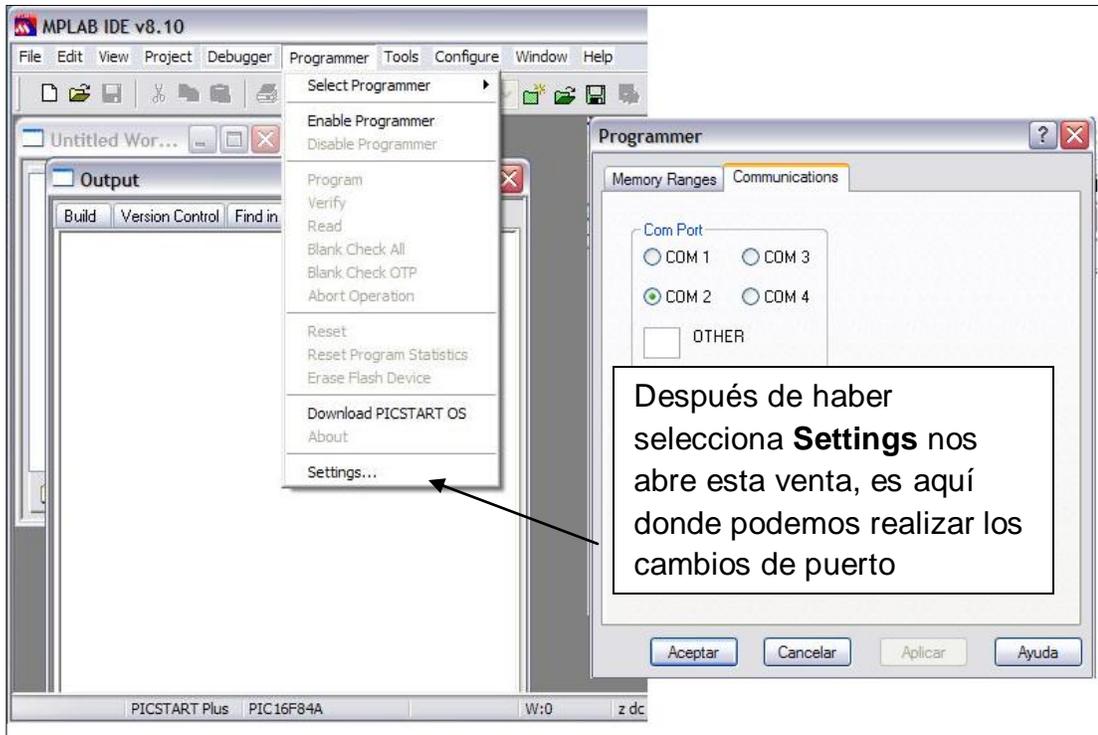


Figura B.7 Selección de puertos

Cuando tengamos habilitado nuestro programa procedemos verificar que el microcontrolador este vacío para hacer el grabado, figura B.8



Seleccionamos Read, para leer en contenido del microcontrolador

Figura B.8. Para leer el contenido

Después verificamos la memoria del PIC para constatar que se encuentra vacío o bien que contenga información, Figura B.9

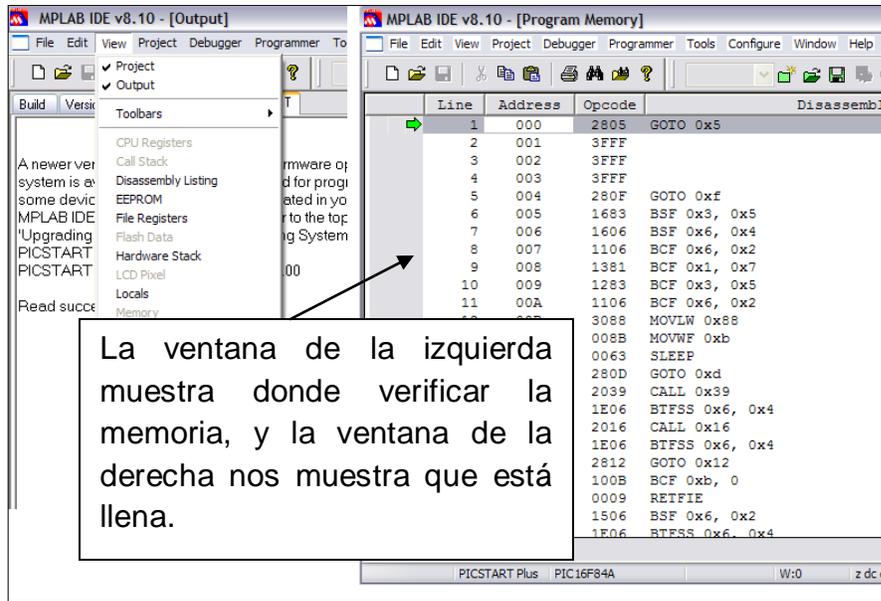


Figura B.9. Verificación de memoria

Si encontramos que la memoria está llena la borramos haciendo clic en Erase Flash Device como lo muestra



Después, del menú de herramientas seleccionamos Debugger, cuando abre la venta seleccionamos Clear Memory y All Memory, y después volvemos a verificar la Memoria del PIC, como se explico antes. Figura 2.10

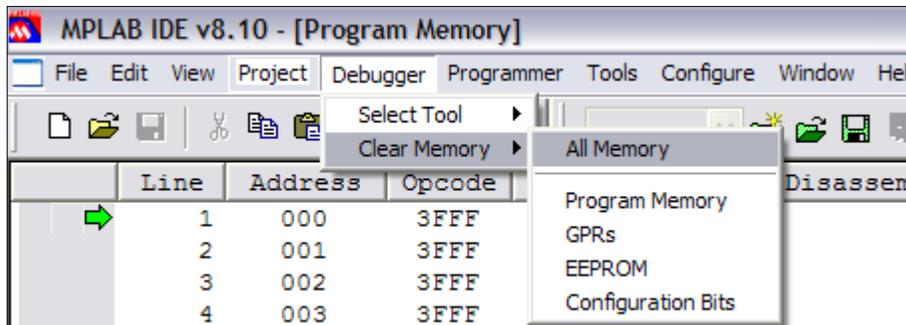


Figura 2.10. Borrar la memoria del microcontrolador

Finalmente cuando hemos verificado que la memoria del microcontrolador está vacía, procedemos a grabar el microcontrolador, seleccionamos Program para grabar en el microcontrolador.



La conexión de los componentes se ilustra en la siguiente Figura 2.11.



Figura 2.11 Conexión del PICSTART Plus

## Anexo C

### C.1 Poner en servicio el convertidor MICROMASTER 420

# SIEMENS



### C.2 Poner en servicio el convertidor MICROMASTER 420

El MM420 se entrega equipado con un panel SDP (Status Display Panel) y parámetros ajustados por defecto que cubren los requisitos siguientes:

- Los datos nominales del motor – tensión, corriente y frecuencia – son todos compatibles con los datos del convertidor (se recomienda utilizar un motor estándar de Siemens).

- Característica V/f lineal de variación de velocidad en el motor, controlada por un potenciómetro analógico.
- Velocidad máxima 3000 min<sup>-1</sup> con 50 Hz (3600 min<sup>-1</sup> con 60 Hz), controlable con un potenciómetro a través de las entradas analógicas del convertidor
- Tiempo de aceleración / tiempo de desaceleración = 10 s

Si se requieren ajustes para aplicaciones más complejas, consultar la lista de parámetros en las Instrucciones de uso. Para poder modificar parámetros es necesario uno de los módulos opcionales "Panel BOP", "Panel AOP" o las opciones de comunicaciones. Ver las

Instrucciones de uso y el Manual de referencia. **En esta Guía rápida se explica la forma de poner en marcha con el panel SDP y la "Puesta en servicio rápida" con el panel BOP (incluyendo los parámetros necesarios).**

La forma de sustituir el panel SDP por un panel de operador se explica en el

**Nota:**

Ajuste de la frecuencia: El convertidor se entrega como sigue:

- Interruptor DIP 2:
  - Posición Off: ajustes por defecto europeos (50 Hz, kW etc.)
  - Posición On: ajustes por defecto norteamericanos (60 Hz, hp etc.)
- Interruptor DIP 1: No para uso del cliente.

### C.3. Ajustes por defecto

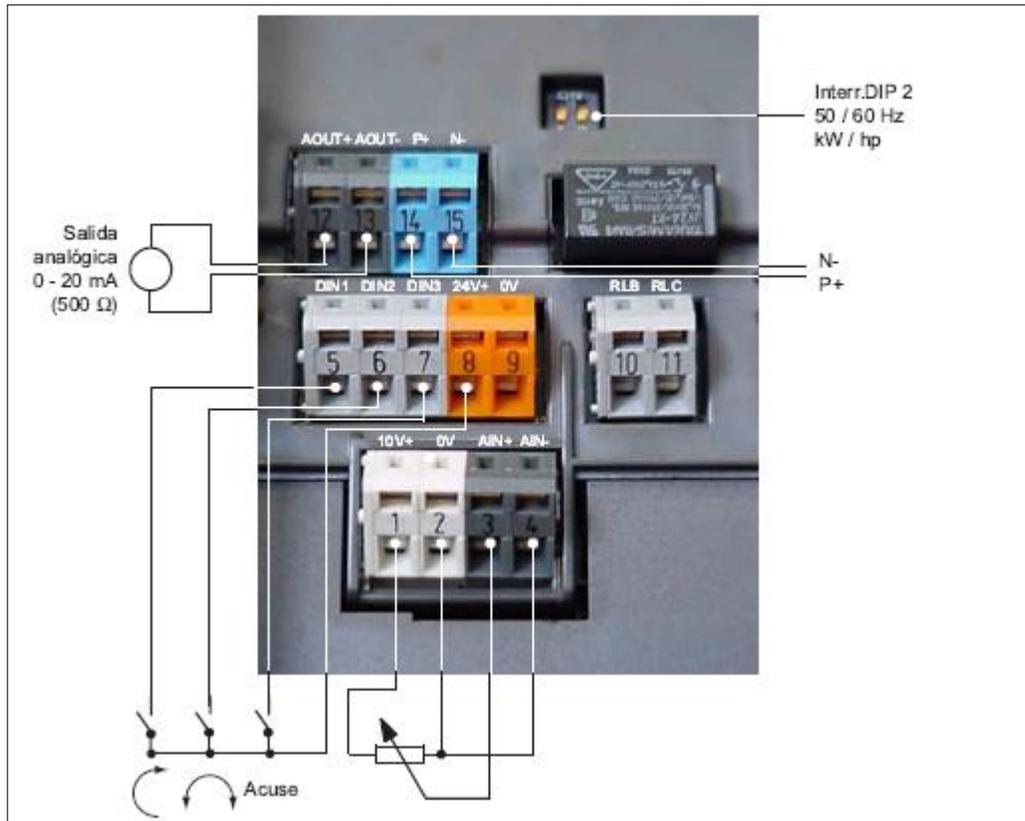


Figura. C.1 Diagrama para controlar en variador de manera externa.

Cuadro C.1. Explicación de conexiones del Diagrama.

Entrada digital	Borne	Parámetro	Operación por defecto
1	5	P0701 = '1'	ON a derechas
2	6	P0702 = '12'	Giro inverso
3	7	P0703 = '9'	Acuse de fallos
Relé de salida	10/11	P0731 = '52.3'	Señalización de fallos
Salida analógica	12/13	P0771 = '21'	Frecuencia de salida

En la figura C.2 se muestra un diagrama de flujo en el que se muestra como poner en marcha el variador de frecuencia.

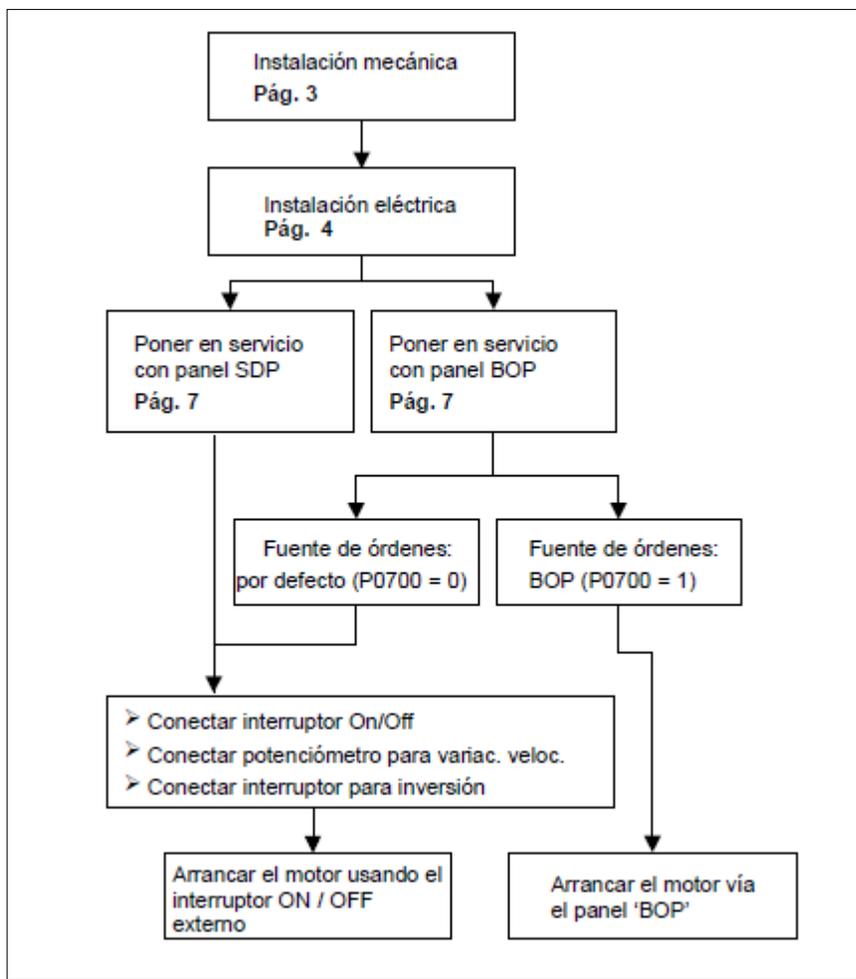


Figura C.2. Diagrama para el funcionamiento del variador

#### C.4. Poner en servicio con el panel SDP

Si el MICROMASTER 420 se pone en servicio usando el panel SDP (Status Display Panel) la aplicación de accionamiento deberá cubrirse con los ajustes por defecto que tiene su MICROMASTER 420.

Conectar un interruptor On/Off en los bornes 5 y 8

- Conectar un interruptor de inversión de sentido en los bornes 6 y 8 (opcional)

- Conectar un interruptor para acuse de fallos en los bornes 7 y 8 (opcional)
- Conectar un display de frecuencia analógico en los bornes 12 y 13 (opcional)
- Conectar un relé de señalización a los bornes 10 y 11 (opcional)
- Conectar en los bornes 1 a 4 (opcional) un potenciómetro de 5,0 kΩ para variación de velocidad.

Con ello el inversor queda listo para arrancar.

### **C.5 Poner en servicio con el panel BOP**

Con el panel BOP (Basic Operator Panel), disponible en calidad de opción, el usuario puede modificar los ajustes por defecto del MM420 para adaptarlo a los requisitos de una aplicación particular. Mediante el BOP es posible acceder a los niveles de juegos de parámetros 1, 2 y 3. El panel BOP tiene las características siguientes:

- Se visualiza, según se desee, la velocidad, la frecuencia, el sentido de giro del motor y la corriente, etc.
- Para mando directo, el panel BOP se monta directamente en el frontal del Convertidor.

Para la “Puesta en servicio rápida” del convertidor es necesario usar la Guía rápida. Para otros métodos de puesta en servicio, ver las Instrucciones de uso o el Manual de referencia.

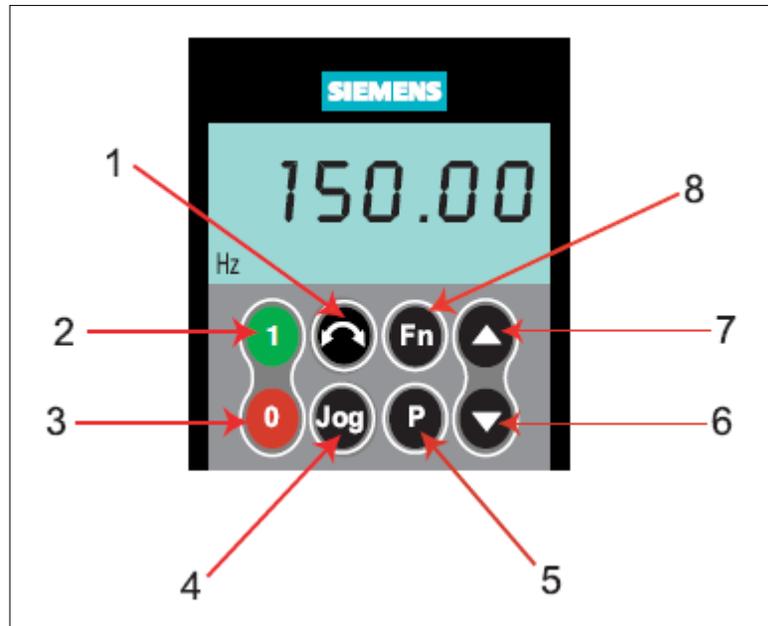


Figura C.3. Esquema del variador de frecuencia.

Tabla C.2 Explicación de los componentes del variador.

1.	Invertir sentido	Cambia el sentido de giro del motor. El sentido negativo se indica por un signo negativo (-) o un punto decimal intermitente.
2.	Marcha	Con este botón se pone en marcha el convertidor. Por defecto, este botón está bloqueado. Para habilitarlo, ajustar P0700 a 1.
3.	Parada	Con este botón se para el motor en el tiempo ajustado en P1121 (tiempo de deceleración).
4.	Jog motor	Mientras no tiene salida el convertidor este botón arranca el motor y lo pone a funcionar a la velocidad Jog preajustada. El convertidor se detiene tan pronto como se suelte el botón.
5.	Acceso a parám.	Pulsado este botón el usuario puede acceder a los parámetros en el nivel de acceso de usuario seleccionado.
6.	Bajar valor	Pulsando este botón se baja el valor visualizado. Para cambiar la consigna (valor deseado) de la frecuencia usando el panel BOP, ajustar P1000 = 1.
7.	Subir valor	Pulsando este botón se sube el valor visualizado. Para cambiar la consigna (valor deseado) de la frecuencia usando el panel BOP, ajustar P1000 = 1.
8.	Funciones	Este botón puede utilizarse para visualizar información adicional. Ver el apt. 51.2 en la pág. 44 de las Instrucciones de uso del MICROMASTER 420.

## C.6. Cambiar parámetros y ajustes usando un panel 'BOP' / 'AOP'

En la figura C.4 se describe la forma de cambiar el parámetro P1082. Usar esta descripción como guía para el ajuste de cualquier parámetro utilizando el panel

BOP

Paso	Resultado en pantalla
1 Pulsar  para acceder a los parámetros	P(1) r0000
2 Pulsar  hasta que se visualice P0010	P(1) P0010
3 Pulsar  para acceder al nivel de valores de P0010	P(1) 0
4 Pulsar  para ajustar P0010 - 1	P(1) 1
5 Pulsar  para salvar y salir del nivel de valores	P(1) P0010
6 Pulsar  hasta que se visualice P1082	P(1) P1082
7 Pulsar  para acceder al nivel de valores de P1082	P(1) 50.00
8 Pulsar  para seleccionar la frecuencia máxima deseada	P(1) 35.00
9 Pulsar  para salvar y salir del nivel de valores	P(1) P1082
10 Pulsar  para volver a P0010	P(1) P0010
11 Pulsar  para acceder al nivel de valores de P0010	P(1) 1
12 Pulsar  para volver a P0010 - 0	P(1) 0
13 Pulsar  para salvar y salir del nivel de valores	P(1) P0010
14 Pulsar  para volver a r0000	P(1) r0000
15 Pulsar  para salir de la parametrización	P(1) 35.00
La visualización en pantalla alternará entre la frecuencia real y la deseada (consigna)	P(1) 00.00

Figura C.3. Descripción cambiar el parámetro P1082

## Anexo D

### D.1. Códigos y diagramas de los programas implementados

En la figura D.1 y D.2 se muestran el cuerpo del programa llamado **Pulsador\_05B** así como su diagrama de su circuito, respectivamente.

```
;
; Cada vez que presiona el pulsador conectado a la línea RA4 conmutará el estado de
; un LED conectado a la línea RB1.
;
; ZONA DE DATOS *****
LIST          P=16F84A
INCLUDE       <P16F84A.INC>
__CONFIG     _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC

CBLOCK 0x0C
ENDC

#DEFINE Pulsador PORTA,4      ; Pulsador conectado a RA4.
#DEFINE LED       PORTB,1     ; Línea donde se conecta el diodo LED.

; ZONA DE CÓDIGOS *****
Inicio      ORG      0          ; El programa comienza en la dirección 0.
            bsf      STATUS,RP0 ; Acceso al Banco 1.
            bsf      Pulsador   ; La línea RA4 se configura como entrada.
            bcf      LED        ; Se configura como salida.
            bcf      STATUS,RP0 ; Acceso al Banco 0.
            bcf      LED        ; En principio diodo LED apagado.
Principal
            btfsc   Pulsador    ; ¿Pulsador presionado?, ¿(Pulsador)=0?
            goto    Fin         ; No. vuelve a leerlo.
            call    Retardo_20ms ; Espera que se estabilicen los niveles de tensión.
            btfsc   Pulsador    ; Comprueba si es un rebote.
            goto    Fin         ; Era un rebote y sale fuera.
            btfsc   LED         ; Testea el último estado del LED.
            goto    EstabaEncendido
EstabaApagado
            bsf      LED        ; Estaba apagado y lo enciende.
            goto    EsperaDejePulsar
EstabaEncendido
            bcf      LED        ; Estaba encendido y lo apaga.
EsperaDejePulsar
            btfss   Pulsador    ; ¿Dejó de pulsar?. ¿(Pulsador)=1?
            goto    EsperaDejePulsar ; No. Espera que deje de pulsar.
Fin
            goto    Principal

INCLUDE <RETARDOS.INC>
END
```

Figura D.1 Cuerpo del programa del Pulsador\_05B

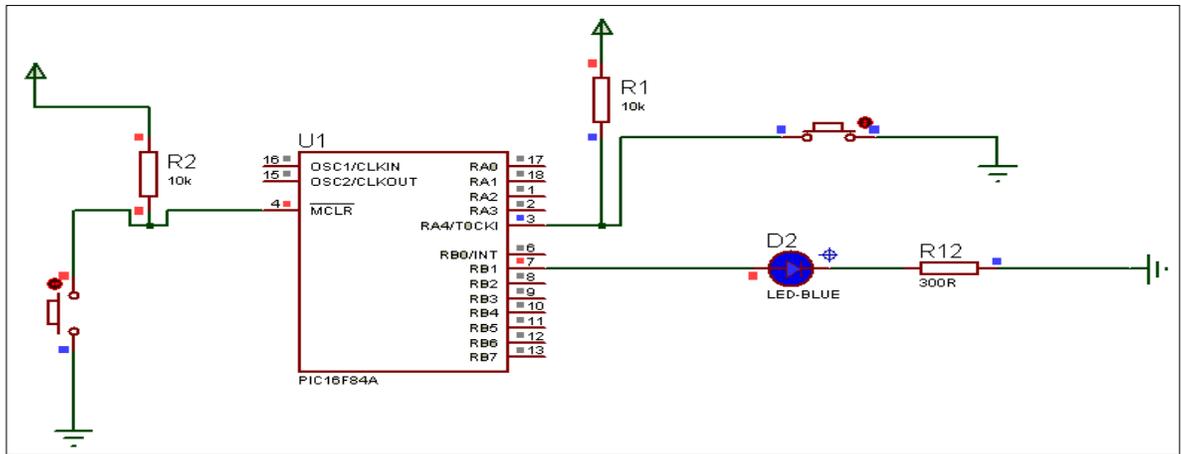


Figura. D.2. Circuito Led\_5

En la figura D.3 se muestra el programa Pulsador\_057 el circuito utilizado fue el mismo que se muestra en la figura anterior.

```

; Cada vez que presiona el pulsador conectado a la línea RA4 conmutará el estado de
; un LED conectado a la línea RB1. La complementación se realiza con ayuda de una
; función XOR.
; ZONA DE DATOS *****
LIST          P=16F84A
INCLUDE      <P16F84A.INC>
__CONFIG    _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC

CBLOCK 0x0C
ENDC

#DEFINE Pulsador    PORTA,4      ; Pulsador conectado a RA4.
; ZONA DE CÓDIGOS *****
Inicio
ORG 0          ; El programa comienza en la dirección 0.
    bsf        STATUS,RP0        ; Acceso al Banco 1.
    bsf        Pulsador          ; La línea RA4 como entrada.
    clrf      PORTB             ; Se configura como salida.
    bcf        STATUS,RP0        ; Acceso al Banco 0.
    clrf      PORTB             ; En principio diodo LED apagado.
Principal
    btfsc     Pulsador           ; ¿Pulsador presionado?. ¿(Pulsador)=0?
    goto     Fin                ; No. vuelve a leerlo.
    call     Retardo_20ms        ; Espera que se estabilicen los niveles de tensión.
    btfsc     Pulsador           ; Comprueba si es un rebote.
    goto     Fin                ; Era un rebote y sale fuera.
    movlw    b'00000010'        ; Se hace una XOR para complementar.
    xorwf    PORTB,F
EsperaDejePulsar
    btfss     Pulsador           ; ¿Dejó de pulsar?. ¿(Pulsador)=1?
    goto     EsperaDejePulsar   ; No. Espera que deje de pulsar.
Fin
    goto     Principal

INCLUDE <RETARDOS.INC>
END

```

Figura D.3. Cuerpo del Programa Pulsador\_057

Programa Rebotos\_Final este programa enciende y apaga el led sin rebotes, figura D.4, y el diagrama utilizado se muestra en la figura D.5.

```

; El led se enciende cuando pulse RB4
;
;
; ZONA DE DATOS *****
LIST          P=16F84A
INCLUDE       <P16F84A.INC>
__CONFIG      _CP_OFF & _WDT_OFF & _PWRTE_ON & _XT_OSC

CBLOCK       0x0C
ENDC

#DEFINE      EntradaRB4  PORTB,4
#DEFINE      Led         PORTB,2

; ZONA DE CÓDIGOS *****

ORG         0
goto       Inicio
ORG         4
goto       ServicioInterrupcion

Inicio
    bsf          STATUS,RP0
    bsf          EntradaRB4
    bcf          Led
    bcf          OPTION_REG,NOT_RBPU
    bcf          STATUS,RP0
    bcf          Led ; En principio, led apagado.
    movlw       b'10001000'
    movwf       INTCON

Principal
    sleep
    goto        Principal

; Subrutina "servicioInterrupcion" -----
;
; ServicioInterrupcion
    call        Retardo_20ms
    btfss      EntradaRB4 ; ¿Está presionado el pulsador de RB4?
    call        Encendido

EsperaDejePulsar ; Espera a que desaparezcan las señales de entrada.
    btfss      EntradaRB4
    goto       EsperaDejePulsar
    bcf        INTCON,RBIF
    retfie

; Subrutinas "PitidoCorto" y "PitidoLargo" -----
;
; Encendido
    bsf          Led
    btfss      EntradaRB4
    goto       Encendido
    bcf          Led
    bcf          INTCON,RBIF
    return

INCLUDE      <RETARDOS.INC>

```

Figura D.4 Cuerpo del programa Rebotos\_Final

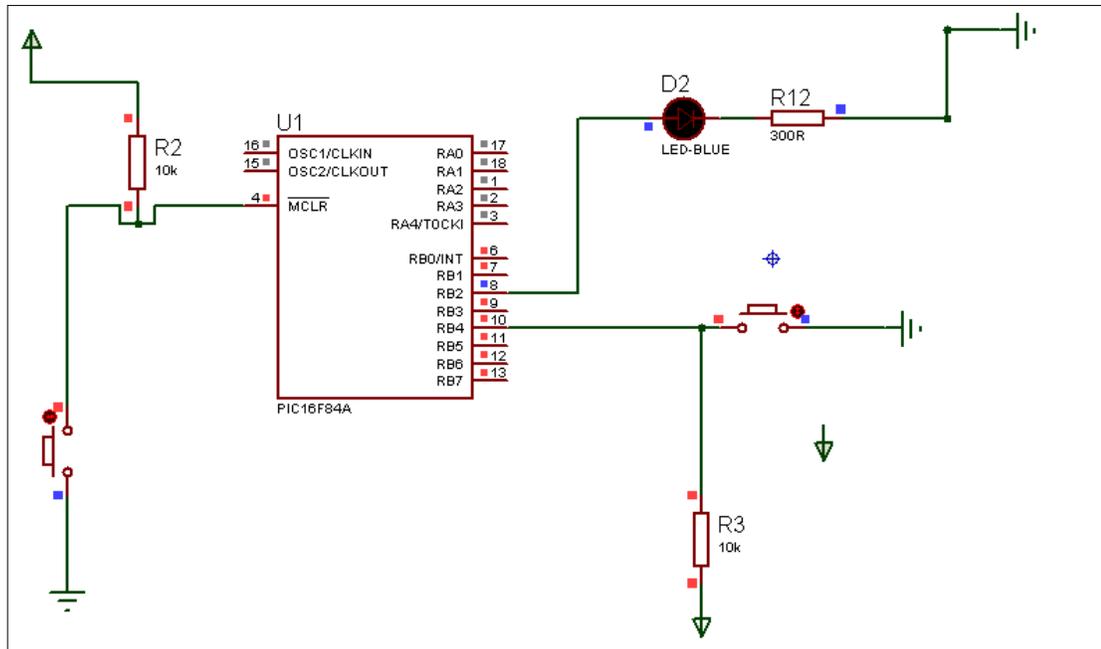


Figura. D.5. Circuito para el Pulsador\_Final.

El siguiente programa nos genera las frecuencias, lleva por nombre **Int\_Cuadradas\_01**

```

;***** Int_Cuadradas_01.asm *****
;
; Por la línea 3 del puerto B se genera una onda cuadrada. La frecuencia de ;la
onda cuadrada cambia
; mediante activación del pulsador conectado al pin 7 del puerto B, de la
;siguiente forma:
;
;
; PULSACIÓN FRECUENCIA SEMIPERODO
; -----
; (Inicial) 10 kHz 50 µs. = 1 x 50 µs
; Primera 5 kHz 100 µs. = 2 x 50 µs
; Segunda 2 kHz 250 µs. = 5 x 50 µs
; Tercera 1 kHz 500 µs. = 10 x 50 µs
; Cuarta 500 Hz 1000 µs. = 20 x 50 µs
; Quinta 200 Hz 2500 µs. = 50 x 50 µs
; Sexta 100 Hz 5000 µs. = 100 x 50 µs
; Septima 50 Hz 10000 µs. = 200 x 50 µs
;
; Al conectarlo por primera vez se genera una frecuencia de 10 kHz, al activar el
; pulsador cambia a 5 kHz, al actuar una segunda vez cambia a 2 kHz, y así
; sucesivamente.

```

```

;
; El módulo LCD visualizará la frecuencia generada. A la línea de salida se ;
puede conectar
; un altavoz que producirá un pitido.

```

```

; ZONA DE DATOS

```

```

*****

```

```

;
    ORG 0
    goto Inicio
    ORG 4
    goto ServicioInterrupcion

```

```

; Subrutina "CargaContador" -----

```

```

CargaContador

```

```

    addwf PCL,F
    retlw .1    ; Semiperiodo 1 x 50 = 50 µs, frecuencia = 10 kHz.
    retlw .2    ; Semiperiodo 2 x 50 = 100 µs, frecuencia = 5 kHz.
    retlw .5    ; Semiperiodo 5 x 50 = 250 µs, frecuencia = 2 kHz.
    retlw .10   ; Semiperiodo 10 x 50 = 500 µs, frecuencia = 1 kHz.
    retlw .20   ; Semiperiodo 20 x 50 = 1000 µs, frecuencia = 500 Hz.
    retlw .50   ; Semiperiodo 50 x 50 = 2500 µs, frecuencia = 200 Hz.
    retlw .100  ; Semiperiodo 100 x 50 = 5000 µs, frecuencia = 100 Hz.
    retlw .200  ; Semiperiodo 200 x 50 = 10000 µs, frecuencia = 50 Hz.

```

```

; Subrutina "CargaMensaje" -----

```

```

CargaMensaje

```

```

    addwf PCL,F
    retlw Mensaje10kHz
    retlw Mensaje5kHz
    retlw Mensaje2kHz
    retlw Mensaje1kHz
    retlw Mensaje500Hz
    retlw Mensaje200Hz
    retlw Mensaje100Hz
    retlw Mensaje50Hz

```

```

; Subrutina "Mensajes" -----

```

```

;
Mensajes
    addwf PCL,F
MensajeFrecuencia
    DT "Frec.: ", 0x00
Mensaje10kHz
    DT "10 kHz. ", 0x00
Mensaje5kHz

```

```

    DT "5 kHz. ", 0x00
Mensaje2kHz
    DT "2 kHz. ", 0x00
Mensaje1kHz
    DT "1 kHz. ", 0x00
Mensaje500Hz
    DT "500 Hz. ", 0x00
Mensaje200Hz
    DT "200 Hz. ", 0x00
Mensaje100Hz
    DT "100 Hz. ", 0x00
Mensaje50Hz
    DT "50 Hz. ", 0x00

```

; Programa Principal -----

Inicio

```

    call    LCD_Inicializa
    bsf    STATUS,RP0
    bcf    Salida
    bsf    Pulsador
    movlw  b'00001000'           ; TMR0 sin prescaler.
    movwf  OPTION_REG
    bcf    STATUS,RP0
    call   EstadoInicial       ; Visualiza la frecuencia inicial.
    movlw  TMR0_Carga50us     ; Carga el TMR0.
    movwf  TMR0
    movlw  b'10101000'
    movwf  INTCON             ; Activa interrupciones del TMR0, RBI y general.

```

Principal

```

    goto  Principal

```

; Subrutina "ServicioInterrupcion" -----

```

;
; Detecta qué ha producido la interrupción y ejecuta la subrutina
; correspondiente.

```

ServicioInterrupcion

```

    btfsc  INTCON,T0IF         ; ¿Interrupción por desbordamiento del TMR0?.
    call   Timer0_Interrupcion
    btfsc  INTCON,RBIF        ; ¿Interrupción por cambio en el Puerto B?.
    call   Pulsador_Interrupcion
    bcf    INTCON,T0IF       ; Repone flag del TMR0.
    bcf    INTCON,RBIF      ; Repone flag del RBI.
    retfie

```

; Subrutina "Pulsador\_Interrupcion" -----

```

;
; Subrutina de atención a la interrupción por cambio en la línea RB7 donde se
;ha conectado un pulsador.
; Incrementa el registro (ApuntadorFrecuencia) desde b'00000000' (que
;corresponde a una frecuencia de 10 kHz) hasta b'00000111' (que ;corresponde
a una frecuencia de 50 Hz),
; según la tabla especificada en el enunciado del ejercicio.
;

```

```

Pulsador_Interrupcion

```

```

    call    Retardo_20ms
    btfsc   Pulsador
    goto    Fin_PulsadorInterrupcion
    incf    ApuntadorFrecuencia,F    ; Apunta a la siguiente frecuencia.
    movlw   NUM_FRECUENCIA          ; Va a comprobar al máximo.
    subwf   ApuntadorFrecuencia,W    ; (W)=(ApuntadorFrecuencia)-NUM_FRECUENCIA
    btfsc   STATUS,C                ; ¿Ha llegado a su máximo?

```

```

EstadoInicial

```

```

    clrf    ApuntadorFrecuencia      ; Si llega al máximo lo inicializa.
    movf    ApuntadorFrecuencia,W    ; Va a cargar el valor del factor de
    call    CargaContador            ; multiplicación del semiperiodo según la
    movwf   Semiperiodo              ; tabla.
    movwf   Timer0_ContadorA
    call    LCD_Linea1                ; Visualiza la frecuencia seleccionada.
    movlw   MensajeFrecuencia
    call    LCD_Mensaje
    movf    ApuntadorFrecuencia,W
    call    CargaMensaje
    call    LCD_Mensaje

```

```

EsperaDejePulsar

```

```

    btfss   Pulsador
    goto    EsperaDejePulsar

```

```

Fin_PulsadorInterrupcion

```

```

    return

```

```

; Subrutina "Timer0_Interrupcion" -----
;

```

```

    CBLOCK
    Timer0_ContadorA
    ENDC

```

```

Timer0_Interrupcion

```

```

    movlw   MR0_Carga50us
    movwf   TMR0                      ; Recarga el TMR0.
    decfsz  Timer0_ContadorA,F        ; Decrementa el contador.
    goto    Fin_Timer0_Interrupcion
    movf    Semiperiodo,W            ; Repone el contador nuevamente.
    movwf   Timer0_ContadorA

```

```

    btfsc      Salida      ; Testea el último estado de la salida.
    goto      EstabaAlto
EstabaBajo
    bsf       Salida      ; Estaba bajo y lo pasa a alto.
    goto      Fin_Timer0_Interrupcion
EstabaAlto
    bcf       Salida      ; Estaba alto y lo pasa a bajo.
Fin_Timer0_Interrupcion
    return

INCLUDE <LCD_MENS.INC>
INCLUDE <LCD_4BIT.INC>
INCLUDE <RETARDOS.INC>
END

```

En la figura D.6 se muestra el circuito para habilitar el programa que envía las frecuencias.

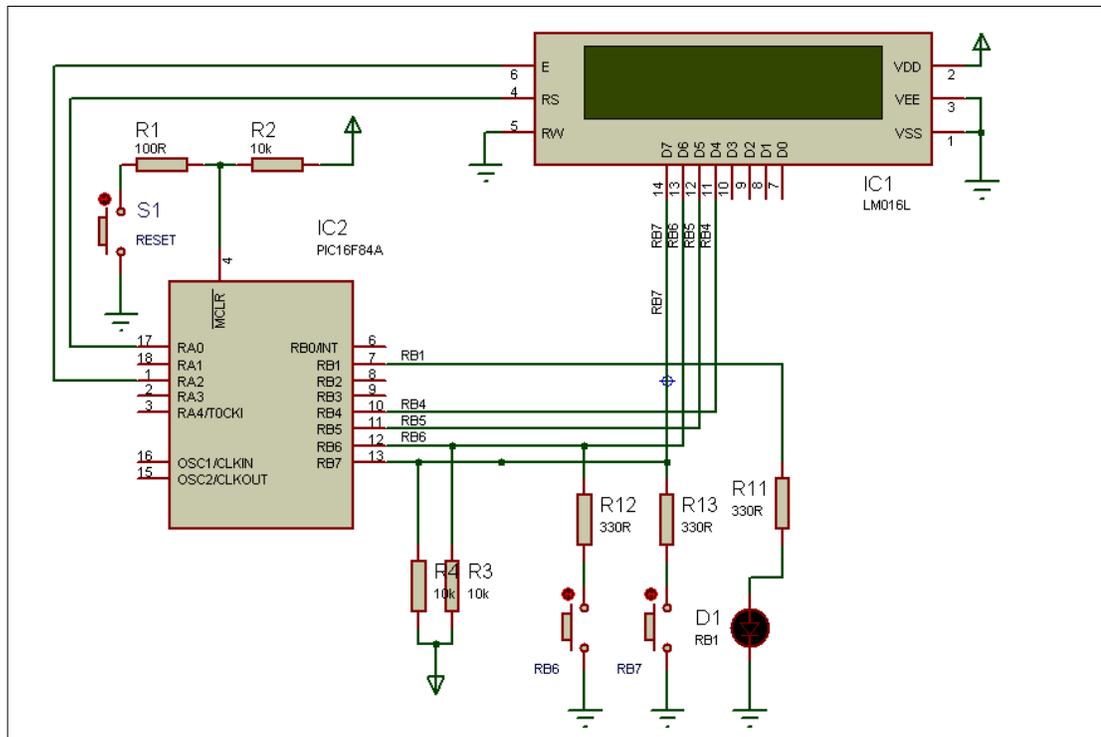


Figura D.6 Circuito para frecuencias del programa Int\_Cuadradas\_01

## Anexo E

### E.1 Simulación con MPLAB & Proteus 7.0

La simulación de los programas y los circuitos es muy importante ya que podemos verificar su funcionamiento correcto, para después hacerlo de manera real sin tener problemas para su ejecución.

Para realizar la simulación cargamos nuestro programa MPLAB, abrimos nuestro archivo en el cual se encuentra el programa que vamos a simular, una vez que lo tenemos cargado lo ensamblamos para verificar si no existe algún error, figura E.1

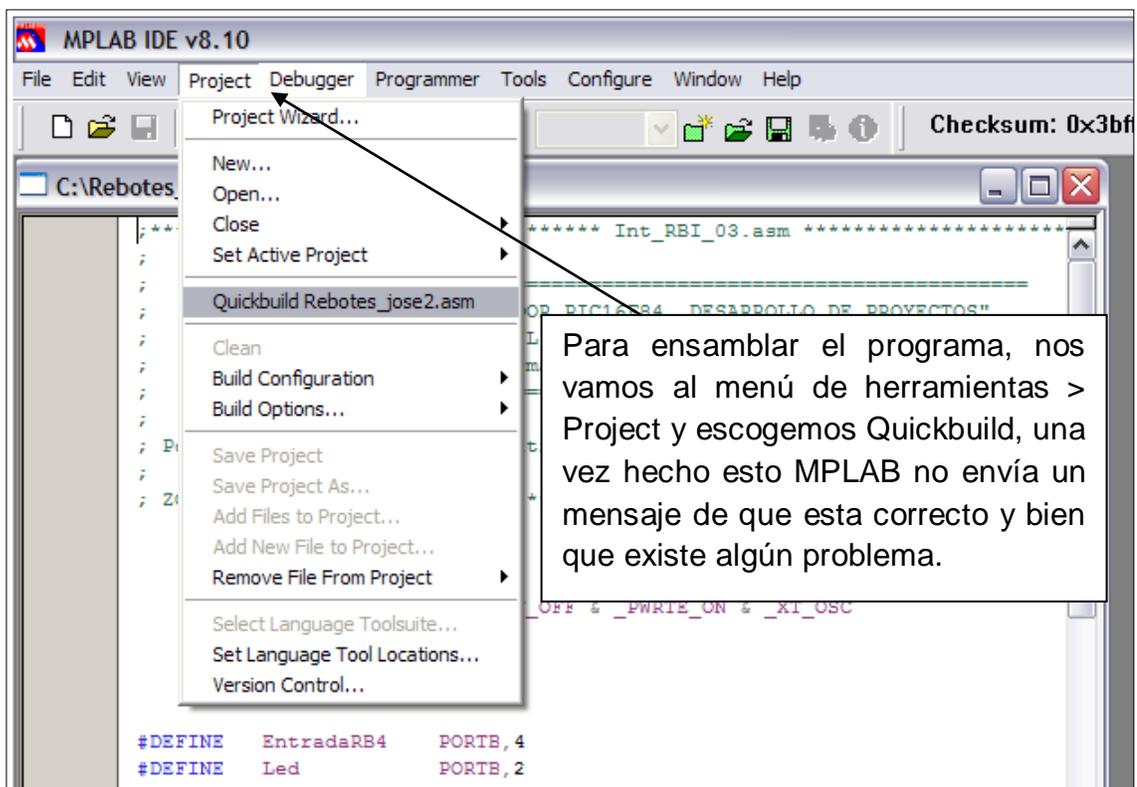


Figura E.1. Ensamblar programas.

Cuando verificamos que funciona correctamente nuestro programa, es momento de cargar el circuito, esto lo hacemos con la herramienta de Proteus desde MPLAB esto lo hacemos de la siguiente manera:

1. Del menú herramientas > Proteus VSM, figura E.2

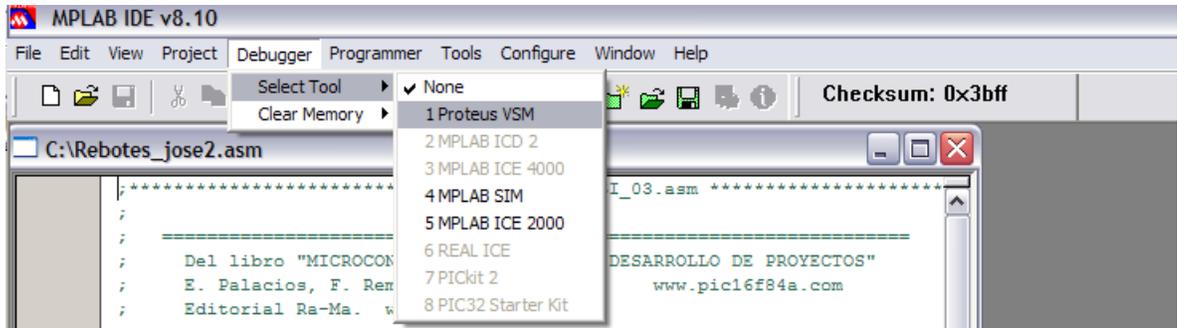


Figura E.2 Habilitar Proteus para simulación.

2. Una vez terminado el procedimiento anterior nos abre la siguiente ventana la cual se muestra en la figura E.3.

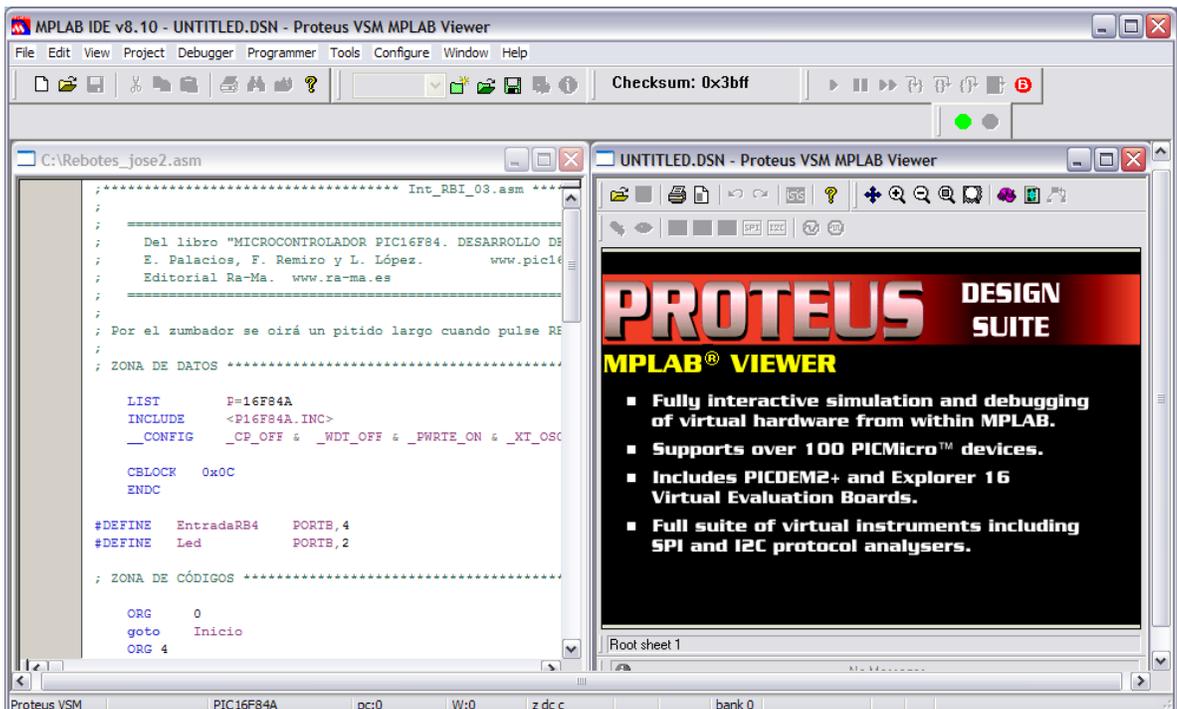


Figura E.3. Ejecución del programa ISIS en MPLAB.

3. Cuando tenemos habilitado el ISIS abrimos nuestro circuito, figura E.4

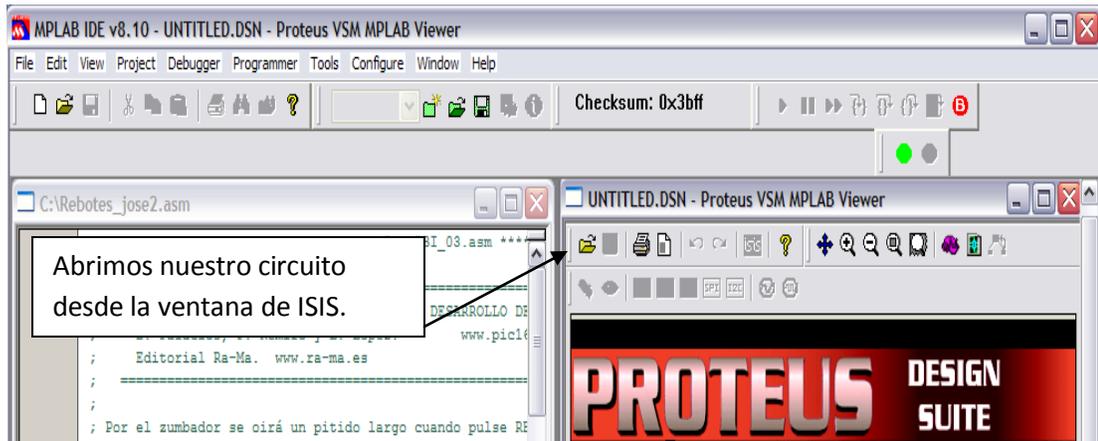


Figura. E.4. Habilitar circuito en ISIS

4. En la figura E.5 se muestra el circuito y el programa listos para la simulación.

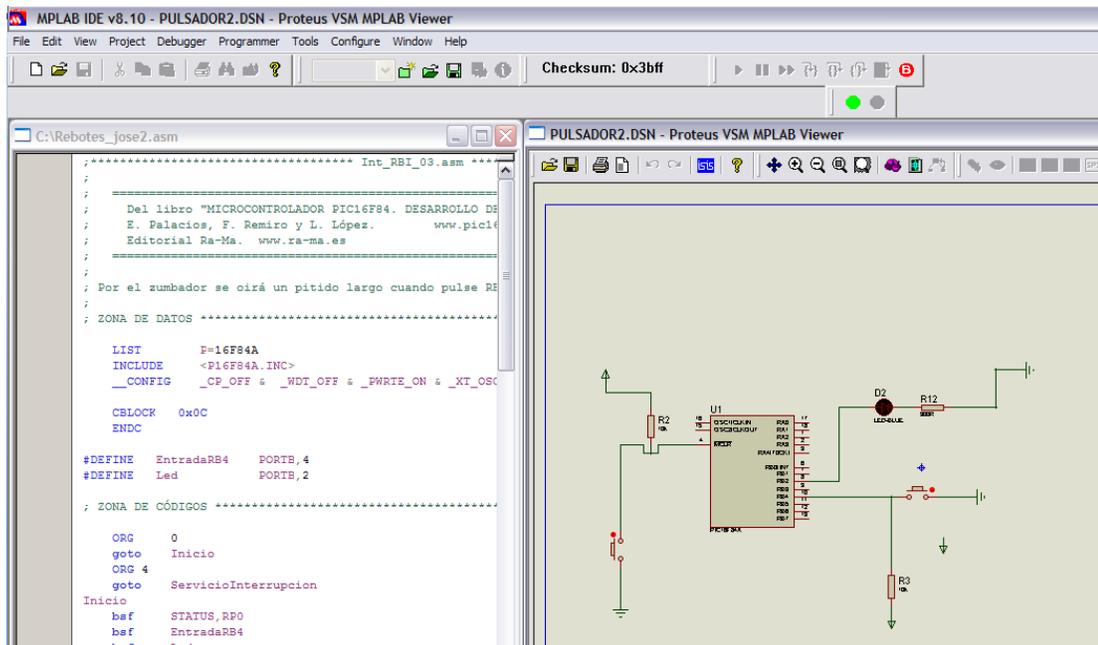


Figura E.5. Programa circuitos para simulación.

5. Para ejecutar la simulación damos click en Start Simulation, después en ejecutamos la simulación. Figura E.6



Figura E.6. Inicio de simulación.

6. Para terminar simulación damos click stop simulation, figura E.7.

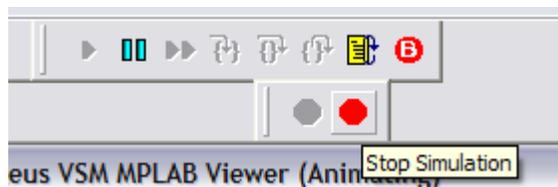


Figura E.7. Terminar simulación.