



**UNIVERSIDAD NACIONAL AUTÓNOMA  
DE  
MÉXICO**

**FACULTAD DE INGENIERÍA**

**Sistema electrónico para la calibración  
de nivel de fluidos en tanques**

**T E S I S**

QUE PARA OBTENER EL TÍTULO DE:

**INGENIERO EN ELÉCTRICA ELECTRÓNICA**

P R E S E N T A

**CRUZ PLACIDO ARELI**

DIRIGIDA POR

**BENJAMÍN VALERA OROZCO**



CIUDAD UNIVERSITARIA  
Enero de 2016



# Agradecimientos

Al M. en I. Rigoberto Nava Sandoval por el apoyo recibido para la elaboración del proyecto.

Al CCADET, que gracias a sus instalaciones, se logró desarrollar el prototipo funcional.

Especialmente a mi tutor de tesis Benjamín Valera Orozco quien desde que tuve el gusto de conocer, me brindó su confianza, apoyo y orientación que fueron esenciales para la realización de ésta Tesis. Creando un gran respeto y admiración de mí parte.

A mis padres Raúl Cruz Aguilar y Estela Maura Placido Tapia que siempre me han apoyado en cada etapa de mi vida, siendo una gran guía para la superación de los obstáculos que se han presentado y que gracias a su apoyo incondicional he logrado terminar otro ciclo más de mi vida, llevando conmigo una gran herencia.

A mi hermana Carolina Geraldin Cruz Placido y a mi gran compañero incondicional Jesús Alberto Camacho Flores que me dieron la fortaleza con su motivación y apoyo para continuar con mi objetivo, convirtiendo esta gran experiencia una de las más gratas.

Por último, a mis tutores Rocío Pantoja y Gregorio Lemus del plan de becarios en DGTIC, que con sus enseñanzas me han brindado un gran crecimiento profesional y me dieron un gran apoyo para poder realizar mi tesis.



# Índice

<b>Resumen de la tesis</b>	<b>1</b>
<b>Introducción</b>	<b>3</b>
Definición del problema	3
Antecedentes	3
Descripción del problema a resolver	3
Relevancia y justificación	4
Alcance y limitaciones	4
Relación con otras áreas	4
Método	4
Resultados esperados	5
<b>1. Conceptos básicos</b>	<b>7</b>
1.1. Medición de nivel de fluidos en tanques	7
1.2. Control de posición de motores	10
1.3. Microcontrolador PIC18F4550	12
1.3.1. Especificaciones generales	12
1.3.2. Herramientas de desarrollo	14
1.4. Desarrollo de aplicaciones en Visual C++	21
1.5. Dispositivos electrónicos empleados	24
1.5.1. Compuerta digital 74HC04	24
1.5.2. Transistor NPN 2N3904	25
1.5.3. Relevador G5LE-1A4-DC12	25
1.5.4. Reguladores de voltaje LM7818 Y LM7805	26
1.5.5. Transistor TIP42C	27
1.5.6. Interruptor óptico H21A1	27
1.5.7. Puente rectificador RS607	28

---

1.5.8. Motor Copley Control SSC 23SMDC-LCSS	28
1.5.9. Circuito de interfaz entre codificador y microprocesador	29
1.5.10. Impulsor de motor de corriente directa	32
<b>2. Descripción del proyecto</b>	<b>37</b>
2.1. Esquema general	37
2.2. Sistema electrónico	41
2.2.1. Etapa de control	42
2.2.2. Etapa para los sensores ópticos	43
2.2.3. Etapa de registro de posición	45
2.2.4. Etapa impulsora de corriente directa	49
2.2.5. Etapa impulsora CD/CA	51
2.2.6. Etapa de alimentación del circuito	52
2.2.7. Algoritmo de medición, control y comunicación	58
2.3. Software de operación	64
2.3.1. Descripción general del proyecto	65
2.3.2. Descripción en el ámbito del operador	66
2.3.3. Descripción en el ámbito del programador	68
<b>3. Resultados</b>	<b>75</b>
3.1. Resultados	75
3.1.1. Desempeño en posición de los motores	77
3.1.2. Indicador de nivel	80
3.1.1. Bomba y válvula	82
3.1.2. Sensores de posición inicial	84
3.2. Discusión	85
3.3. Trabajo a futuro	86
<b>Conclusiones</b>	<b>87</b>
<b>Bibliografía</b>	<b>89</b>
<b>Anexos</b>	<b>91</b>
A. Diagrama esquemático	91
B. Circuito impreso	93
C. Lista de partes	94

# Resumen de la tesis

En el presente proyecto de tesis se propone desarrollar un sistema electrónico que permita la adquisición de la señal de un sensor de nivel colocado dentro de un tanque de prueba cuando éste se encuentra en movimiento, para posteriormente, determinar su rango de operación. Adicionalmente, el sistema electrónico impulsará el movimiento del tanque mediante el control de posición de dos motores eléctricos como también el llenado y vaciado del fluido de prueba en el tanque.

Este trabajo está dividido en:

La sección de introducción que plantea el problema enfrentado en este trabajo de tesis, el método empleado para resolverlo y expone los resultados esperados.

El capítulo uno presenta los conceptos básicos que se utilizaron en el desarrollo del proyecto de tesis. Se presenta de manera resumida la lista de dispositivos electrónicos empleados en el desarrollo y construcción del sistema electrónico.

En el capítulo dos se describe detalladamente el desarrollo del sistema electrónico y su componente software.

El capítulo tres contiene los resultados, conclusiones y trabajo a futuro.

Finalmente los anexos contienen información detallada acerca de los recursos desarrollados como parte del presente proyecto de tesis.





# **Introducción**

## **Definición del problema**

La medición de nivel de fluido en tanques se utiliza en diversos ámbitos donde es preciso conocer la cantidad de fluido que contiene un tanque. El nivel alcanzado se utiliza para calcular el volumen del líquido en el tanque.

Los medidores comúnmente usados en la vida cotidiana son los que realizan la medida de manera directa, un ejemplo es la cinta y plomada, que consiste de un peso (plomada) sostenido por una cinta graduada que descenderá al final del contenedor dando así la medida del fluido. Pero la desventaja de estos instrumentos es que se debe tener un fluido sin movimiento y de baja viscosidad.

Por lo contrario, en las Industrias se necesitan medidores de nivel que puedan dar una medición lo más exacto posible a pesar del tipo de fluido que se contenga en el tanque, la temperatura y las variaciones del movimiento, por lo que se requiere de sistemas electrónicos más sofisticados.

## **Antecedentes**

En el Laboratorio de Ingeniería de Proceso del CCADET UNAM se desarrollan diversos proyectos en el área de reología. La planta académica identificó que la medición de nivel en tanques móviles es afectada por diversos factores. De esta manera, es necesario desarrollar y construir instrumentos electrónicos que permitan estudiar el problema dado que estos no están disponibles comercialmente.

## **Descripción del problema a resolver**

En el presente proyecto de tesis se propone desarrollar un sistema electrónico para controlar un dispositivo mecánico con dos grados de libertad sobre el cual se montarán tanques con cierta cantidad de fluido. El dispositivo mecánico desplaza el tanque en dos ejes con la finalidad de analizar el comportamiento del sensor de nivel colocado dentro del tanque para su caracterización y determinar su rango de operación. Se requiere que el sistema electrónico controle la posición de los dos grados de libertad mientras

se captura la lectura del nivel. Adicionalmente, el sistema electrónico debe permitir el llenado y vaciado del tanque con algún fluido de prueba. La operación del equipo se debe realizar desde una computadora para el posterior procesamiento de los datos de posición y nivel.

### Relevancia y justificación

La relevancia del proyecto radica en el fomento a la creación de infraestructura de desarrollo propio que puede ser utilizada con fines académicos, de investigación y que eventualmente puede transferirse a la industria.

### Alcance y limitaciones

El alcance del presente proyecto es la creación de la componente electrónica e interfaz de operación para la computadora personal del prototipo funcional. En una segunda etapa se deberá contemplar el desarrollo y construcción de la componente mecánica.

La principal limitación del prototipo a desarrollar, radica en que se prescinde de la componente mecánica. No obstante, con las especificaciones del grupo de académicos del CCADET UNAM se puede trabajar de forma aislada en la componente electrónica e interfaz de operación.

### Relación con otras áreas

Existe una gran relación con las áreas de sistemas de control digital, sistemas electrónicos analógicos, electrónica de potencia, microcontroladores, instrumentación y medición.

### Método

En este trabajo se propone desarrollar las componentes electrónica y software de operación para el equipo utilizado en la calibración de nivel a partir del esquema general mostrado en la figura 1.

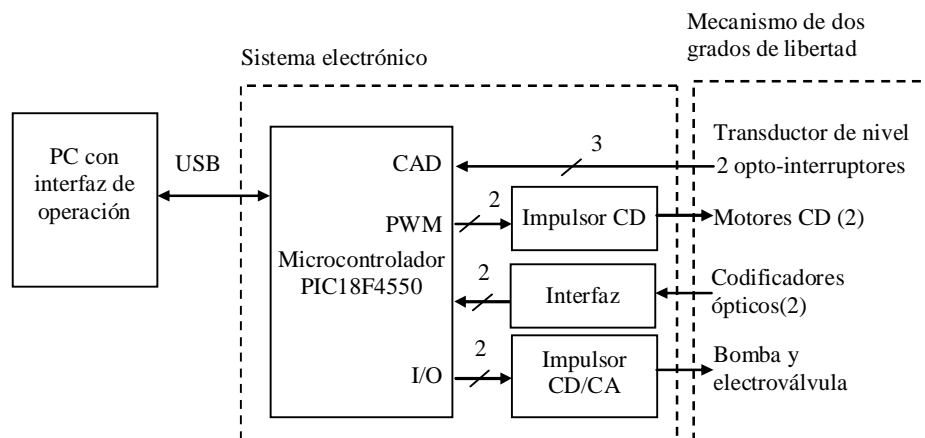


Figura 1. Esquema general del sistema.

El sistema electrónico tiene como base un microcontrolador PIC18F4550 de 8 bits con las siguientes características relevantes para la presente aplicación: convertidor analógico a digital (CAD) de 10 bits, modulación por ancho de pulso (PWM) de 10 bits en dos canales y salidas y entradas digitales (I/O) de propósito general.

En el esquema de la figura 1, el microcontrolador recibe la señal del transductor de nivel en su canal de conversión analógica a digital, CAD, para registrar esta variable. Las señales de referencia de posición o “set point” son especificadas por el operador en la interfaz de la computadora para enviarlas al sistema electrónico en donde se cierra el lazo de control de posición de los dos grados de libertad. Cada uno de los grados de libertad en el mecanismo es impulsado por motores de corriente directa (CD) conectados al sistema electrónico. El microcontrolador genera las señales moduladas por ancho del pulso, PWM, que son aplicadas a los motores mediante circuitos impulsores de corriente directa. La componente de directa en las señales de PWM es calculada en el algoritmo de control de posición con base en la posición registrada mediante codificadores ópticos y circuito de interfaz. Los sensores detectan la posición inicial de la plataforma para que los motores ajusten una posición requerida. El llenado y vaciado del tanque se consigue al impulsar el conjunto de bomba hidráulica y electroválvula desde la interfaz de operación y sistema electrónico con las terminales I/O. El diseño contempla que la bomba y electroválvula sean de corriente directa, CD, o de corriente alterna, CA. La interfaz de operación es un programa desarrollado específicamente para esta aplicación utilizando Visual C++.

## **Resultados esperados**

Desarrollar un sistema electrónico y software de operación para controlar un dispositivo mecánico de dos grados de libertad utilizado en la calibración de nivel de fluido en tanques.



# Capítulo 1

## Conceptos básicos

En este capítulo se presentan algunos conceptos básicos que fueron utilizados en el desarrollo del presente proyecto.

### 1.1. Medición de nivel de fluidos en tanques

El nivel de un fluido se puede definir como la altura de un fluido o sólido que está contenido en un recipiente con respecto a un valor de referencia (fondo del recipiente y superficie del líquido).

En la Industria, la medición de nivel de un tanque es esencial para determinar la cantidad de fluido que está contenido y así poder realizar un control o procesos para diversos fines; esto se puede hacer por medio de una variedad de dispositivos, algunos de ellos usan la presión o el peso del fluido para determinar su nivel.

Los métodos utilizados para ésta medición básicamente pueden ser clasificados en dos:

- a) Métodos de medición directa. Los cuales trabajan midiendo la altura del fluido sobre una línea de referencia (regla graduada, varilla con gancho, cinta métrica metálica graduada con plomada, indicador de cristal).
- b) Métodos de medición indirecta. Mide otros efectos que cambian con el nivel del líquido como el aprovechamiento de la presión hidrostática (manométrico, caja de diafragma, rampa de aire, presión diferencial) o por las características eléctricas del líquido.

Los diversos tipos de medidores de nivel se muestran en la figura 1.1.

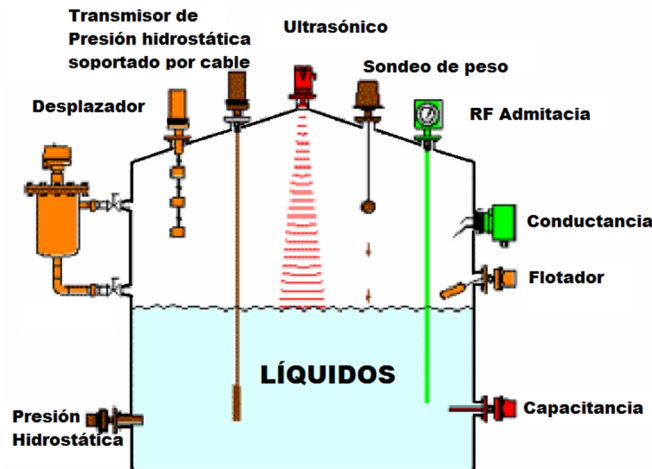


Figura 1.1. Tipos de medidores de nivel.

Todos estos métodos de medición de nivel requieren que los líquidos sean homogéneos y con densidad constante debido a que algunas propiedades del líquido tales como la viscosidad del fluido, temperatura, densidad, corrosión, movimiento, etc. pueden afectar la lectura y dar un valor erróneo.

Debido a la popularidad de los medidores de nivel utilizados en los automóviles para determinar la cantidad de gasolina almacenada en el tanque, a continuación se proporciona una breve explicación de su principio de operación.

La mayoría de los sistemas indicadores de nivel de combustible en los vehículos están formados por los siguientes elementos:

1. Un sensor de nivel que da una salida proporcional al nivel del depósito.
2. Un elemento indicador en el tablero que mide la magnitud de la salida del sensor y tiene su escala calibrada en valores de nivel.

Todos los combustibles utilizados en los motores de los vehículos son líquidos, como tales, forman olas dentro del depósito durante la circulación del coche debido a las aceleraciones y frenadas, así como al empuje lateral en las curvas. Estas olas hacen que sea difícil determinar el nivel del combustible real en un instante de tiempo cuando el coche circula, si no se dispone de un sistema adecuado, la aguja del indicador estaría constantemente moviéndose en la escala, y la determinación del nivel verdadero por el conductor sería muy imprecisa. Para minimizar este efecto los sistemas de medición de combustible usan ciertos artificios que casi eliminan el problema del cambio de nivel debido a las olas, entre ellos están:

1. Utilización de tabiques divisorios "rompe olas" dentro del depósito.

2. Colocación del sensor en la zona central del depósito donde el efecto de incremento del nivel por las olas es menor.
3. Utilización de sensores de nivel con movimiento amortiguado o demorado para que no reaccionen con rapidez y no copien el perfil de las olas.
4. Utilización de indicadores en el tablero de respuesta lenta.

El sensor de la figura 1.2 muestra un esquema de cómo está construido el sensor de nivel que se coloca en el tanque. El cuerpo metálico del sensor está montado en la superficie del depósito y tiene un flotador en el extremo de una palanca giratoria cuya posición dependerá del nivel del líquido. El otro extremo de la palanca del flotador tiene un contacto deslizante sobre una resistencia eléctrica que se mueve en sincronización con él, de manera que la posición del contacto sobre la resistencia también dependerá del nivel del líquido en el depósito [Bruzos, 2015].

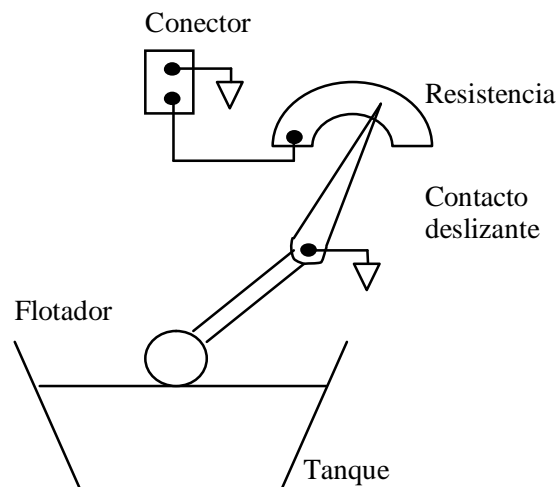


Figura 1.2. Sensor de nivel automotriz.

Esta resistencia se conecta en serie con el indicador del tablero, de forma tal que el circuito se cierra a tierra por la vía resistencia => palanca de flotador => cuerpo del sensor => cuerpo del depósito.

De todo esto se desprende que para cada valor del nivel en el depósito, corresponderá un valor de resistencia en serie con el indicador del tablero y por tanto una indicación de la aguja en la escala.

Como indicador del tablero son comunes los indicadores de lámina bimetálica o el de cuadros cruzados.

La utilización cada vez más común de microprocesadores en la administración del vehículo hace que en la actualidad, en muchos coches, la señal de sensor sea previamente procesada electrónicamente antes de ser enviada al indicador.

## 1.2. Control de posición de motores

Un sistema de control está formado por subsistemas y procesos (o plantas) unidos con el fin de controlar las salidas de los procesos, es decir, produce una salida o respuesta para una entrada o estímulo dado, como se muestra en la figura 1.3.

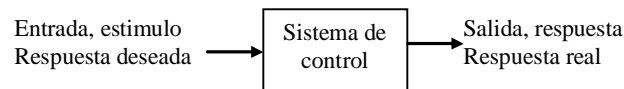


Figura 1.3. Sistema de control.

Hay dos tipos de sistemas de control, el sistema en lazo abierto y el sistema en lazo cerrado (realimentado, ver figura 1.4). Las desventajas del primer sistema es su incapacidad de corregir las perturbaciones que se presenten, mientras que el sistema realimentado compensa perturbaciones al medir la respuesta de salida, alimentando esa medida a una trayectoria de realimentación y comparando esa respuesta con la entrada en el punto suma. Si hay alguna diferencia entre las dos respuestas el sistema tratará de corregir el error, pero si no hay diferencia se mantendrá en ese mismo resultado ya que es la respuesta deseada, por lo tanto el segundo sistema tiene la ventaja de una mayor precisión a pesar de ser menos sensibles al ruido, a perturbaciones y a cambios en el entorno [Norman, 2004].

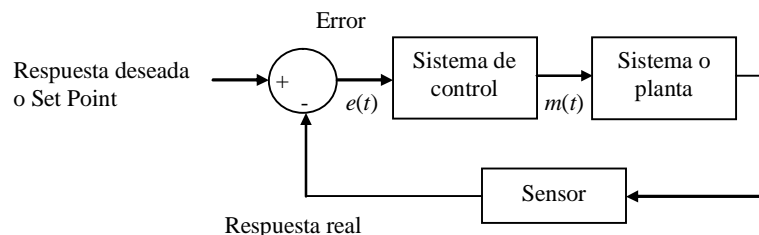


Figura 1.4. Sistema de control realimentado.

Con referencia a la figura 1.5, existen tres requerimientos para el diseño de estos sistemas:



1. Respuesta transitoria. Las entidades físicas no pueden cambiar su estado (por ejemplo, su posición o velocidad) de manera instantánea. El estado cambia en forma que está relacionada con el dispositivo físico y con la manera en que adquiere o disipa energía. Así la parte mecánica puede experimentar el cambio deseado a medida que hace lo que se le indica. A esta parte se le llama respuesta transitoria.
2. Estabilidad. Después de la respuesta transitoria, un sistema físico aproxima su respuesta en estado estable, que es su aproximación a la respuesta deseada.
3. Errores en estado estable. Cuando la salida es diferente con respecto a la entrada se produce un error en estado estable el cual es inherente en el sistema diseñado.

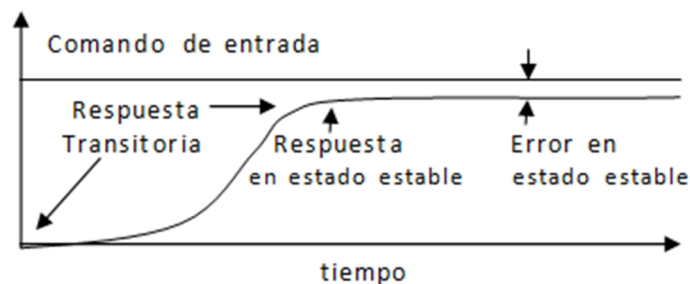


Figura 1.5. Comportamiento deseable de un sistema de control básico.

Un sistema de control analógico PID (Proporcional, Integral y Derivativo) está dado por [Ogata, 1996]:

$$m(t) = K \left[ e(t) + \frac{1}{T_i} \int_0^t e(t) dt + T_d \frac{de(t)}{dt} \right]$$

En donde  $K$  es la ganancia proporcional,  $T_i$  es el tiempo integral y  $T_d$  es el tiempo derivativo. Estas constantes deben ajustarse para cada caso particular de manera que muestren el mejor desempeño.

En el dominio de Laplace:

$$M(s) = K \left[ E(s) + \frac{1}{T_i} \frac{E(s)}{s} + T_d s E(s) \right]$$

En el dominio discreto:

$$M(z) = E(z) \left[ K_P + \frac{K_I}{1 - z^{-1}} + K_D (1 - z^{-1}) \right]$$

En donde las constantes, proporcional  $K_P$ , derivativa  $K_D$ , e integral  $K_I$  también deben ajustarse a las condiciones particulares de cada problema. Anti-transformando la ecuación anterior para obtener la salida del controlador digital en el tiempo discreto:

$$m(k)=m(k-1)+e(k)[K_P+K_I+K_D]-e(k-1)[K_P-2K_D]+e(k-2)K_D \quad (1)$$

En donde:

- $m(k)$  Voltaje discreto actual generado por el controlador.
- $m(k-1)$  Voltaje discreto anterior generado por el controlador.
- $e(k)$  Error discreto actual.
- $e(k-1)$  Error discreto anterior.
- $e(k-2)$  Error discreto anterior al anterior.

La ecuación (1) es importante en el desarrollo de este trabajo de tesis ya que permite la implementación del control de posición digital en lazo cerrado.

### 1.3. Microcontrolador PIC18F4550

Existe una gran cantidad de microcontroladores con características específicas de cada modelo para satisfacer las diferentes necesidades de los sistemas a crear y con la ventaja de ser económicos en comparación con otros dispositivos electrónicos. Uno de estos modelos es el PIC18F4550, el cual contiene los módulos requeridos para el desarrollo del presente proyecto de tesis.

#### 1.3.1. Especificaciones generales

En la hoja de datos se presenta la siguiente tabla 1.1 donde se dan las características de la familia de microcontroladores PIC18FXXXX.

Features	PIC18F2455	PIC18F2550	PIC18F4455	PIC18F4550
Operating Frequency	DC – 48 MHz	DC – 48 MHz	DC – 48 MHz	DC – 48 MHz
Program Memory (Bytes)	24576	32768	24576	32768
Program Memory (Instructions)	12288	16384	12288	16384
Data Memory (Bytes)	2048	2048	2048	2048
Data EEPROM Memory (Bytes)	256	256	256	256
Interrupt Sources	19	19	20	20
I/O Ports	Ports A, B, C, (E)	Ports A, B, C, (E)	Ports A, B, C, D, E	Ports A, B, C, D, E
Timers	4	4	4	4
Capture/Compare/PWM Modules	2	2	1	1
Enhanced Capture/ Compare/PWM Modules	0	0	1	1
Serial Communications	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART	MSSP, Enhanced USART
Universal Serial Bus (USB) Module	1	1	1	1
Streaming Parallel Port (SPP)	No	No	Yes	Yes
10-Bit Analog-to-Digital Module	10 Input Channels	10 Input Channels	13 Input Channels	13 Input Channels
Comparators	2	2	2	2
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST), MCLR (optional), WDT
Programmable Low-Voltage Detect	Yes	Yes	Yes	Yes
Programmable Brown-out Reset	Yes	Yes	Yes	Yes
Instruction Set	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled	75 Instructions; 83 with Extended Instruction Set enabled
Packages	28-pin PDIP 28-pin SOIC	28-pin PDIP 28-pin SOIC	40-pin PDIP 44-pin QFN 44-pin TQFP	40-pin PDIP 44-pin QFN 44-pin TQFP

Tabla 1.1. Características del dispositivo [Microchip, 2007-1].

Como se ve en la tabla 1.1, el PIC18F4550 cuenta con 2 módulos de PWM, 13 entradas de conversión analógica digital, puertos A-E, “resets”, memoria para programar, cantidad de memoria EEPROM entre otros.

Otras de las características importantes para el funcionamiento de éste dispositivo son:

- Voltaje sobre VDD con respecto a Vss es de -0.3V a +7.5V.
- Voltaje en MCRL con respecto a Vss es de 0V a +13.25V.
- Máxima corriente de salida del pin Vss es de 300mA.
- Máxima corriente de salida del pin VDD es de 250mA.
- Tiene una salida interna de regulador de voltaje de USB en 3.3V en el pin 18.
- Requiere de un oscilador externo o un reloj externo de valor máximo a 48MHz asociado al pin con función de OSC1 y el cual proporciona el tiempo de cada ciclo en el que trabajará el PIC.

A continuación se presenta la figura 1.6 que muestra el diagrama del PIC18F4550 de 40pines, con la descripción de cada uno de ellos.

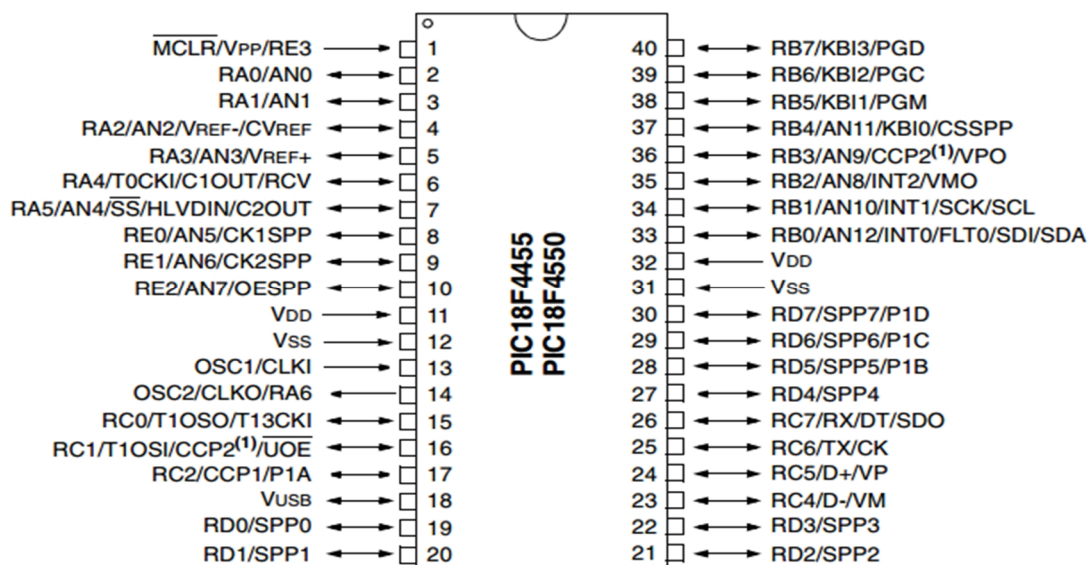


Figura 1.6. Diagrama del PIC18F4550 [Microchip, 2007-1].

Por otra parte, para la comunicación entre la interfaz de la PC con el PIC se tienen también las siguientes características:

- Baja velocidad de 1.5 Mb/s y Alta velocidad de 12 Mb/s
- Soportes de control, interrupción y transferencias.
- Soporta hasta 32 Endpoints (16 bidireccional).
- Tiene 1Kbyte Dual Access RAM para USB.
- On-Chip USB transmisor y receptor con On-Chip regulador de voltaje.
- Interfaz para Off-Chip USB transmisor y receptor.
- Puerto de transmisión paralela (SPP) para la transferencia de transmisión USB (Solo dispositivos de 40/44 pines).

### 1.3.2. Herramientas de desarrollo

Un microcontrolador no puede funcionar con solo la alimentación, si no que necesita de diferentes herramientas para desempeñar las funciones para las cuales fue diseñado. Estas son de dos tipos:

- Softwares: Ayudan a la programación del PIC para realizar una tarea específica.
- Electrónicas: Son los complementos que necesita el PIC para ser programado o embebido en una aplicación.

A continuación se describen con más detalle algunos de estos complementos.

### **Lenguaje en C++**

El lenguaje de programación C es un lenguaje estructurado que no permite declarar subrutinas dentro de otras subrutinas como lo hacen en otros tipos de lenguajes. Permite la conversión entre diferentes tipos de datos y la asignación entre tipos de datos diferentes.

Consta de un conjunto de funciones, teniendo como principal función la llamada “main”, la cual es la primera que se ejecuta al comenzar el programa y llamándose desde ella al resto de funciones que compongan un programa.

Por otro lado el lenguaje C++, es un derivado del lenguaje C que aprovecha su estructura para facilitar la creación de programas, pero teniendo la gran diferencia de ser un lenguaje orientado a objetos donde se permite la definición, creación y manipulación de objetos (Un objeto es una entidad con datos y formas de operar sobre ellos).

### **Compilador CCS C**

Es un compilador desarrollado para Microchip PIC® MCU. Dispone de una gran cantidad de librerías predefinidas, comandos de procesado y también suministra los controladores para diversos dispositivos.

El compilador es un “cross-compiler”, donde los programas son editados y compilados a instrucciones de máquina desde la computadora. El código de máquina puede ser cargado desde la PC al PIC mediante cualquier programador y ser depurado desde la misma computadora.

Con relación a la figura 1.7, la estructura básica de un programa C con CCS es la siguiente:

- Directivas de preprocesado. Controlan la conversión de programa a código máquina.
- Programas o funciones. Conjunto de funciones, donde la función principal es main().
- Instrucciones. Indican el comportamiento que adoptara el PIC.
- Comentarios. Permiten describir el significado de cada comando.

```

TESIS Are\Controlador_de_Motor_TESIS-USB\PicControladorM-USB\PFC7.c*
#include <18F4550.h>
#define ADC_10
#define HSPDLL, NOWDT, NOPROTECT, NOLVP, NODEBUG, USBDIV, DLLS, CPUDIV1, VREGEN
#define delay(clock=2000000)
#define fixed_io (e_outputs=pin_e2, pin_e1, pin_e0)
#define USB_HID_DEVICE FALSE //deshabilitamos el uso de las directivas HID
#define USB_EP1_TX_ENABLE USB_ENABLE_BULK //turn on EP1(EndPoint1) for IN bulk/interrupt transfers
#define USB_EP1_RX_ENABLE USB_ENABLE_BULK //turn on EP1(EndPoint1) for OUT bulk/interrupt transfers
#define USB_EP1_TX_SIZE 3 //size to allocate for the tx endpoint 1 buffer
#define USB_EP1_RX_SIZE 4 //size to allocate for the rx endpoint 1 buffer
#include <pic18_usb.h> //Microchip PIC18Fxxx Hardware layer for CCS's PIC USB driver
#include <PicUSB.h> //Configuración del USB y los descriptores para este dispositivo
#include <usb.c> //handles usb setup tokens and get descriptor reports

int8 LB, MB, HS, LBY, MSY, HBY;

void comandoR1X(byte comando)
{
    output_b(0x01);
    output_d(comando);
    delay_cycles(1);
    delay_ms(3);
}

void main(void)
{
    int8 recibe[4]; // declaración de variables
    int8 envia[3];
    int8 envial[3];
    signed int32 posrX=0;
    // inicializa PWM
    setup_ccp1 (ccp_pwm); //habilita el pwm
    setup_ccp2 (ccp_pwm); //habilita el pwm
    setup_timer_1 (T1_INTERNAL, T1_DIV_BY_8); // Configura el timer 1
    setup_timer_2 (T2_DIV_BY_16, 255, 1); // Configura el timer 2
}

```

Figura 1.7. Estructura básica en CCS C.

## MPLAB IDE

Es un editor de programas gratuito, el cual ayuda en la escritura correcta del código basado en microcontroladores PIC desde la computadora. Es llamado como entorno de desarrollo integrado, o IDE, por que proporciona un único ambiente integrado para desarrollar códigos para microcontroladores.

MPLAB tiene componentes llamados "debuggers" y simuladores de software libre para ayudar a examinar el código de todos los dispositivos PIC MCU. Incluso sí el hardware no está todavía definido, se puede comenzar a depurar el código con el simulador.

Los componentes de MPLAB IDE son los siguientes:

- Edición. Este editor de programas ayuda a escribir correctamente un código con el lenguaje elegido, apoyándose con los cambios de color en el código y asegurando así una sintaxis correcta.
- Assembler/Linker. "Assembler" puede ser usado de forma autónoma para ensamblar un único archivo. Por otro lado "linker" es el responsable de posicionar el código compilado dentro de las áreas de memoria del microcontrolador.
- Project manager. Proporciona integración y comunicación entre los IDE y las herramientas de lenguaje, permitiendo gestionar varios archivos para ser editados en una aplicación por medio de las carpetas: "source files", "header files" y "library files".
- Debugger. Depura el código por simulación, permite puntos de rupturas y ejecución paso a paso. Funciona en conjunto con el editor para hacer referencia a la información del destino que está siendo depurado.

- Programmer. Crea un archivo “.hex” para ser grabado en el microcontrolador.

Para crear un nuevo proyecto en MPLAB IDE se siguen los siguientes pasos:

- Se crea un archivo tipo ASM con el nombre del proyecto.
- Se elige el tipo de PIC que se va a utilizar.
- Escoger el tipo de compilador (CCS C.exe) y su ubicación.
- Dar el nombre y la ubicación que va a tener el proyecto a crear.
- Se agrega el archivo .ASM en el SOURCE FILE.

Al terminar de crear el proyecto, se puede empezar con la realización del programa para después compilarlo y si el programa no tiene errores de sintaxis, se creará un archivo “.hex” que es el archivo que se debe de grabar en el PIC.

### **Microchip MCHPFSUSB Bootloader**

“Bootloader” es un software que ayuda en la grabación de un programa al microcontrolador sin tener que utilizar un dispositivo externo.

Las herramientas requeridas para el Bootloader MCHPFSUSB [Microchip, 2007-2] son las siguientes:

- Necesita de un archivo hexadecimal “.hex” que va a ser proporcionado por el programa MPLAB.
- Se requiere de un cable USB para la conexión entre la PC y la tarjeta del microcontrolador.
- Un microcontrolador con el controlador “bootloader” grabado en su memoria para poder ser reconocido en la computadora y así programar el microcontrolador con el archivo de MPLAB.

Tomando como referencia la figura 1.8, el modo para emplear el MCHPFSUSB es el siguiente:

- Se abre el archivo PDFSUSB.exe.
- Se conecta la tarjeta del microcontrolador para tener comunicación con la computadora.
- Al ser reconocido el dispositivo, el programa lo reconocerá como PICDEM FS USB 0(Boot), por lo tanto se elige como dispositivo.
- Se escoge la opción “Erase Device” para limpiar el microcontrolador.

- Se carga el archivo .hex.
- Se programa el dispositivo.
- Se ejecuta el programa. Al escoger esta opción, el programa terminará de ejecutar el “bootloader” y lo mandará directo al programa grabado en el PIC.

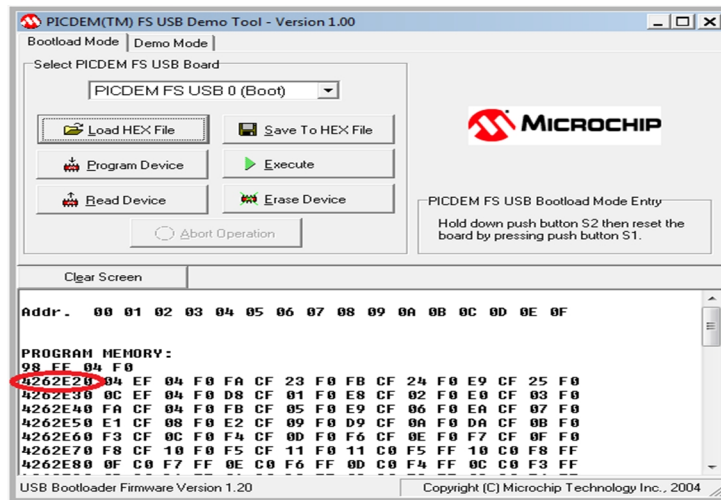


Figura 1.8. Programa “Bootloader”.

## Tarjeta para programar el PIC

Para hacer la comunicación con la computadora y utilizar el software bootloader se necesita de una tarjeta que actúe como interfaz entre ambas.

Se utiliza el cable “USB-A/B” para la comunicación entre el microcontrolador y la PC. Como la computadora es la que proporciona los voltajes al microcontrolador, se recomienda conectar un fusible para que no se de una afectación en ambas partes.

En la conexión USB el pin RC4 y el pin RC5 son utilizados como “transceivers” cuando se conectan para el modulo USB, es decir, estarán mandando y recibiendo la información entre la PC y el microcontrolador. Se requiere de resistencias en ambos pines para disminuir la corriente y no afecte al dispositivo.

El pin Vusb es un regulador de voltaje interno de 3.3V, la especificación del USB dice que requiere 3.3V de operación para la comunicación, sin embargo, el resto del chip puede correr con más voltaje, por lo tanto el transceiver interno se le suministra energía de forma separada por medio de éste pin. Es importante mencionar que en las especificaciones del regulador



de voltaje USB (pin Vusb), se requiere de un capacitor externo con un valor mínimo de 220nF.

Los pines 11 y 32(Vdd) del microcontrolador son utilizados para la alimentación del dispositivo. El valor aceptado para Vdd es de 2V a 5.5V. Debido a que la tensión puede tener fluctuaciones a causa de la componente de alterna que pasa por encima del voltaje de directa, causando ruido y problemas para el funcionamiento del circuito, se agrega a cada pin capacitores de “bypass” para eliminar las frecuencias presentes, almacenando una carga eléctrica que es cedida a la línea de alimentación durante una bajada transitoria de la tensión, proporcionando una pequeña alimentación añadida que minimiza el ruido.

Los valores más comunes para los capacitores de bypass son 0.1  $\mu$ F. y 1  $\mu$ F. Pero para obtener el valor deseado se requiere saber el valor de la frecuencia. Cuanto mayor sea la frecuencia, menor es el valor del capacitor, mientras más baja sea la frecuencia, el valor será mayor. Lo anterior se puede intuir de la siguiente ecuación.

$$f = \frac{1}{2t_R}$$

Donde  $t_R$  = tiempo de subida.

Por otra parte, la mayoría de los circuitos impresos son diseñados para mantener una corta distancia entre masa y alimentación. Esto suele realizarse laminando la línea de alimentación con el plano de masa, con lo que puede haber capacitancias inherentes debido a la proximidad eléctrica. Esta no es una solución suficiente para eliminar las fluctuaciones de alimentación, por lo que deberemos adoptar otro método. La solución está en situar el capacitor de bypass lo más próximo a los pines de alimentación del dispositivo. La razón de esta proximidad está en la propia inductancia de las pistas del circuito impreso, que ejerce un efecto contrario al del capacitor impidiendo su función y manteniendo el ruido en la alimentación [Bolaños, 2015].

Los pines 12 y 31 (Vss) son las tierras, los cuales cerrarán el circuito dando funcionalidad al microcontrolador para que pueda realizar las tareas programadas en él.

El pin RE3 es un puerto exclusivo de entrada digital, con dos tipos de funcionalidad: la línea de reset externa (MCLR) y la línea de programación (VPP). Si se quiere desactivar la función de reset y utilizar RE3 como entrada

digital se debe de poner a “cero lógico” el bit MCLR o utilizar el comando “#fuse NOMCLR” en la programación del microcontrolador.

Si se requiere ir a la función de programación desde RE3, entonces no se debe de poner el comando antes mencionado y para realizar esta función a voluntad, se requiere de un “push-button” (cerrado o abierto) con el cual se debe aplicar voltaje alto a RE3.

El pin RB5 puede ser usado como programación LVP (Programación de Bajo Voltaje). Cuando el pin de configuración LVP es programado, este pin pierde la función I/O y se convierte en una función de prueba de programación [Huan, 2005].

El pin OSC1/CLKI tiene la función de entrada para el oscilador de cristal, mientras que OSC2/CLKO/RA6 tiene la función de salida de oscilador de cristal. Para éste modo, se conecta el oscilador de cristal en OSC1 y OSC2 en un corte en paralelo para el establecimiento del oscilamiento.

Se requieren de ciertas resistencias y capacitores para la operación del cristal, como se muestra en la figura 1.9.

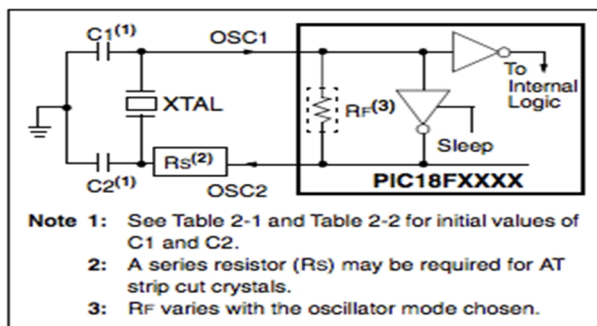


Figura 1.9. Configuración del oscilador [Microchip, 2007-1] Rs se requiere para evitar la saturación del cristal con la impulsión de bajo nivel de especificación.

En la tabla 1.2 se muestra la relación del valor de los capacitores con la frecuencia del cristal a utilizar.

Osc Type	Crystal Freq	Typical Capacitor Values Tested:	
		C1	C2
XT	4 MHz	27 pF	27 pF
HS	4 MHz	27 pF	27 pF
	8 MHz	22 pF	22 pF
	20 MHz	15 pF	15 pF

Tabla 1.2. Valor Cristal - Capacitor [Microchip, 2007-1].

La alta capacitancia incrementa la estabilidad del capacitor, pero también incrementa el tiempo de inicio.

Con todas las especificaciones mencionadas en esta sección, se muestra en la figura 1.10 el esquema final para el funcionamiento de la tarjeta que permite programar al microcontrolador desde la computadora.

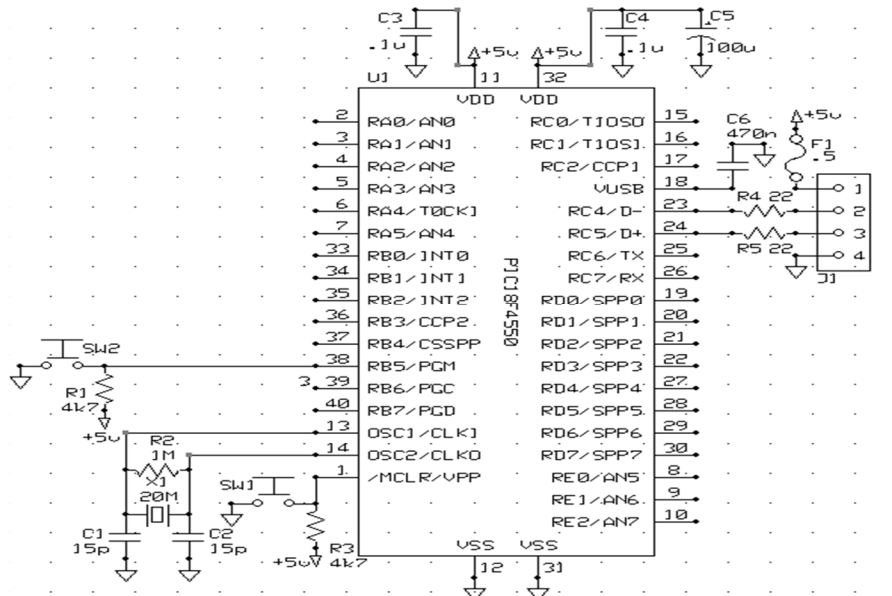


Figura 1.10. Esquema final de la tableta del PIC18F4550.

## 1.4.Desarrollo de aplicaciones en Visual C++

Visual C++ es un entorno de desarrollo integrado que permite la programación orientada a objetos de manera sencilla, incluyendo así las siguientes herramientas principales para su funcionamiento:

- Editor de texto.
- Compilador/Enlazador.
- Depurador.
- Visor de datos y dependencias.

Visual C++ trabaja en conjunto con el sistema de desarrollo SDK de Windows (encabezados, bibliotecas y herramientas que permiten ejecutar los componentes como ventanas, menús, diálogos, etc.). Para tener éstas funciones, se necesita de la librería de clases MFC (Microsoft FoundationClasses) la cual crea y gestiona los componentes típicos de

Windows sin tener que utilizar directamente la interfaz de programación de aplicaciones de Windows (API), ya que la MFC agrupa las librerías de Windows en clases C++.

El ambiente MFC también proporciona el modelo llamado “visor de datos y dependencias”, que permite diseñar aplicaciones de manera que los datos de dicha aplicación se encuentren separados de los elementos que componen la interfaz del usuario, logrando así, disminuir la complejidad del programa y poder modificarlo de manera independiente.

MFC también proporciona las ventanas de diálogos, que son elementos utilizados para la interacción con el usuario, figura 1.11.

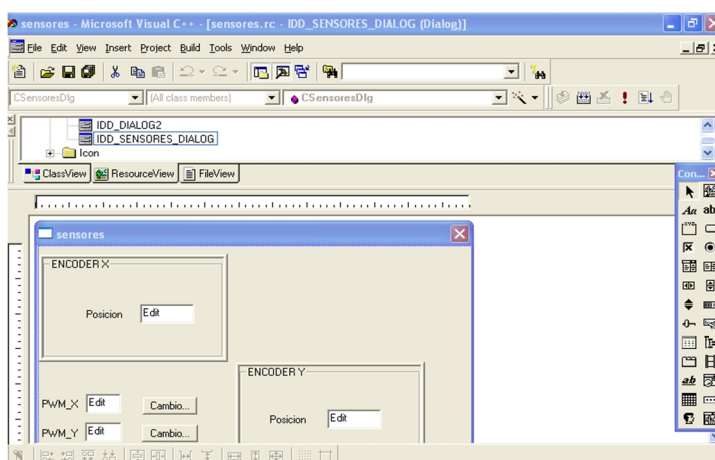


Figura 1.11. Ambiente MFC.

Un diálogo es una ventana que contiene en su interior ventanas dependientes o hijas que son controles como botones, listas, cajas de edición, etc., identificados por un número único. Cuando un diálogo se despliega, se convierte en una ventana exclusiva dejando inhabilitada cualquier operación con el resto de la aplicación. Las plantillas de estos diálogos pueden diseñarse mediante herramientas integradas en el entorno de desarrollo como la carpeta ResourceView (vease en la figura 1.11).

En la creación de un proyecto, Visual C++ genera para cada proyecto dos ficheros que lo definen, el fichero de “workspace” (con extensión wsp) y un “makefile” (con extensión mak) estándar que permitiría la utilización del mismo proyecto en otro entorno distinto. Desde el punto de vista funcional, el proyecto contiene referencias a cada uno de los ficheros fuentes (C/C++, con extensiones c y cpp respectivamente), objetos, librerías o ficheros de recursos (extensión rc) que se deben utilizar para construir el objetivo final del

proyecto. En definitiva, para crear cualquier programa con Visual C++ debemos comenzar creando un proyecto para él, codificando y añadiendo los módulos necesarios a dicho proyecto, y definiendo los recursos asociados [Instituto Norbert Wiener, 2015].

Para la creación de un proyecto MFC (Windows XP) se debe hacer lo siguiente:

- Se crea un nuevo proyecto en la opción FILE.
- Aparece el dialogo que permite especificar el tipo de “workspace”, se da la opción Aplicación (.EXE) basada en la MFC (MFC AppWizard) que da el tipo de proyecto.
- Se nombra al proyecto creado.
- En el paso 1 se permite identificar el modelo de ventana principal de la aplicación: SDI, MDI o basada en diálogo. Se escoge “Basada en un dialogo”.
- Aparece una secuencia de diálogos, los cuales se deben de mantener con la configuración que el mismo programa le ha especificado, por tanto se debe proceder a escoger la opción “NEXT” hasta finalizar.

Una vez creado el programa, se crea la caja de dialogo, la cual se puede programar en un entorno distinto ayudado por el “Visor de Proyectos”. El Visor de Proyectos es una ventana en la que se puede visualizar el contenido de los proyectos. Está formada por tres carpetas:

- ClassView, o Visor de clases. Representa el proyecto con las clases que lo componen. Aquí se puede añadir datos o funciones fácilmente.
- ResourceView, o Visor de recursos. Permite añadir, modificar y eliminar recursos de la aplicación.
- FileView, o Visor de ficheros. Representa el proyecto mediante la lista de ficheros fuente que lo componen. La gestión y tratamiento de los recursos sólo se pueden hacer mediante ResourceView, mientras que ClassView y FileView son casi similares en cuanto a su objetivo último: el mantenimiento del código del programa [Instituto Norbert Wiener, 2015].

Finalizado el programa, se compila para verificar la correcta sintaxis del mismo con relación a la caja de dialogo, para después proceder a la ejecución de la aplicación creada, como se muestra en la figura 1.12.

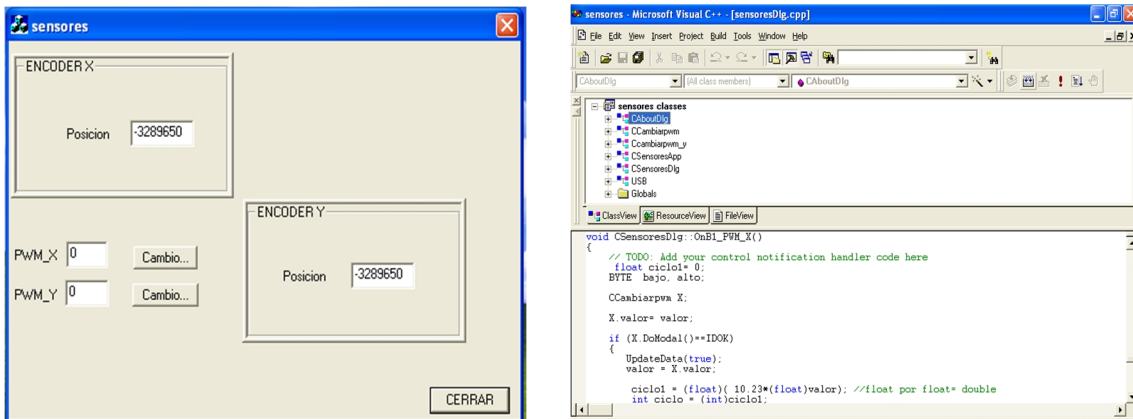


Figura 1.12. A) Programación MFC en Visual C++. B) Aplicación del programa (.exe).

## 1.5. Dispositivos electrónicos empleados

Para el desarrollo del proyecto se utilizaron diversos dispositivos electrónicos de los cuales se proporciona una breve descripción.

### 1.5.1. Compuerta digital 74HC04

Proporciona seis inversores independientes con salidas estándar push-pull, el comportamiento de la función Boleana del dispositivo es:  $Y = -Y$ , ver figura 1.13. Sus características son:

- Voltaje de operación 2V a 6V.
- Corriente por  $V_{cc}$  de 50mA.
- Voltaje de salida  $V_o = V_{cc}$ .

Se recomienda poner capacitores de 22pF para disipación de voltaje en la compuerta, si se tiene una frecuencia de 1MHz y un capacitor de 4pF para el capacitor de entrada si  $V_i = V_{cc}$  o  $-V_{cc}$ .

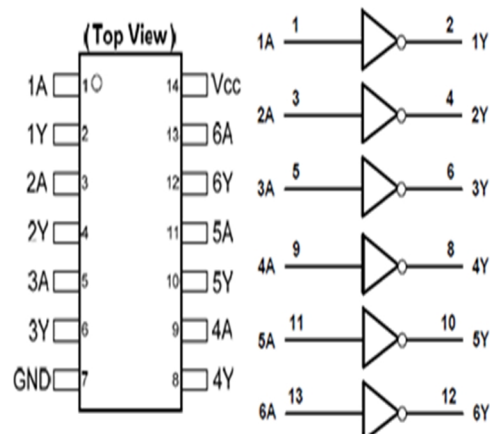


Figura 1.13. Diagrama de conexión [Texas Instruments, 2015].

### 1.5.2. Transistor NPN 2N3904

Es un transistor NPN, como el mostrado en la figura 1.14, de propósitos generales que es comúnmente utilizado para amplificación de señales. Sus principales características son:

- Voltaje colector-base máximo 60V.
- Voltaje colector-emisor máximo 40V.
- Voltaje emisor-base máximo 6V.
- Corriente de colector máxima 200mA.

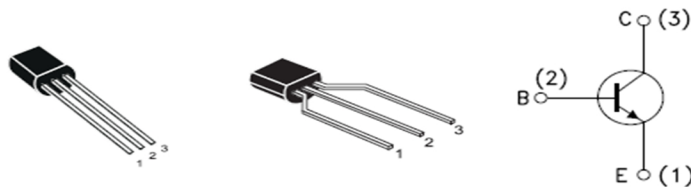


Figura 1.14. A) Diagrama de conexión B) Diagrama interno.

### 1.5.3. Relevador G5LE-1A4-DC12

Este tipo de relevador, mostrado en la figura 1.15, es un interruptor controlado que cambia de estado abierto/apagado cuando por medio de un circuito eléctrico se induce corriente a la bobina, provocando el cambio de estado del interruptor por medio del magnetismo producido por ella. Sus características principales son las siguientes:

- Tensión nominal 12V.
- Corriente nominal 30mA.
- Resistencia de la bobina 400ohm.
- Consumo de energía aproximado 360mW.

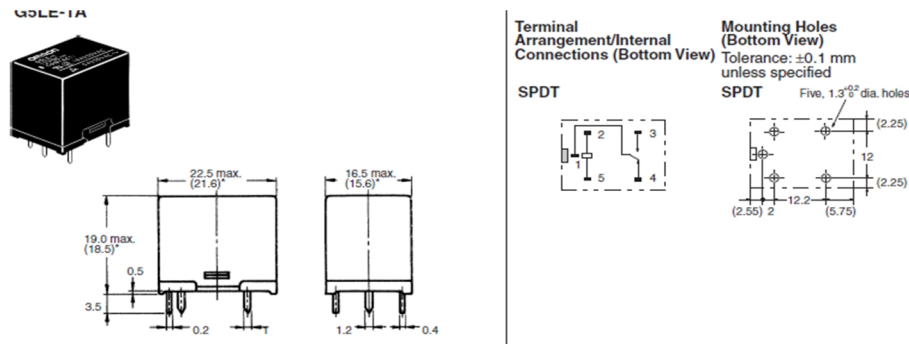


Figura 1.15. Diagrama de conexión [Omron, 2013].

#### 1.5.4. Reguladores de voltaje LM7818 Y LM7805

Los reguladores de voltaje de la familia 78XX, que se muestran en la figura 1.16, tienen como función el mantener el voltaje a un valor constante independientemente de las variaciones que se tengan en la carga o alimentación de voltaje. Son adecuados para aplicaciones que requieren corriente arriba de 1A. Sus características principales son las siguientes:

- Corriente de salida de 1.5A.
- Voltajes de salida de 5V, 6V, 8V, 9V, 12V dependiendo del regulador a utilizar.
- Protección SOA de corto circuito.

Las características particulares para el 7805 son:

- Voltaje de salida de 4.75V a 5.25V.
- Regulación de carga 50mV máxima.
- Corriente pico de salida 1.8A.

Las características particulares para el 7818 son:

- Voltaje de salida de 17.3V a 18.7V.
- Regulación de carga 180mV máxima.
- Corriente pico de salida 1.8A.

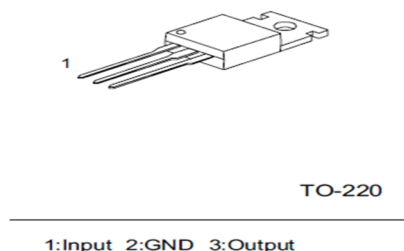


Figura 1.16. Diagrama de conexión [Fairchild, 2014].



### 1.5.5. Transistor TIP42C

Son transistores de silicón epitaxial PNP, como se muestra en la figura 1.17, que amplifican, conmutan o rectifican señales eléctricas soportando altos valores de corriente y voltaje. Es recomendable colocarles disipadores debido a las corrientes que maneja. Sus características principales son las siguientes:

- Voltaje colector-base máximo -100V.
- Voltaje colector-emisor máximo -100V.
- Voltaje emisor-base máximo -5V.
- Corriente de colector -6A.
- Voltaje de saturación colector-emisor -1.5V.
- Ganancia de corriente de 15 a 75.

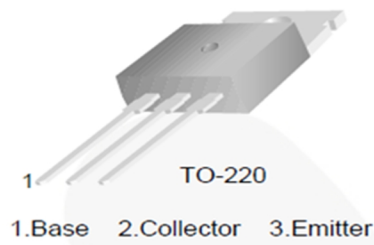


Figura 1.17. Diagrama de conexión [On Semiconductor, 2005].

### 1.5.6. Interruptor óptico H21A1

Es un interruptor óptico, como se muestra en la figura 1.18, que consiste de un diodo emisor de infrarrojos junto con un fototransistor de silicio. La brecha de la carcasa proporciona un medio para interrumpir la señal con un material opaco. Si la brecha se encuentra libre de material, la salida del óptico será un ON ( $V_{cc}$ ). Si la brecha está con el material externo, la salida del óptico será OFF (GND). Sus características principales son:

- Voltaje inverso 6V.
- Entrada de corriente directa continua 50mA.
- Salida de ruptura del sensor colector a emisor 30V.
- Voltaje emisor a colector 4.5V.
- Corriente de colector 20mA.
- Tiempo prendido y apagado  $I=30\text{mA}$ ,  $V_{cc}= 5\text{V}$ ,  $R= 2.5\text{Kohm}$ .

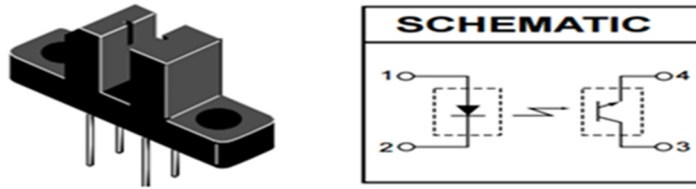


Figura 1.18. Diagrama de conexión [Fairchild, 2001].

### 1.5.7. Puente rectificador RS607

Es un puente rectificador de onda completa, como se muestra en la figura 1.19, esto se logra por medio de 4 diodos. Su función principal es separar los semi-ciclos positivos y negativos para después, por medio de un capacitor, rectificar la señal a un valor constante. Sus principales características son:

- Máximo voltaje RMS 700V.
- Máximo voltaje de bloqueo 1000V.
- Máxima corriente rectificada 6 A.

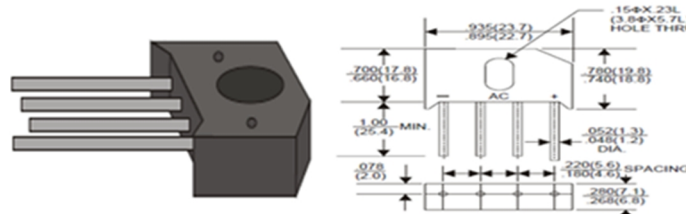


Figura 1.19. Diagrama de conexión [Rectron Semiconductor, 2005].

### 1.5.8. Motor Copley Control SSC 23SMDC-LCSS

Es un servomotor de corriente directa como el mostrado en la figura 1.20. Una de las aplicaciones de este tipo de motor es controlar la velocidad o control de posición mediante codificadores ópticos comúnmente conocidos como “encoders” que constan de un disco transparente, el cual tiene un patrón único formado por áreas transparentes y negras que determinan la resolución del encoder para poder hacer un conteo con respecto al movimiento del servomotor. Sus principales características son:

- Voltaje máximo 60V.
- Máxima velocidad de operación 6000RPM.
- Pico de torque 400 oz-in.
- Voltaje constante 10.2 V/KRPM.
- Corriente constante máxima 4 Amps.
- Máxima corriente pico 34 Amps.



Figura 1.20. Servomotor DC [ServoSystems, 2004].

### 1.5.9. Circuito de interfaz entre codificador y microprocesador

Para poder controlar la posición de un motor a través del microcontrolador, se requiere de una interfaz que obtenga la información de los codificadores ópticos y mandarla en forma de paquetes de datos al microcontrolador, esta función la desempeña el circuito electrónico LS7266R1-S que se muestra en la figura 1.21. El LS7266R1 es un dispositivo que puede ser configurado para la realización de un conteo y mandar la información recolectada en paquetes de datos al microcontrolador.

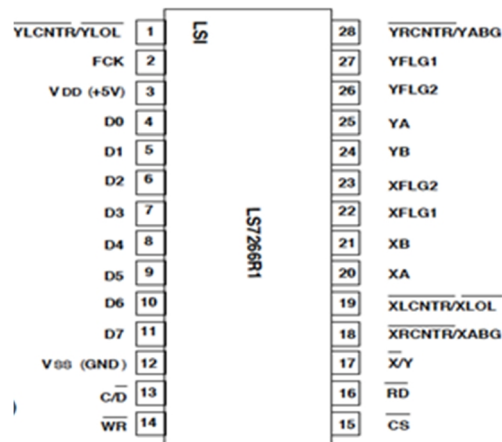


Figura 1.21. Configuración de terminales [USDigital, 2000].

Sus características principales son las siguientes:

- Tiene dos contadores de 24-bits para soportar dos ejes X y Y en aplicaciones de control de movimiento.
- Programación independiente en cada eje.
- Tipos de modo de conteo:
  - Quadrature (X1, X2, X4) / Non-quadrature.
  - Normal / Modulo-N / RangeLimit / Non-Recycle.
  - Binary / BCD.
- En modo non-quadrature, se requiere la frecuencia de conteo arriba de 30MHz.

- En modo X4 quadrature, se requiere una frecuencia de conteo arriba de 4.3MHz.
- TTL/CMOS compatible con I/Os.
- 8-bit de bus de datos I/O en 3 diferentes estados.
- Voltaje de operación 3V a 5.5V (Vdd - Vss).

Este dispositivo tiene un conjunto de registros asociados a cada eje X y Y correspondientes a los dos canales de entrada. Esta asociación se da con el prefijo X y Y con el objetivo de seleccionar los registros específicos y modificar el dispositivo en modo de lectura o escritura mediante los 3 bits más significantes del bus de datos (D7-D5). Los registros del dispositivo se muestran en la tabla 1.3.

Registro	Descripción
XPR y XYR	24 bits de datos pueden ser escritos dentro de cada PR, un byte a la vez, en una secuencia de tres datos de ciclos escritos.
XCNTR and YCNTR	Cada CNTR tiene 24 bits en los contadores síncronos UP/DOWN, el contador de los relojes de cada CNTR se deriva de sus entradas asociadas A/B. Cada CNTR se puede cargar con el contenido de su PR asociado.
XPSC y YPSC	El PSC proporciona la habilidad para generar frecuencias de reloj de filtro independientes para cada canal. Generan el filtro de reloj interno, FCKn es usado para validar las entradas XA, XB, YA, YB en el modo cuadratura.

Tabla 1.3. Registros del LS7266R1.

El filtro de frecuencia de reloj final es

$$f_{FCKn} = \frac{f_{FCK}}{n + 1}$$

donde  $n = PSC = 0$  hasta FFH. Para un conteo apropiado en modo cuadratura

$$f_{FCKn} \geq 8f_{QA} \text{ o } 8f_{QB}$$

Donde  $f_{QA}$  y  $f_{QB}$  son las frecuencias de reloj de entrada de A y B.















En modo non-quadrature el filtro de reloj no es necesario y la entrada FCK, debe ser ligada a Vdd.

Los pines de entrada y salida del circuito integrado se describen en la tabla 1.4.

Entrada/salida	Descripción
WR	Entrada de escritura. Bytes de control/datos.
RD	Entrada de lectura. Un nivel bajo aplicado habilita las banderas y OLs para ser lectura en el bus de datos.
CS	Chip de selección de entrada. Un nivel bajo habilita el chip en lectura o escritura.
C/D	Entrada de Control/Datos. Esta entrada selecciona entre un registro de control o registro de datos para Lectura/Escritura. Cuando es bajo, un registro de datos es seleccionado. Cuando es alto, un registro de control es seleccionado.
D0 - D7	Entrada/Salida del bus de datos.
FCK	Entrada de Filtro de reloj en modo cuadratura. El FCK está dividido internamente por 2 secciones de 8 bits programables, uno por cada canal.
X/Y	Selección entre los ejes X y Y para lectura y escritura. Si X/Y = 0 selecciona al eje X, si es X/Y = 1 selecciona el eje Y. X/Y es deshabilitado por D7 = 1 en modo control de escritura (C/D = 1).

Tabla 1.4. Descripción de los pines

Para el mejor entendimiento en la configuración del dispositivo se muestran, en la tabla 1.5, los diferentes modos de configuración y su significado.

D7	D6	D5	C/D	RD	WR	X/Y	CS	FUNCTION
X	X	X	X	X	X	X	1	Disable both axes for Read/Write
X	X	X	0	1		0	0	Write to XPR byte segment addressed by XBP (Note 3)
X	X	X	0	1		1	0	Write to YPR byte segment addressed by YBP (Note 3)
0	0	0	1	1		0	0	Write to XRLD
0	0	0	1	1		1	0	Write to YRLD
1	0	0	1	1		X	0	Write to both XRLD and YRLD
0	0	1	1	1		0	0	Write to XCMR
0	0	1	1	1		1	0	Write to YCMR
1	0	1	1	1		X	0	Write to both XCMR and YCMR
0	1	0	1	1		0	0	Write to XIOR
0	1	0	1	1		1	0	Write to YIOR
1	1	0	1	1		X	0	Write to both XIOR and YIOR
0	1	1	1	1		0	0	Write to XIDR
0	1	1	1	1		1	0	Write to YIDR
1	1	1	1	1		X	0	Write to both XIDR and YIDR
X	X	X	0	0	1	0	0	Read XOL byte segment addressed by XBP (Note 3)
X	X	X	0	0	1	1	0	Read YOL byte segment addressed by YBP (Note 3)
X	X	X	1	0	1	0	0	Read XFLAG
X	X	X	1	0	1	1	0	Read YFLAG

X = Don't Care

Note 3: Relevant BP is automatically incremented at the trailing edge of RD or WR pulse

Tabla 1.5. Registros de modo de dirección [USDigital, 2000].

En el siguiente esquema de bloques de la figura 1.22 se muestra un ejemplo de conexión para registrar los valores de dos codificadores ópticos (encoder) y mandarlos al microcontrolador.

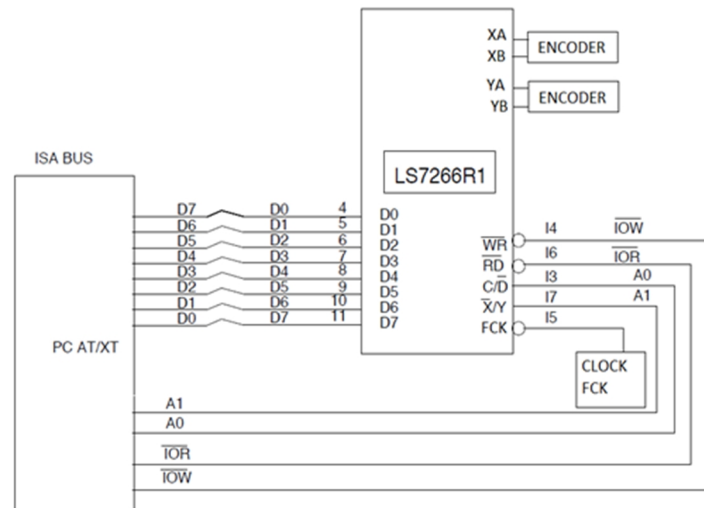


Figura 1.22. Esquema de aplicación.

Se conectan los pines del 13 al 14 para poder cambiar de registro entre X o Y, como también habilitar la escritura/lectura del microcontrolador hacia el LS7266R1.

Por otro lado, se tienen que registrar los valores de los codificadores ópticos en el microcontrolador y esto se logra conectando los pines 4 a 11 los cuales tienen la función de bus de datos. Por último, cada codificador óptico debe de ser conectado en un respectivo canal X/Y para que el contador del LS7266R1 pueda hacer un conteo con cada línea que pasa al girar el motor.

Se requiere de un reloj externo para que FCK pueda registrar los valores de manera cíclica.

### 1.5.10. Impulsor de motor de corriente directa

El puente H de la figura 1.23 es un conjunto de transistores y diodos que permiten a un motor de CD moverse en ambos sentidos.

El funcionamiento del puente H consiste en suministrar la corriente que se necesita para abrir y cerrar los transistores, comportándose como interruptores. Cuando se aplica tensión positiva los transistores Q1 y Q4 se activan produciendo el movimiento del motor hacia un sentido. Cuando se aplica la tensión inversa se activan ahora Q2 y Q3 provocando el movimiento hacia el otro sentido. Este puente cuenta con diodos que sirven de protección al momento de la conmutación de los transistores.

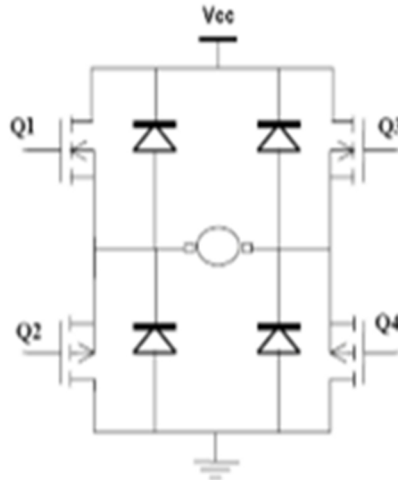


Figura 1.23. Puente H.

Teniendo en cuenta el funcionamiento de un puente H, entonces se procede a utilizar el LMD18201.

LMD18201 es un puente H utilizado como interfaz entre un motor y el microcontrolador, debido a que es designado para aplicaciones de control de movimiento. La configuración del puente H es ideal para el manejo de motores de DC, motores paso a paso, servomecanismos de posición y velocidad, entre otros. Sus principales características son las siguientes:

- Salida aproximada de 3A.
- Entradas compatibles con TTL y CMOS.
- Diodos de sujeción internos.
- Protección de corto de carga.
- Carga de bombeo interno con capacidad de Bootstrap externo.
- Voltaje de entrada  $V_s$  máximo 60V.
- Voltaje de bootstrap  $V_{out} = 16V$ .
- Corriente continua de salida 3A.

En la figura 1.24 se tiene el diagrama funcional del dispositivo como también la relación de los pines.

Para éste dispositivo la dirección y la magnitud de la señal se encuentran separadas, donde la dirección (pin 3) es manejada por la señal PWM y la entrada PWM (pin 5) es manejada por la magnitud de la señal, por lo que se encuentra ligado a un alto lógico (5V). La corriente entregada a la carga es proporcional al pulso.

En la figura 1.25 se muestra el funcionamiento que tiene el componente con respecto a la señal PWM.

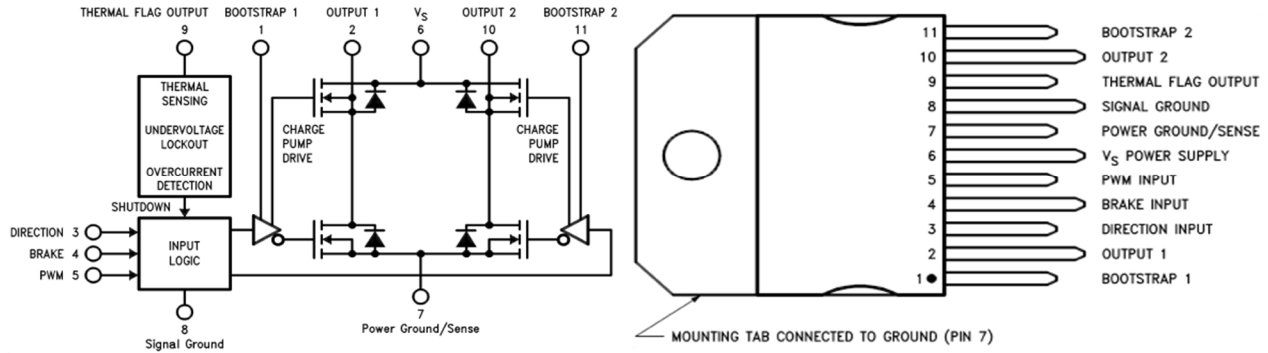


Figura 1.24. A) Diagrama de conexiones. B) Diagrama funcional [National Semiconductor, 1996].

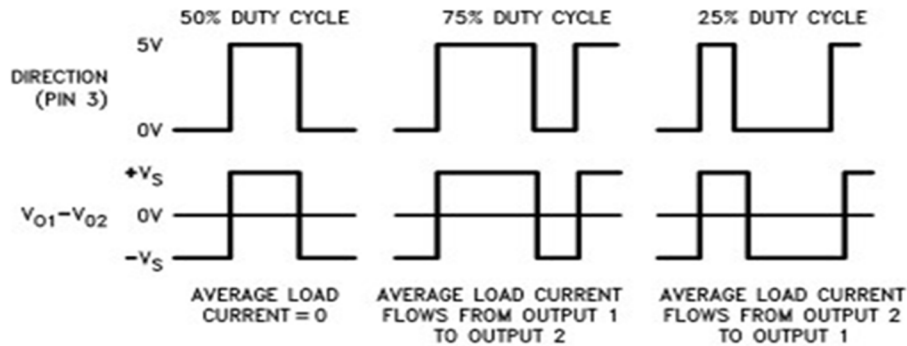


Figura 1.25. Control por ancho del pulso [Dee, 2015].

En la figura se explica que con el 50% del ciclo de trabajo de la señal PWM representa la unidad cero, es decir, el motor se quedará estático. Arriba del 50% del ciclo de trabajo la corriente fluirá de la salida 1 a la salida 2, pero si el ciclo de trabajo es menor al 50%, entonces la corriente fluirá de la salida 2 a la salida 1.

Una de estas aplicaciones generales se muestra en la figura 1.26 donde el motor debe de ser conectando entre la “sink” y “source”, teniendo las siguientes consideraciones:

- Durante los niveles de transición de conmutación se pueden crear inductancias parasitas causando problemas con el funcionamiento, por lo que el fabricante recomienda capacitores de bypass conectados en las entradas de voltaje, de preferencia capacitores de alta frecuencia cerámicos de  $1\mu\text{F}$ , pero si se manejan altas corrientes en el dispositivo entonces el valor del capacitor debe de ser de  $100\mu\text{F}$  por cada Amper.
- Por otra parte, el pin de “bootstrap” es en esencia una segunda bomba de carga mediante el cual se utiliza un condensador externo de gran valor que tiene energía suficiente para cargar rápidamente la las cargas parasitarias del dispositivo de potencia que resulta en mucho más



rápidos tiempos de subida, permitiendo también frecuencias mayores de 500kHz. Se recomienda un capacitor de 10nF.

- Si se quiere sensar la corriente que pasa a través del puente H se puede agregar una resistencia de 0.1 ohm.

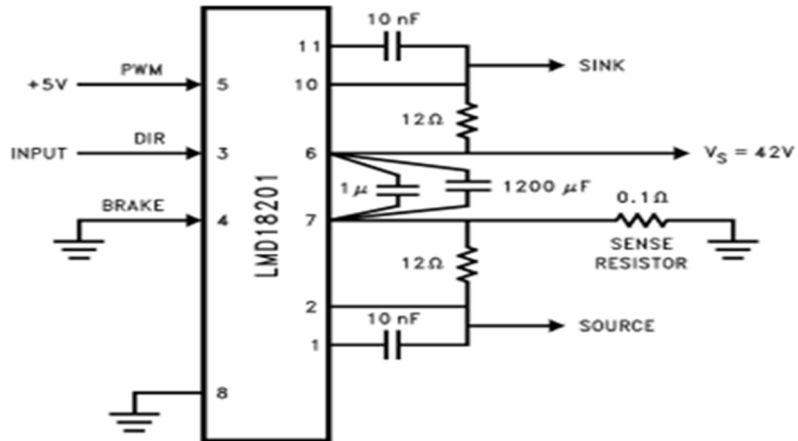


Figura 1.26. Ejemplo de aplicación [National Semiconductor, 1996].



# Capítulo 2

## Descripción del proyecto

En este capítulo se expone el diseño del sistema electrónico para la calibración de nivel de fluidos en tanques.

### 2.1. Esquema general

En el siguiente esquema se muestran las etapas del funcionamiento del sistema para la calibración de nivel de fluidos en tanques.

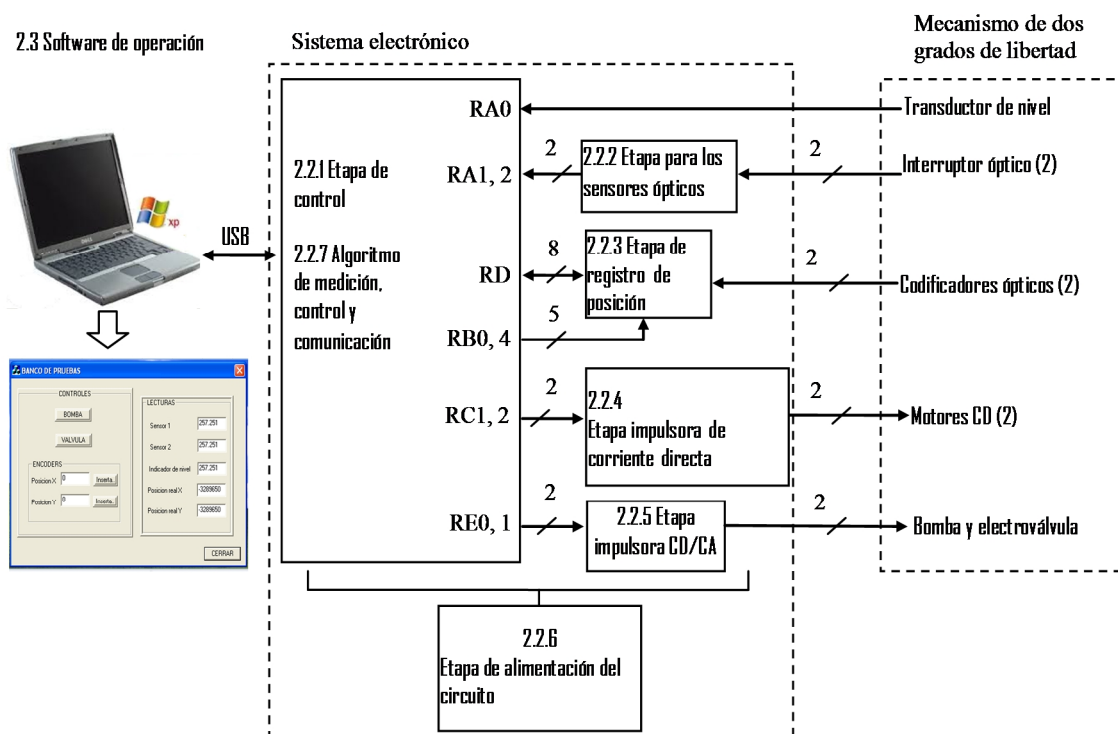


Figura 2.1. Esquema general del proyecto.

Este sistema consta de tres etapas esenciales:

**1. Mecanismo de dos grados de libertad (No reportado en el presente trabajo de tesis).** Es un dispositivo electromecánico al cual se sujeta un tanque contenedor de fluidos. Los dos grados de libertad permiten el desplazamiento del conjunto fluido y tanque para analizar el comportamiento del medidor de nivel integrado al tanque en diferentes posiciones. Los dos grados de libertad son impulsados por dos motores de corriente directa de los cuales es posible conocer su posición ya que tienen integrados codificadores ópticos.

**2. Software de operación (2.3).** Contiene la aplicación basada en Visual C++ para que el host pueda manipular el circuito electrónico y a la vez, tener la adquisición de medición de nivel para diferentes posiciones del tanque. Esta aplicación contiene:

- Visualización del valor del nivel del tanque.
- Detección de la posición origen de la plataforma.
- Visualización y control de posición de dos motores.
- Control de llenado y vaciado del tanque.

Para poder realizar esta comunicación se requiere de un programa controlador en la computadora y la activación del USB en el sistema electrónico, por último se requiere de un conector USB- A/B para la transmisión y recepción de datos.

**3. Sistema electrónico (2.2).** Para que pueda funcionar la aplicación creada, se requiere de un sistema electrónico con base en microcontrolador, con el objetivo de poder manipular los dispositivos electrónicos en conjunto con el mecanismo de dos grados de libertad.

El microcontrolador del sistema electrónico tiene pines con diferentes tipos de aplicaciones, por lo que los pines utilizados para el proyecto se explican a continuación:

El pin RA0 es un puerto de entrada que convierte la señal analógica a digital. Conociendo esta función, el pin se utiliza para obtener el valor del nivel del tanque, como se muestra en la figura 2.2. En este caso, se requiere de un transductor de nivel colocado dentro del tanque, entregando una señal eléctrica al pin RA0 y este la convierte en una señal digital para que pueda ser interpretada y leída en la aplicación final.

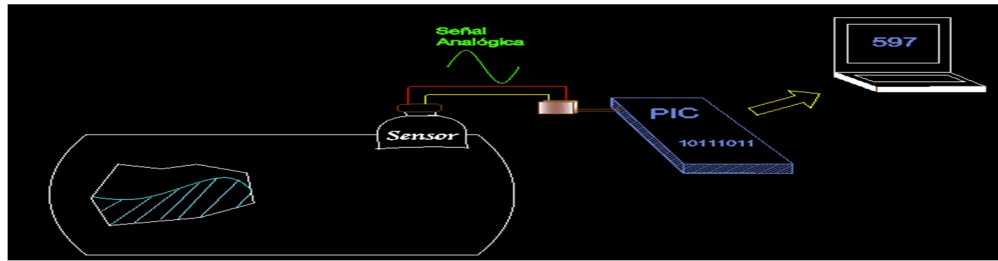


Figura 2.2. Funcionamiento del transductor de nivel.

Los pines RA1 y 2, son también convertidores A/D, pero estos son programados para dar una señal cuando la plataforma se encuentre en la posición de origen. Esto se logra mediante la etapa para los sensores ópticos (2.2.2.), debido a que estos dos sensores ópticos permitirán el paso de voltaje en el momento de estar posicionados en forma aproximada a una vertical, se utiliza esta función para que los motores tengan un punto de partida en el momento de cambiar de posición y poder tener una adquisición de valores reales, ver figura 2.3.

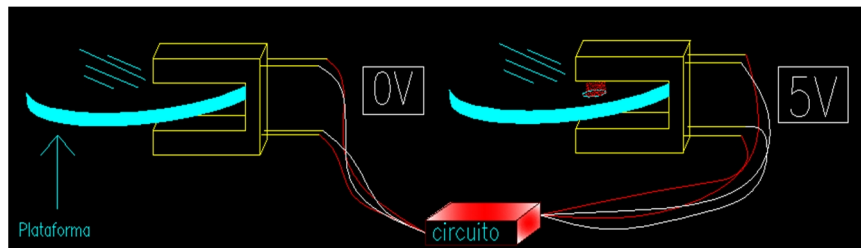


Figura 2.3. Funcionamiento de los sensores.

En la etapa de registro de posición (2.2.3.) se requieren de 8 pines que actúan como bus de transmisión y recepción de datos entre el microcontrolador y el circuito de interfaz para codificadores LS7266R1, como también de 5 pines más para el control de lectura y escritura en los registros X/Y del circuito de interfaz. En este paso, se utilizan los pines RD0 a RD7 como entradas/salidas digitales para el bus de datos y los pines RB0 a RB4 para la escritura/lectura del dispositivo. El objetivo de esta etapa es controlar la posición de los motores, obteniendo la lectura de los codificadores ópticos que van conectados al circuito de interfaz y a su vez son mandados al microcontrolador mediante el bus de datos, ver figura 2.4. Para lograr el control de los motores se necesita también de la etapa de control (2.2.1.), que es un programa que funciona como un sistema de control de lazo cerrado.

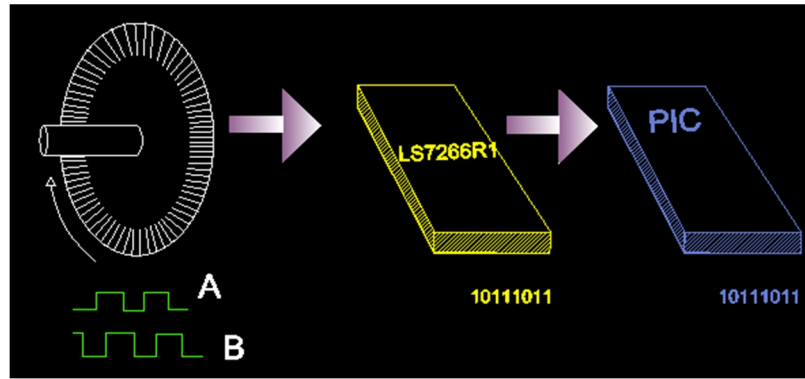


Figura 2.4. Funcionamiento del circuito de interfaz LS7266R1.

Por último, la función que se tiene en la aplicación, es poder manejar el valor de la posición y que se pueda visualizar valor real de la posición del motor corroborando que se tenga un valor mínimo en el margen de error.

Los pines RC1 y RC2 tienen la función de PWM que van conectados a cada LMD18201 (etapa impulsora de corriente directa 2.2.4.) con la finalidad de dar la dirección y velocidad que pueden tomar los motores. Esta etapa se encuentra ligada con la de control (2.2.1.), ya que, para poder cerrar el lazo del sistema de control se necesita manipular el valor del PWM.

Los pines RE0 y RE1 son utilizados como salidas, el objetivo es que actúen como interruptores para el apagado/prendido de la bomba y la electroválvula para el llenado y vaciado de fluido del tanque. Como el microcontrolador solo puede dar una salida de 5V se necesita de la etapa impulsora CD/CA (2.2.5.), para poder habilitar o deshabilitar la carga que se va a colocar. Esta etapa se configuró para que se pueda utilizar tanto corriente directa o alterna, debido a que no se saben las necesidades mecánicas que requieran.

Para que funcione todo el sistema electrónico se requiere de la etapa de alimentación del circuito (2.2.6.), que como su nombre lo dice, son circuitos diseñados para dar los voltajes requeridos para el funcionamiento de los componentes. Aquí se tienen dos fuentes de voltaje, una para los circuitos externos al microcontrolador dando un voltaje de 5V y otra fuente de 18V diseñada solo para los puentes H, ya que los motores deben manejar cantidades altas de voltaje y corriente para su funcionamiento.

## 2.2. Sistema electrónico

El sistema electrónico consta de siete etapas para manipular el sistema mecánico. Estas etapas se describen a continuación:

**Etapas de control.** Es la forma en cómo va a estar configurado del microcontrolador, para que al ejecutar el programa diseñado se realicen los diferentes procesos que se requieren en los sistemas específicos conectados a él.

**Etapas para los sensores ópticos.** Para saber la posición en la que se inicia la plataforma, se necesita de los sensores ópticos, los cuales mandan una señal dependiendo si hay o no obstrucción entre las separaciones que hay entre sus componentes. Si no hay obstrucción en la señal del sensor, éste mandará los 5V de entrada, en caso contrario, manda 0V.

**Etapas de registro de posición.** En esta etapa se registra la posición de los motores, por lo que se requiere de un componente externo al microcontrolador. Para esto se utiliza el circuito LS7266R1 que actúa como interfaz para que el microcontrolador envíe instrucciones para que el LS7266R1 obtenga el valor real de la posición alcanzada por los motores.

**Etapas impulsoras de corriente directa.** Como los motores deben ser capaces de mover una plataforma con un tanque lleno, es necesario utilizar motores que requieren grandes cantidades de corriente. Para lograr esta función, se utilizan puentes H que suministran la cantidad necesaria para los motores como también pueden controlar el movimiento de los mismos.

**Etapas impulsoras CD/CA.** Aquí se necesitan dispositivos que actúen como interruptores para que puedan prender o apagar una bomba y una válvula. Para esto, se utilizan relevadores que permiten manejar diferentes valores de voltaje a los del circuito de control como también manejar voltajes alternos o de directa dependiendo de los que utilicen los componentes mecánicos para funcionar.

**Etapas de alimentación del circuito.** Todos los dispositivos electrónicos necesitan de fuentes de alimentación de DC para poder funcionar, algunos de ellos se alimentan con diferentes valores de voltajes o se utilizan diferentes fuentes con el mismo valor para aislar los componentes y dar protección de los mismos.

**Algoritmo de medición, control y comunicación.** En esta etapa, se tiene el algoritmo que se diseña para que el microcontrolador pueda manejar las diferentes etapas antes mencionadas.

A continuación una explicación detallada de las etapas mencionadas y definidas en la figura 2.1. Los anexos A, B y C muestran los detalles de la implementación.

### 2.2.1. Etapa de control

Como se muestra en el esquema de la figura 2.5, el microcontrolador tiene en pines específicos componentes eléctricos como electrónicos para la realización de tareas específicas.

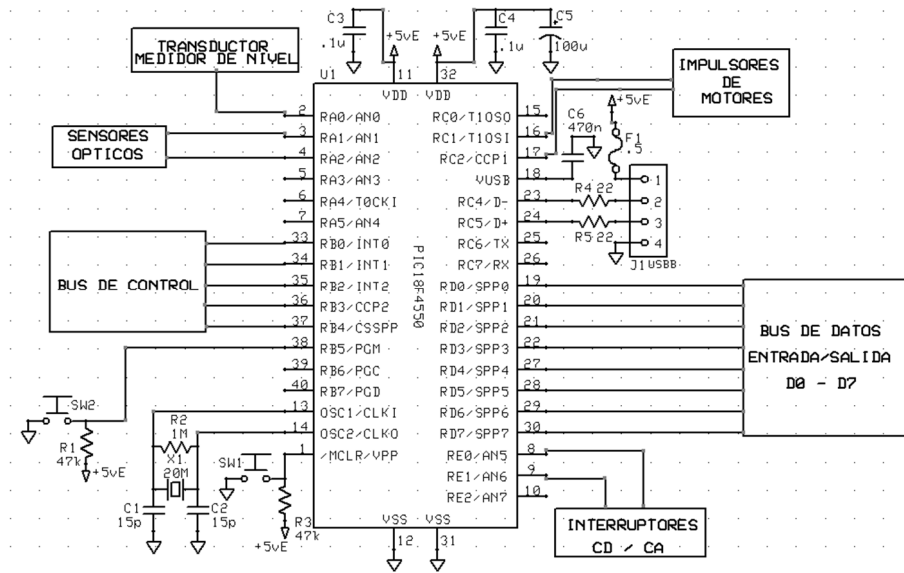


Figura 2.5. Esquema de configuración del microcontrolador.

Para que el microcontrolador pueda realizar su función se necesitan los componentes que se describen a continuación.

En cada entrada de voltaje tiene capacitores de bypass con el fin de eliminar las corrientes parasitas para que no interfieran con el funcionamiento del dispositivo. Por otra parte, consta de dos botones tipo “pushbutton” para establecer inicialmente el microcontrolador, un botón es para establecerlo de forma general y el otro es para que entre al modo de programación.

Para realizar los procesos programados, se debe poner un oscilador de cristal, en este caso se le incorpora uno de 20 MHz, así con cada ciclo que pase, el microcontrolador hará todo lo que se encuentra en la función principal de la programación.



Como el proyecto requiere la utilización de la computadora, entonces se debe activar el modo USB donde se transmiten y reciben los datos de ambos dispositivos, se necesitan de resistencias para en RC4 y RC5 para protección contra la corriente.

Los pines RD0 a RD7 van a recibir o enviar bits con la finalidad de que el dispositivo LS7266R1 los interprete. Los pines RD7 a RD5 tienen la capacidad de enviar y recibir, mientras que los pines RD4 a RD0 solo recibirán los valores que mande el LS7266R1. Si el microcontrolador recibe la señal enviada por el LS7266R1, éste las interpreta para después mandarlas hacia la computadora.

Los pines RC1 y RC2 son habilitados en el modo de PWM, con el fin de mandar las señales a cada LMD18201, los cuales tienen la capacidad de controlar el movimiento y la velocidad de cada motor por medio de esas señales.

Los pines RE0 y RE1 son habilitados como salidas digitales, con lo que solo mandarán un 1 digital o un 0 digital a los interruptores diseñados con relevadores.

El pin RA0 es una entrada de voltaje analógico con la finalidad de convertir la señal del sensor de nivel a una señal digital para procesarla y mandarla para su manipulación.

De igual forma los pines RA1 y RA2 son entradas analógicas, pero con la diferencia que de que dependerán de sensores ópticos para que puedan convertir y mandar el valor para su manejo en la aplicación.

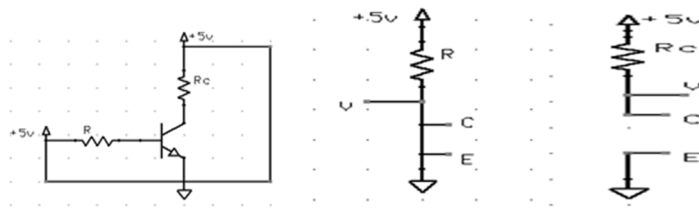
Por último RB0 a RB4 son pines habilitados como salidas para que sean enviadas al LS7266R1 y éste a su vez sea programado para escritura o lectura.

### **2.2.2. Etapa para los sensores ópticos**

Los sensores ópticos que se requieren para este proyecto son los H21A1 que consisten de un diodo acoplado y un fototransistor.

En general, el diodo es un dispositivo semiconductor que al ser polarizado en directa va a dar paso a la corriente comportándose como un corto circuito, mientras, si se polariza en inversa el dispositivo se comporta como un circuito abierto denegando el paso de la corriente. Para este circuito, el diodo es polarizado en directa con una resistencia de  $330\Omega$  para protección contra la corriente.

Por otra parte, el fototransistor está configurado para funcionar como un interruptor, es decir, solo va a estar en la región de corte y la región de saturación, como se muestra en la figura 2.6.



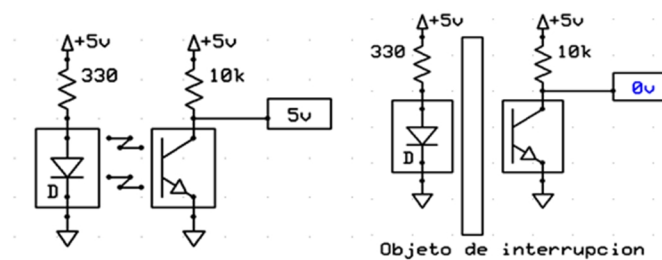
A) Fototransistor. B) Corte. C) Saturación.

Figura 2.6. Configuración del fototransistor.

Para que se encuentre en la región de saturación, se proporciona un voltaje positivo desde la base al emisor, provocando que el otro circuito formado por el colector-emisor se comporte como un corto circuito con una gran corriente que va del emisor al colector (aumentada por el circuito base-emisor), por lo tanto, se diseña el transistor con una resistencia de  $10\text{k}\Omega$  para proteger contra la corriente de saturación, véase en la figura 2.6.C.

Para que se encuentre en la región de corte, no debe de haber corriente desde la base al emisor, provocando que el circuito de salida emisor-colector se comporte como un circuito abierto y no permita el paso de corriente, véase en la figura 2.6.B.

Teniendo en cuenta los diseños de ambos componentes del H21A1, se procede a mostrar el funcionamiento de esta etapa como se muestra en la figura 2.7.



A) Sin interrupción. B) Con interrupción.

Figura 2.7. Configuración para los sensores ópticos.

En la figura 2.7.A el diodo, al ser alimentado, emite una luz infrarroja en línea recta recibiendo la el fototransistor, este al recibir la señal entrega el mismo voltaje con una corriente de saturación controlada por la resistencia de  $10\text{k}\Omega$ .

En la figura 2.6. La señal es interrumpida por un objeto, provocando que la base del fototransistor no reciba la corriente, comportándose como un circuito abierto emisor- colector, dando como salida un voltaje de 0V.

### 2.2.3. Etapa de registro de posición

En esta etapa se necesita de un decodificador de señal en cuadratura que proviene de los 2 codificadores ópticos y enviar la información de posición al microcontrolador para su procesamiento. Para esto, se ocupa el LS7266R1 que tiene dos contadores independientes y son manejados por canal X y canal Y. La configuración del circuito es la mostrada en la figura 2.8.

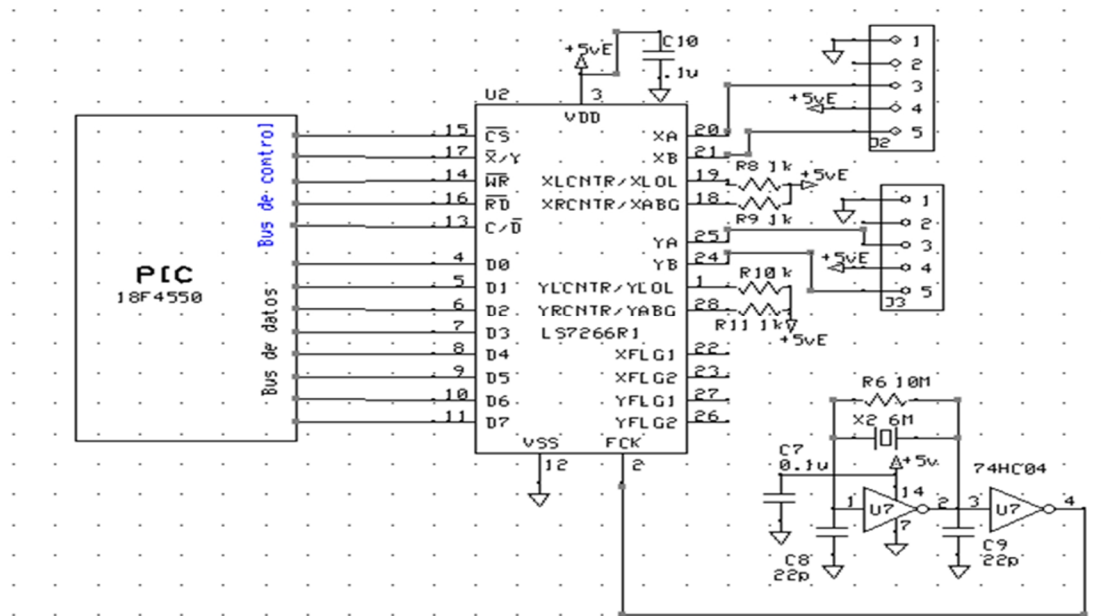


Figura 2.8. Configuración del LS7266R1.

Los pines 4 -11, como los pines 13- 17 del LS7266R1 deben de ir conectados al microcontrolador con el objetivo de tener un bus de datos y un bus de control para el manejo del dispositivo.

En el pin 3 de alimentación se coloca un capacitor de bypass de  $0.1\mu\text{F}$  para la eliminación de corrientes parasitas.

Se conectan los pines 1, 18, 19 y 28 al voltaje 5V con una resistencia de protección, con la finalidad de configurar al LS7266R1 en carga directa, pasando así los datos de los contadores a los registros sin verificar la cantidad de pulsos que mandan los codificadores ópticos.

En los pines 20 y 21 (Canales X) se coloca el codificador del primer motor, donde se registrarán los pulsos con cada movimiento de la flecha del motor.

En los pines 24 y 25 (Canales Y) se coloca el segundo codificador para el otro motor.

Se coloca un oscilador de onda cuadrada de 6MHz en el pin 2 FCK, tomando en cuenta la especificación del fabricante, donde la señal FCK para un conteo en cuadratura debe cumplir con la siguiente condición:  $f_{FCK} \geq 8f_{QA}$  en donde  $f_{QA}$  es la frecuencia de las transiciones de las señales en cuadratura.

Para este reloj se utiliza parte de la configuración llamada oscilador de Pierce que se muestra en la figura 2.9.

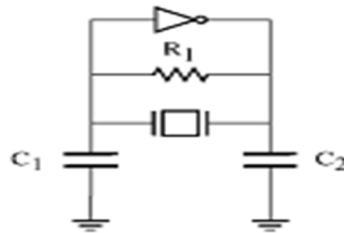


Figura 2.9. Oscilador de Pierce.

Como se muestra en la figura 2.9, consiste de un inversor CMOS con una resistencia de polarización que fija un punto de trabajo en continua para el cual la ganancia (relación entrada-salida) es máxima. Conectado entre las terminales del inversor un cristal de cuarzo, resultarán osciladores de tensión a la frecuencia de resonancia determinada por el cristal. Las dimensiones de los transistores NMOS y PMOS, los valores de resistencia y condensadores externos determinarán la amplitud de oscilaciones, alrededor del punto de trabajo en continua [Rubio, 2003].

Para el inversor se utiliza un inversor 74HC04 que maneja más de 8MHz. El objetivo de la utilización de los inversores es para darle mayor estabilidad al oscilador de cristal, quitando el ruido que pueda generar.

Se coloca una resistencia de  $10M\Omega$ , un cristal de 6MHz y por recomendación del fabricante, se utilizan capacitores de 22pF entre la salida y entrada del primer inversor.

En el pin de alimentación se coloca un capacitor de bypass de  $0.1\mu F$ . Por último, se coloca el primer inversor para actuar como amplificador buffer de salida encargado de conformar la señal.

Para operar el circuito de interfaz desde el microcontrolador se requieren las tareas que se describen a continuación.

## Configuración lógica del LS7266R1

Para que el dispositivo realice las tareas de escritura necesarias en sus registros internos se necesitan programar las funciones de las tablas 2.1 y 2.2.

FUNCIÓN	HEXADECIMAL	C/D RD WR x/y CS
Deshabilita CS	0x01	0 0 0 0 1
Habilita la escritura	0x18	1 1 0 0 0
Deshabilita CS	0x01	0 0 0 0 1

Tabla 2.1. Bus de control en el modo de escritura del canal X.

FUNCIÓN	HEXADECIMAL	C/D RD WR x/y CS
Deshabilita CS	0x01	0 0 0 0 1
Habilita la escritura	0x1a	1 1 0 1 0
Deshabilita CS	0x01	0 0 0 0 1

Tabla 2.2. Bus de control en el modo de escritura del canal Y.

Para que el dispositivo realice las tareas de lectura necesarias en sus registros internos se necesitan programar las funciones de las tablas 2.3 y 2.4.

FUNCIÓN	HEXADECIMAL	C/D RD WR x/y CS
Deshabilita CS	0x01	0 0 0 0 1
Habilita lectura	0x06	0 0 1 1 0
Deshabilita CS	0x01	0 0 0 0 1

Tabla 2.3. Bus de control en el modo de lectura del canal Y.

FUNCIÓN	HEXADECIMAL	C/D RD WR x/y CS
Deshabilita CS	0x01	0 0 0 0 1
Habilita lectura	0x04	0 0 1 0 0
Deshabilita CS	0x01	0 0 0 0 1

Tabla 2.4. Bus de control en el modo de lectura del canal X.

## Opción de registros

El bus de datos puede trabajar como lectura de valores o como escritura de operaciones a realizar en el dispositivo. En este caso, los tres primeros bits más significativos indican el registro y los restantes las opciones.

Para el registro RLD las configuraciones son las mostradas en las tablas 2.5 a 2.8.

REGISTRO	OPCIONES	HEXADECIMAL
D7 D6 D5	D4 D3 D2 D1 D0	
1 0 0	0 0 1 1 0	0x86
	Ninguna operación	

	-----  Reset E  -----  Ninguna operación	
--	---	--

Tabla 2.5. Resetear E (La bandera de error).

REGISTRO	OPCIONES	HEXADECIMAL
D7 D6 D5 1 0 0	D4 D3 D2 D1 D0 0 0 0 0 1  Ninguna operación  -----  Reset E  -----  Ninguna operación	0x81

Tabla 2.6. Resetear BP (Apuntador para saber en qué byte va a escribir).

REGISTRO	OPCIONES	HEXADECIMAL
D7 D6 D5 1 0 0	D4 D3 D2 D1 D0 1 1 0 0 0  Ninguna operación  -----  Ninguna operación  -----  Transfiere de PR0 a PSC	0x98

Tabla 2.7. Seleccionar reloj filtrado.

REGISTRO	OPCIONES	HEXADECIMAL
D7 D6 D5 1 0 0	D4 D3 D2 D1 D0 0 0 0 1 0  Ninguna operación  ----  Resetea CNTR  -----  Ninguna operación	0x82

Tabla 2.8. Resetea CNTR (Resetea el contador).

Para el registro CMR la configuración es la mostrada en la tabla 2.9.

REGISTRO	OPCIONES	HEXADECIMAL
D7 D6 D5 1 0 1	D4 D3 D2 D1 D0 0 1 0 0 0   Cuenta binaria  -----  Cuenta normal  -----  Cuadratura por 1	0xa8

Tabla 2.9. Lectura sin cuadratura, binario y normal.

Para el registro IOR la configuración es la mostrada en la tabla 2.10.

REGISTRO	OPCIONES	HEXADECIMAL
D7 D6 D5 1 1 0	D4 D3 D2 D1 D0 0 0 0 0 1  Habilita entradas Ay B  ---  FLG1 transporta la salida	0xc1

	El pin FLG2 pide prestada la salida	
--	-------------------------------------	--

Tabla 2.10. Selecciona las entradas A y B.

### 2.2.4. Etapa impulsora de corriente directa

En esta etapa se utiliza el componente LMD18201, el cual es un puente H que tiene como función el proporcionar la corriente necesaria para mover el motor, como también darle el direccionamiento necesario.

En la figura 2.10 se muestra la composición interna del LMD18201.

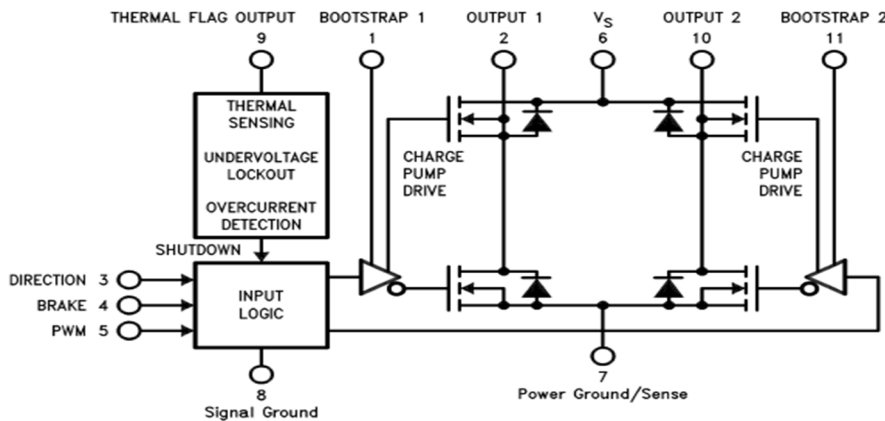


Figura 2.10. Descripción funcional del LMD18201.

Los pines tales como el “bootstrap” 1, 2 y las salidas “output” 1 y 2 manipularán al motor por medio del direccionamiento de la corriente.

El pin 3 es el de direccionamiento, el cual va a decidir hacia donde gira el motor. El pin 5 que es la señal que mantendrá la magnitud de la señal del dispositivo. Y por último, se encuentran los pines de alimentación para el funcionamiento del dispositivo.

Para la construcción del circuito, primero se plantean las recomendaciones del fabricante para el componente.

Se recomienda colocar capacitores de bypass en las entradas de voltaje para la eliminación de inductancias parasitas. El valor recomendado es de  $1\mu\text{F}$ , pero si se manejan altas corrientes entonces se procede al valor del capacitor de  $100\mu\text{F}$  por cada Amper.

En el pin de “bootstrap” se recomienda un capacitor de  $10\text{nF}$  para cargar rápidamente las cargas parasitarias del dispositivo de potencia que resulta en tiempos más rápidos de subida, permitiendo también frecuencias mayores de  $500\text{kHz}$ .

Teniendo en cuenta las recomendaciones, el circuito queda de la siguiente manera.

- Se utiliza una fuente de 18V para el movimiento de los motores, pero como la cantidad de corriente puede ser de hasta 3A, entonces se pone un capacitor de 330  $\mu$ F (tomando la recomendación antes mencionada).
- En el pin 5 se coloca un voltaje de 5V, como también un capacitor de bypass de 1 $\mu$ F.
- En el pin 1 y 2 como también el pin 11 y 10 se coloca el valor del capacitor de 10nF.
- La señal de PWM se coloca en el pin 3, teniendo el siguiente funcionamiento: Con el 50% del ciclo de trabajo de la señal PWM el motor se quedará estático. Arriba del 50% del ciclo de trabajo la corriente fluirá de la salida 1 a la salida 2. Menor al 50%, entonces la corriente fluirá de la salida 2 a la salida 1.
- Por último, en las aplicaciones de manejo de motores proporcionada por el fabricante se recomienda un voltaje de 5V en el pin 9 protegido con una resistencia de 10k $\Omega$ . Este pin ofrece una señal que es de advertencia térmica, si el dispositivo alcanza una temperatura igual o mayor de 170°C se apagará.

La configuración descrita se muestra en la figura 2.11.

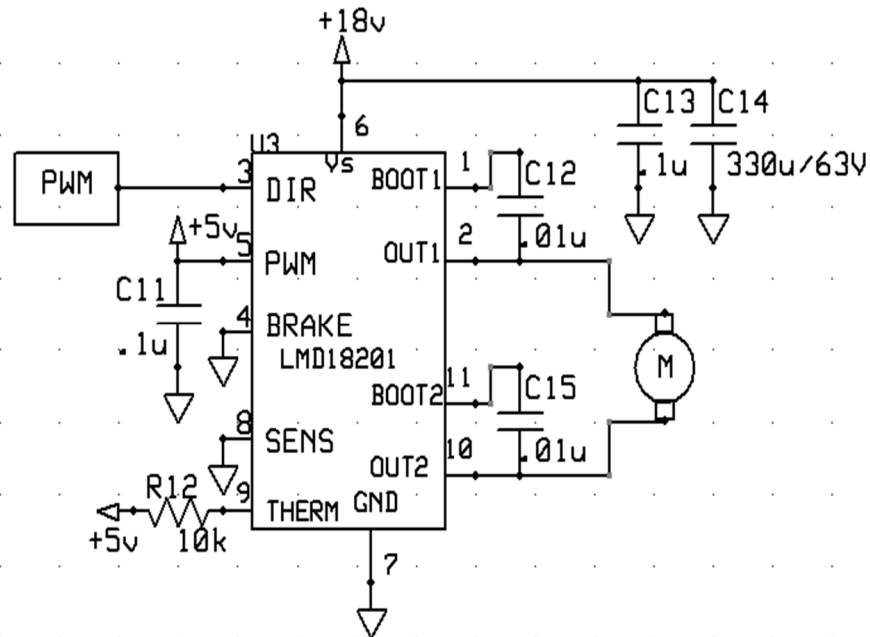


Figura 2.11. Configuración del LMD18201 utilizada.



### 2.2.5. Etapa impulsora CD/CA

Como se presenta esta etapa se requiere de un interruptor para el apagado y encendido de una bomba y una válvula, por lo que se utilizan dos relevadores para que logren esta función sin afectar al microcontrolador.

Como ya se mencionó, el relevador es un dispositivo que cambia de estado de una terminal a otra mediante la inducción de corriente a la bobina. Apreciando la figura 2.12, el pin 1 se mantiene en el pin 4 si no se le proporciona corriente de inducción a la bobina, en caso contrario, se cambia al pin 3, lo que significa que el pin 3 es normalmente abierto.

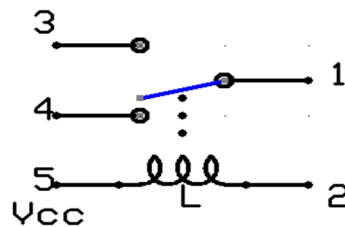


Figura 2.12. Diagrama interno de un relevador.

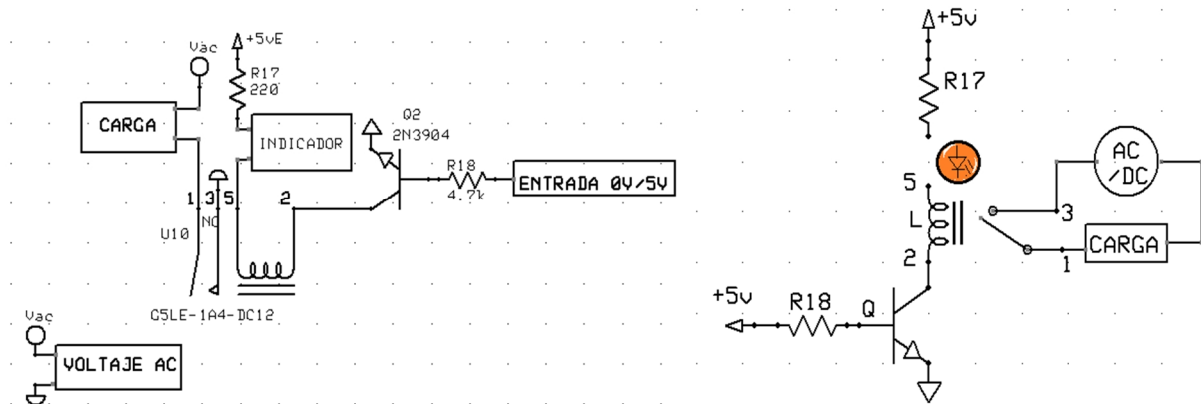
Para los fines del proyecto, no se utiliza el pin 4, teniendo así un solo interruptor por cada relevador.

Como el microcontrolador tiene que mandar la señal para que funcione el relevador, es necesario colocar un transistor NPN con la configuración de interruptor entre ambos circuitos, esto es, para tener protección entre el microcontrolador y el relevador en caso de una subida de corriente, como también permitir el uso de relevadores con tensiones de alimentación de la bobina diferentes respecto a la tensión de trabajo del circuito de control.

Se coloca una resistencia en la base del transistor de  $4.7k\Omega$  para disminuir la corriente y que no aumente demasiado en el circuito colector-emisor. La resistencia R17 se escoge con un valor de  $220\Omega$  solo para protección del transistor.

Por cuestiones prácticas se coloca un LED entre la bobina del relevador y la resistencia de la fuente para comprobar el funcionamiento del circuito.

Ya terminado el diseño se muestra en la figura 2.13 el circuito final de esta etapa como su circuito equivalente.



A) Esquema de la configuración. B) Circuito equivalente.

Figura 2.13. Configuración utilizada para la etapa impulsora CD/CA.

La entrada 0V/5V es la salida proporcionada por el microcontrolador, ésta va a entrar en la base del transistor con la finalidad de activar o apagar la bobina del relevador. Cuando la salida del microcontrolador sea un “1” digital (5V), el transistor se comportará como un circuito cerrado, que va a ser indicado con el LED haciendo circular corriente hacia la bobina que inducirá al pin 1 y pin 3 a entrar en contacto, provocando la circulación de la corriente de la fuente de voltaje AC/DC hacia la carga. Cuando la salida del microcontrolador sea un “0” digital (0V), el transistor se comporta como un circuito abierto, lo cual no alimenta al relevador manteniéndolo como un interruptor abierto. Esto se comprueba con el LED que se mantendrá apagado.

### 2.2.6. Etapa de alimentación del circuito

En el diagrama de la figura 2.14 se explica la conexión de las fuentes con los dispositivos.

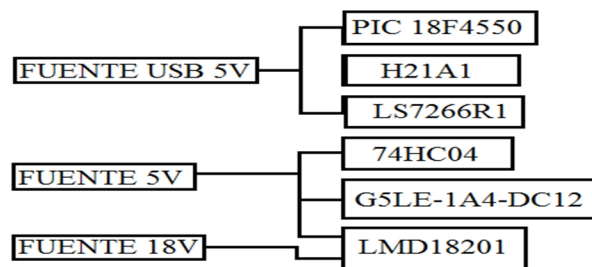


Figura 2.14. Fuentes de alimentación del sistema electrónico.

Como se ha mostrado, el sistema requiere de 3 fuentes de voltaje, la primera fuente es proporcionada de la conexión USB de la computadora con el microcontrolador y por diseño se define alimentar a los sensores y al

contador con la misma fuente. Esto es porque los dos dispositivos no manejan voltajes altos y es difícil que puedan tener una subida de voltaje.

Se decide diseñar otra fuente de voltaje con el mismo valor del voltaje USB para el 74HC04, el relevador y el puente H, debido a que no dependen del microcontrolador, por lo que se trata de proteger a los cambios de corriente que puedan tener y así no afectar al dispositivo y a la computadora.

Por último, el puente H también requiere de corrientes altas para poder mover un motor y por cuestiones prácticas se utiliza un voltaje de 18V teniendo en cuenta que se tiene que mover una plataforma con un tanque lleno encima.

Para el diseño de las fuentes, cabe mencionar que se requieren de las etapas que se muestran en la figura 2.15.

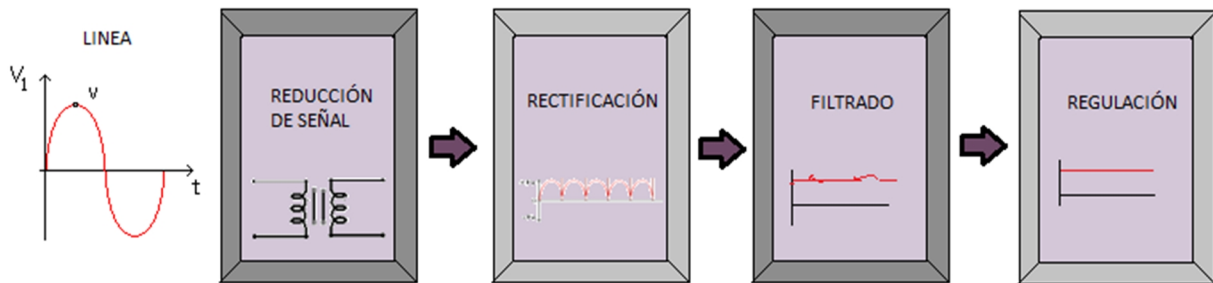


Figura 2.15. Etapas de una fuente de voltaje.

En la primera etapa se necesita del transformador, ya que su función es reducir el voltaje de la línea (AC) al nivel que pueda funcionar el circuito sin ser dañado. El transformador puede realizar la función anterior, debido a que consta de dos bobinas enrolladas alrededor de un núcleo, como se muestra en la figura 2.16.

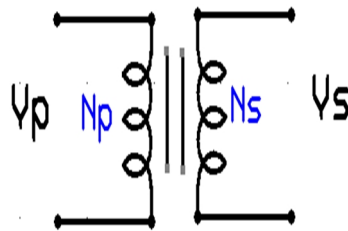


Figura 2.16. Esquema interno del transformador.

Como éstos embobinamos están montados en el mismo circuito magnético, el flujo magnético en ambos será el mismo, teniendo así la siguiente relación entre el voltaje con el número de vueltas en el primer y segundo embobinado.

$$\frac{V_p}{N_p} = \frac{V_s}{N_s}$$

Donde  $V_p$  es el voltaje primario.

$V_s$  es el voltaje en el secundario.

$N_p$  el número de vueltas en el primario

$N_s$  el número de vueltas en el secundario.

Gracias a esta relación se puede obtener los números de vueltas que requiere el transformador, solo si se desea diseñar y construir.

Cabe mencionar que todos los transformadores comerciales indican el voltaje alterno con el valor RMS, por lo que se requiere de la formula  $V_{pico} = \sqrt{2} V_{RMS}$  para saber cuál debe ser la salida real del devanado secundario.

Ya reducido el voltaje, se requiere de la separación de la señal alterna, es decir, tener la separación de sus ondas positivas y negativas. Esto se hace mediante diodos, los cuales conducen con polarización directa, si se tienen con polarización inversa no conducen.

Para una mayor eficacia se utilizan cuatro diodos en configuración de rectificador de onda completa, ver figura 2.17, donde al entrar el semi-ciclo positivo de la señal se activará el diodo entre la terminal 2 y 1, como el diodo entre 3 y 4, dejando el paso de esta parte de la señal. Sin embargo, con el otro semi-ciclo negativo se activan los otros dos diodos produciendo un semi-ciclo positivo en la carga.

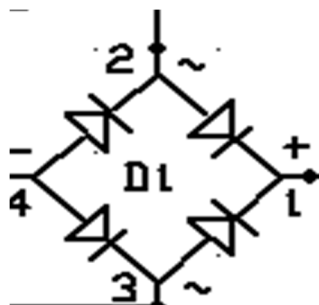


Figura 2.17. Rectificador de onda completa.

Esto significa que se aprovecharán los dos semi-ciclos de la señal como se muestra en la figura 2.18.

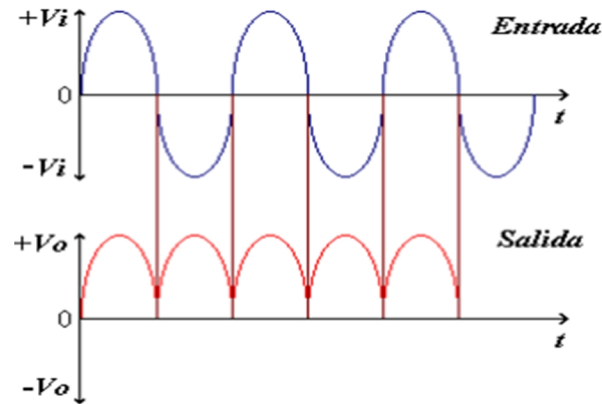


Figura 2.18. Señal rectificada.

Una vez explicadas las primeras etapas generales de las fuentes, a continuación se explican los circuitos de cada una de ellas.

### Fuente de 5V

Para la fuente de voltaje de 5V se utiliza un transformador de 127 a 12Vrms con 1.2 A.

Se coloca un fusible de 1 A en la línea para protección del circuito.

Se utiliza el RS607 que es un rectificador de onda completa.

Para la etapa de filtrado se utiliza un capacitor electrolítico debido a que son útiles para bajas frecuencia. Pero para su valor, se puede obtener la capacitancia con respecto al porcentaje de rizado de la siguiente manera:

$$C = \frac{100}{4fR_cP_r(\%)}$$

Donde  $R_c$  = carga.

$P_r$  = porcentaje de rizado.

$f$  = frecuencia del circuito.

$C$  = capacitancia.

Esta fórmula es para un rectificador de onda completa.

Considerando en éste circuito el 5% deseable en el rizado, la frecuencia de la línea de 60Hz y una carga medida en el laboratorio de 48Ω y substituyendo los valores, se tiene:

$$C = \frac{100}{4(60)(48)(5)} = 1.74\text{mF}$$

por lo que se opta a poner un capacitor comercial de 2200μF.

La siguiente etapa es el regulador de voltaje, considerando que los dispositivos funcionan con 5V, se opta por poner un regulador de +7805, manteniendo constante éste voltaje. Se colocan dos capacitores de 100 $\mu$ F y .1 $\mu$ F en paralelo con el regulador para la disminución del voltaje de rizo y del ruido que se pueda provocar.

Por último se procede a poner un LED en la salida de la fuente con una resistencia de protección de 470 $\Omega$ . Esto es por cuestiones de prueba, en dado caso de que no funcione algún componente de la fuente diseñada, el LED lo indicará. La configuración final es la mostrada en la figura 2.19.

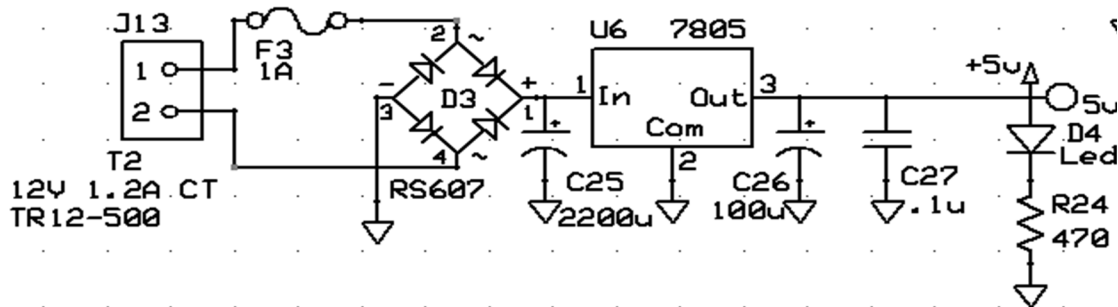


Figura 2.19. Fuente de 5V.

### Fuente de 18V

Para este circuito se utiliza un transformador de 127 a 24Vrms con 3 A, debido a que los motores manejan altas corrientes.

Se coloca un fusible de 5 A en la línea para protección del circuito.

Se utiliza el RS607 que es un rectificador de onda completa.

En la etapa de filtrado se maneja la misma fórmula de la fuente anterior con la diferencia del valor de la carga que se obtuvo de 62 $\Omega$ , quedando como:

$$C = \frac{100}{4(60)(62)(5)} = 1.35\text{mF}$$

Por lo tanto se decide poner el mismo capacitor de la fuente anterior, capacitor de 2200 $\mu$ F.

En la etapa de regulación, el dispositivo +7818, por cuestiones de fabricación, no puede soportar ni mandar una corriente mayor a 1 Amper, por lo que se utiliza un TIP 42 PNP el cuál es el transistor de potencia que suministrará la corriente necesaria que no puede dar el regulador.

Para que pueda funcionar el TIP, se requiere entre el emisor y su base una tensión de 0,7V y una corriente menor a 0.5A.

Se debe de colocar una resistencia entre el emisor y la base, debido a que mientras no encienda el TIP, la corriente se irá hacia la resistencia y por lo tanto al regulador, que a su vez, depende de la carga que exista en la salida de la fuente.

Cuando el TIP encienda, se tendrá en el regulador y en la resistencia 20 de la figura 2.20:

$$I_{R20} = \frac{0.7}{10} = 7\text{mA}$$

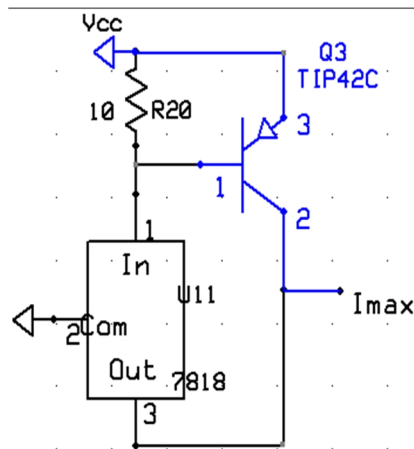


Figura 2.20. Configuración del TIP42.

La demás corriente solicitada por la carga la entregará el TIP.

Por otra parte, en las hojas de datos del fabricante, se recomienda colocar capacitores de bypass entre las terminales del regulador con el objetivo de optimizar la estabilidad y la respuesta transitoria, como también deben de estar colocados lo más cerca de las terminales. Los capacitores utilizados son de  $100\mu\text{F}$  y el de  $0.1\mu\text{F}$  sólo es para el rizo. Al igual que la fuente anterior, se coloca un LED con su resistencia de  $2.2\text{K}\Omega$  para prueba. La configuración final se muestra en la figura 2.21.

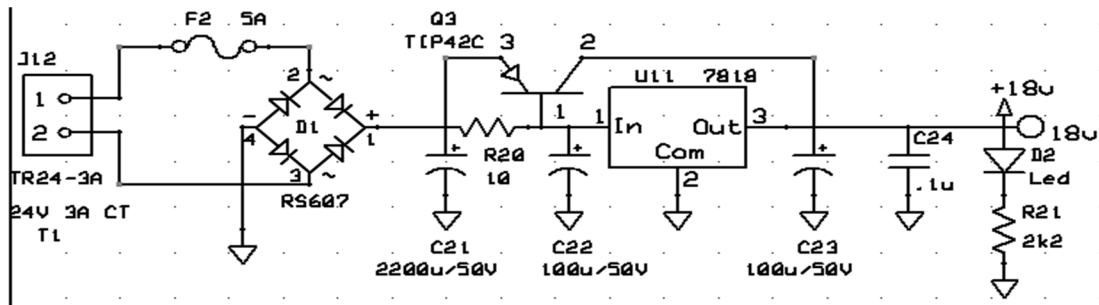


Figura 2.21. Configuración final de la fuente de 18V.

### 2.2.7. Algoritmo de medición, control y comunicación

Para poder programar el microcontrolador PIC18F4550, se ocupó el software de programación MPLAB el cual está basado en el lenguaje C.

La programación se hizo de la siguiente manera:

- Se debe de activar la función de USB para que se tenga la comunicación entre la computadora y la PIC.
- Se define el número de datos que se enviarán a la computadora y los que recibirá el PIC18F4550.
- Se configuran las funciones de escritura y lectura de cada canal X/Y del dispositivo externo LS7266R1 para conocer la posición actual de los motores. Se mandan y reciben los datos por los puertos RD0-RD7 para la escritura y lectura del dispositivo, como también se utilizan los pines RB0-RB4 como bus de control, solo configurados para mandar bits.
- Se define la configuración en la que los pines estarán trabajando.
- Se configuran las variables necesarias para el registro de los valores mandados por el LS7266R1 del canal X y el canal Y, para poder ser leídos en la computadora. Éste dispositivo cuenta con un apuntador que por cada byte se incrementa mandando así el valor por separado, debido a que sus valores son de 24bits. Por otro lado, el microcontrolador solo puede mandar 1 byte a la vez, entonces solo se procede al envío de los datos por separado.
- Se habilitan los pines de conversión analógica para que se pueda hacer lectura de los dos sensores, como también la lectura del transductor para el registro del nivel del tanque. En el microcontrolador seleccionado, la función del convertidor analógico-digital utiliza máximo 10bits, tomando ese número como la equivalencia al voltaje máximo de entrada de cada pin, por lo que se requiere de la separación de éstos para poder hacer el envío de los hacia la computadora. Con los siguientes comandos se logra la separación de cada byte.

```
res1= lectura& 0xFF;
```

```
res2= lectura>> 8;
```

Donde *res1* y *res2* son las variables a enviar y “*lectura*” es la variable que contiene el valor digital de la señal analógica.

- Se utilizan dos pines configurados como salidas digitales, generando ceros y unos digitales para el prendido y apagado de una bomba y una



válvula. Esto es logrado con la función TOGGLE que tiene el lenguaje en C.

- Para la manipulación del movimiento del motor se requiere recibir los datos de la PC, pero como el LS7266R1 maneja 24 bits para el movimiento del motor, entonces se procede a concatenar los datos enviados con el siguiente código:

```
rec=(int32)param1+(256*(int32)param2)+((int32)param3*65536);
```

Donde *rec* es la variable de concatenación.

*Param* 1, 2 y 3 son las variables que reciben el dato mandado de la PC.

### Control de lazo cerrado

Se configura la parte de control para que cada motor pueda alcanzar la posición especificada o “Set Point”. Para poder lograr éste funcionamiento, se requiere de la siguiente fórmula:

$$m(k)=m(k-1)+e(k)[KP+KI+KD]-e(k-1)[KP-2KD]+e(k-2)KD$$

La cual es una formula general que conjunta las constantes derivativas, proporcionales e integrales.

Para propósitos de ésta tesis se decide probar con cada uno de estos controles PID, quedando desde un principio como mejor opción el controlador proporcional.

El controlador proporcional varía instantáneamente con el error y alcanza un valor estacionario cuando tienen el mismo valor. Con esto, la formula anterior se modifica a la siguiente:

$$m(k)=m(k-1)+e(k)[KP]$$

Donde:

*m(k)* Voltaje discreto actual generado por el controlador.

*m(k-1)* Voltaje discreto anterior generado por el controlador.

*e(k)* Error discreto actual.

Ahora, teniendo en cuenta como se hace el cálculo, entonces se procede a programarlo.

Para el cálculo del error se requiere obtener el valor real de la posición en la que se queda el motor con respecto al valor que se desea, el cual es enviado por la PC a través del puerto USB.

$$e(k) = \text{posición deseada} - \text{posición real}$$

Para el cálculo de la constante proporcional se obtuvo un valor de 0.6 de manera experimental.

$$KP=0.6$$

Para el cálculo del voltaje discreto anterior, el dispositivo LMD18201 modifica el movimiento del motor por medio de la señal PWM, donde al estar dicha señal al 50% del ciclo de trabajo, el motor se queda estático, menor al 50% el motor se mueve hacia un lado y mayor al 50% se mueve al otro. Por otra parte, la señal PWM está dada por el microcontrolador PIC18F4550 que en sus hojas de especificaciones solo tiene un alcance de 1023, siendo así, el decimal 511 el 50% del ciclo de trabajo de la señal. Teniendo en cuenta todo esto, el voltaje discreto anterior tendrá el valor de 511, inicialmente para posteriormente irse actualizando.

$$m(k-1) = 511$$

El objetivo es que cuando el error alcance el valor del 511 el motor se quedará estático, quedando así en la posición deseada.

Por último, para que esto tenga efecto en el motor,  $m(k)$  debe ser la señal del ciclo de trabajo que manejara el PWM, quedando la fórmula para la programación como:

$$\text{Ciclo} = 511 + (0.6)(\text{posición deseada} - \text{posición real})$$

### **Diagramas de flujo**

A continuación, en las figuras 2.22 a 2.25 se muestran los diagramas de flujo de la programación del PIC18F4550.

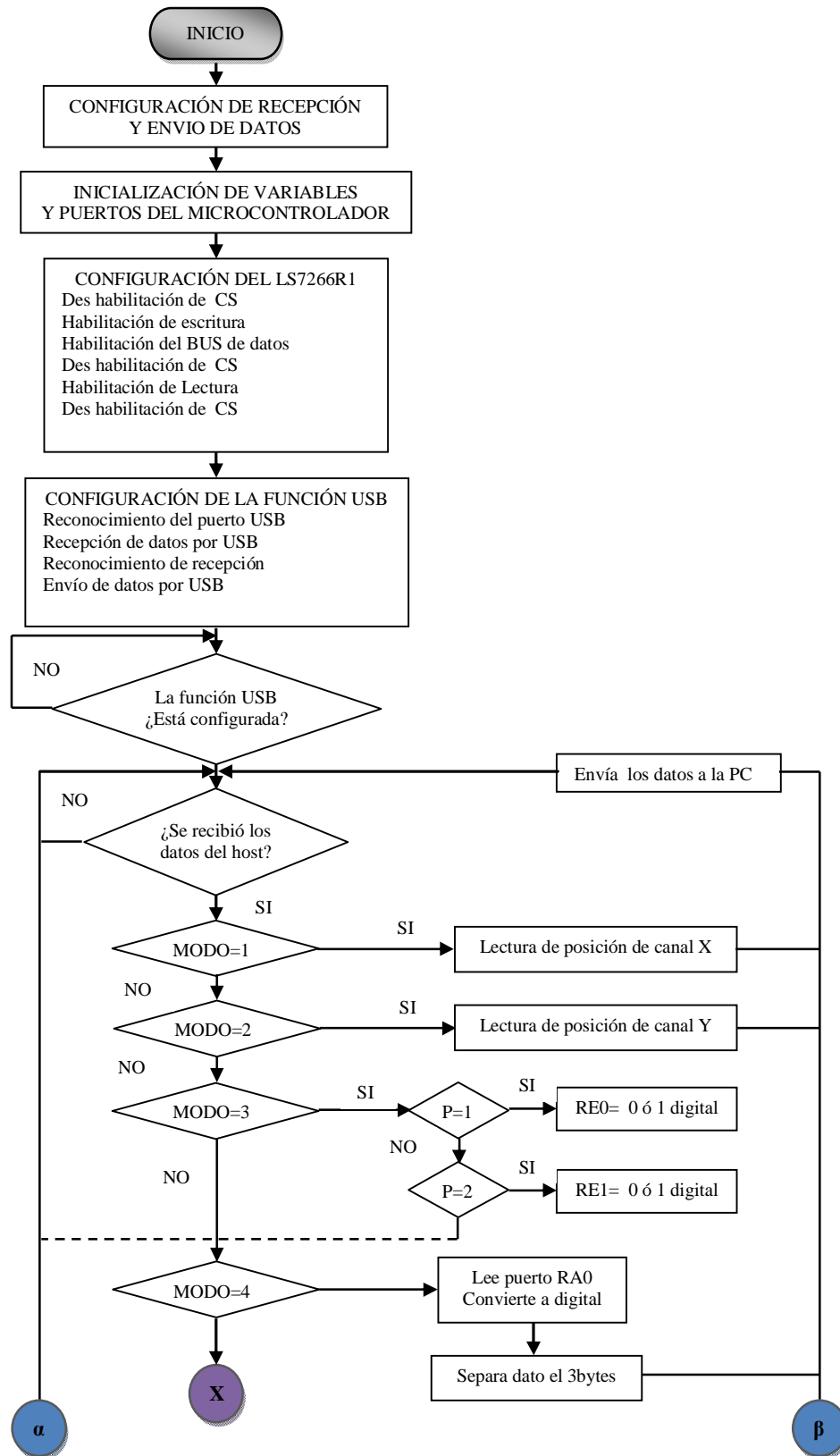


Figura 2.22. Rutina principal.

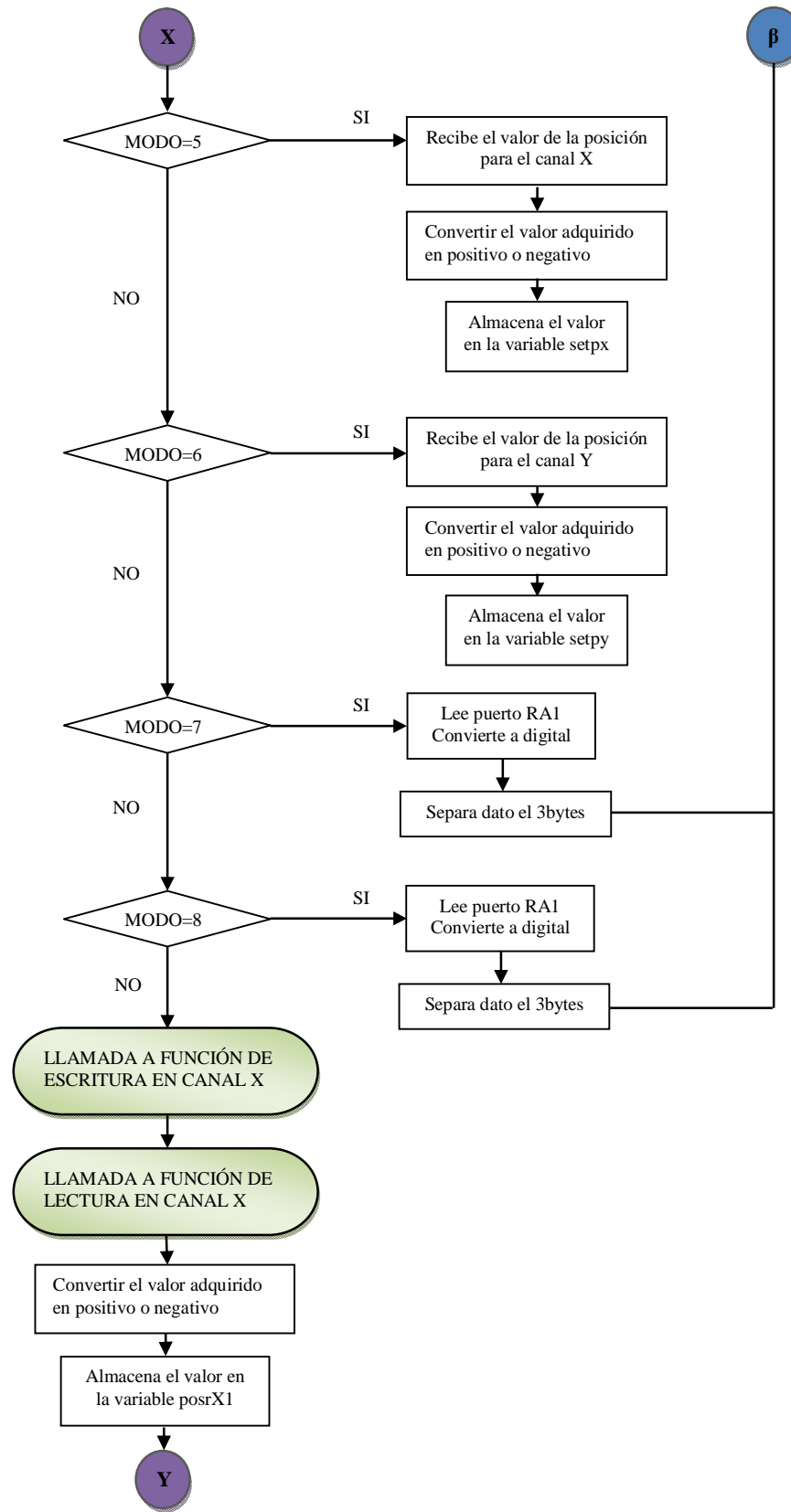


Figura 2.23. Rutina principal, continuación.

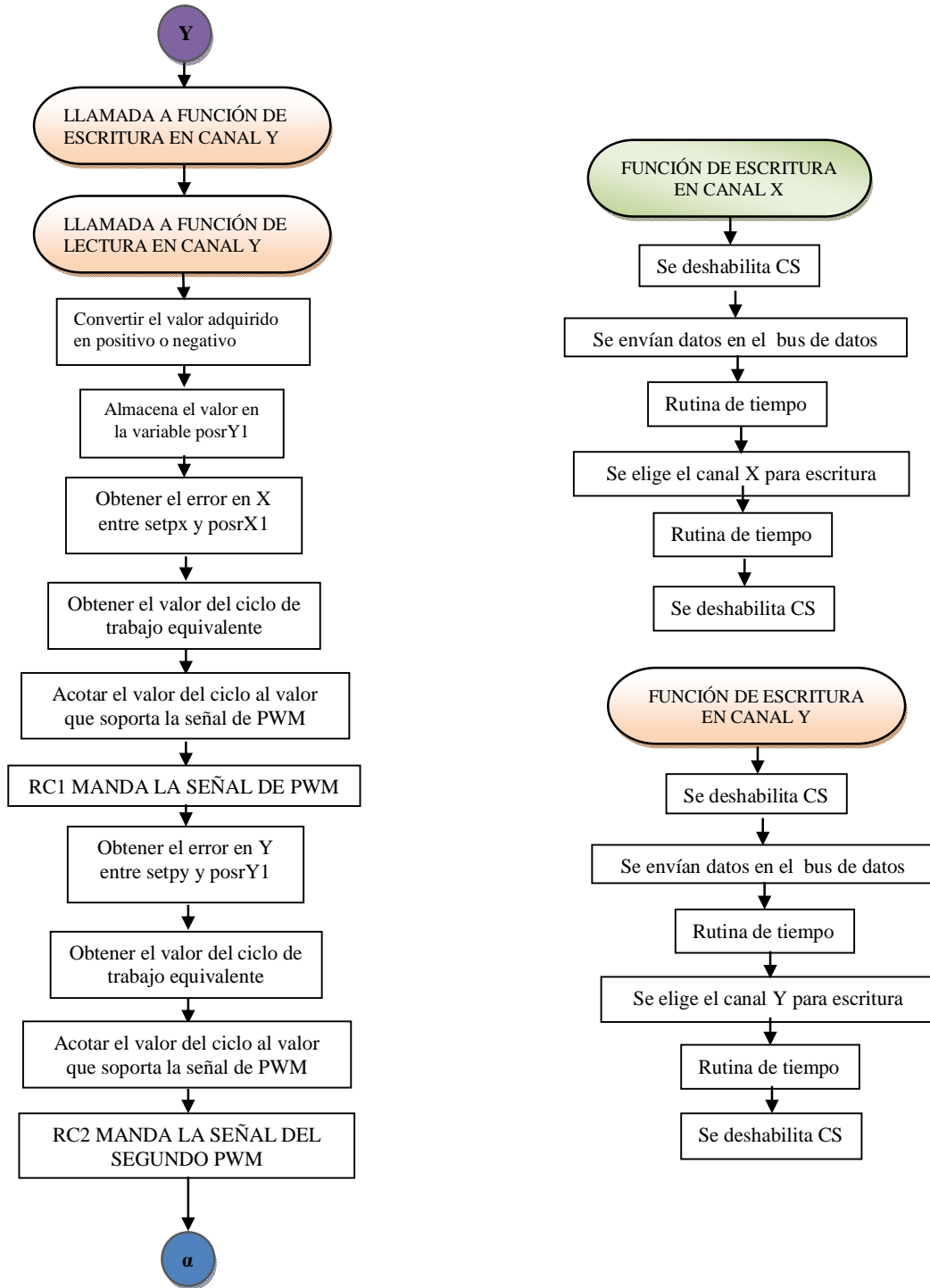


Figura 2.24. Rutina principal, continuación.

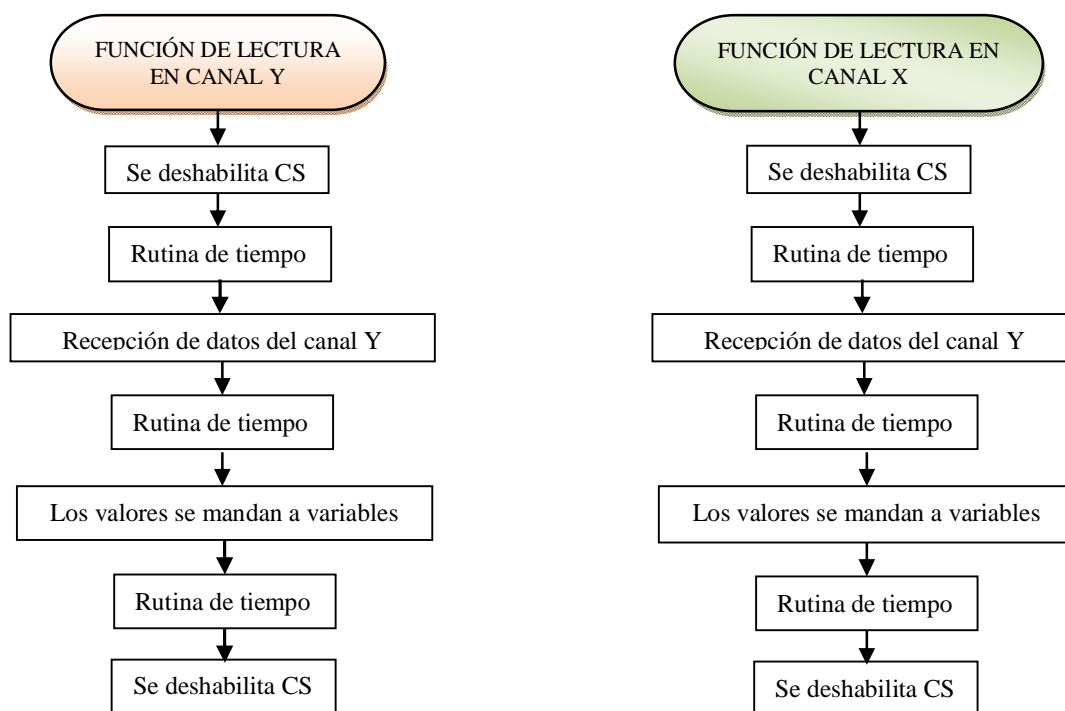


Figura 2.25. Rutina principal, continuación.

### 2.3. Software de operación

El microcontrolador PIC18F4550 es el encargado de manejar diferentes dispositivos electrónicos, como también obtener la adquisición de datos por medio de éstos.

Para que se puedan visualizar los datos obtenidos, se necesita de una interfaz gráfica que establezca la comunicación entre la computadora y el microcontrolador.

Para esta tesis, la herramienta de desarrollo de la interfaz gráfica es Visual C++ la versión 2006 que permite la creación de aplicaciones utilizando componentes del mismo sistema operativo (Windows XP), por medio de la librería MFC.

El sistema operativo que se utiliza es Windows XP, debido a que se necesita de un controlador para el reconocimiento del sistema electrónico, que cuando se instala en Windows 7 y 8 se encuentran problemas de comunicación.

### 2.3.1. Descripción general del proyecto

Para la programación en Visual C++ se requirieron de las clases mostradas en la figura 2.26.

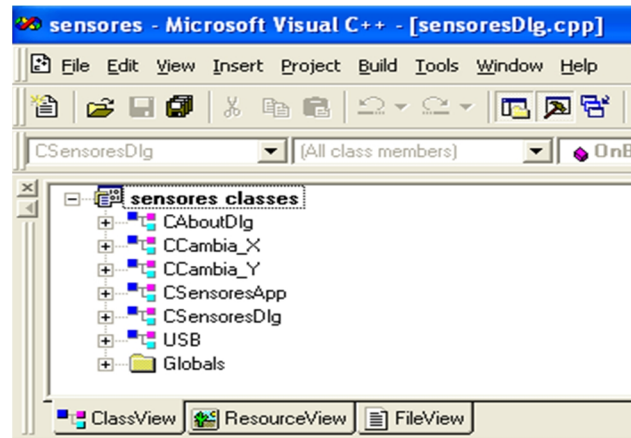


Figura 2.26. Clases del programa en Visual C++.

El programa en Visual C++ para esta tesis tiene el nombre de “sensores”, donde también se crean diferentes clases que se hacen referencia entre ellas.

- Clase CAboutDlg, es la clase utilizada para desplegar el típico diálogo “Acerca de...”.
- Clases CCambia\_X y CCambia\_Y se crean en el momento en que se crean dos cajas de diálogos dependientes de la principal.
- Clase CSensoresApp contiene todos los archivos de cabeceras de la aplicación del proyecto creado.
- Clase CSensoresDlg es la esencia del programa. Se colocan todas las variables relacionadas con los botones y etiquetas de edición proporcionadas por las cajas de diálogos, como también el nombre de las funciones para cada botón a utilizar. Aquí también se inicializan las interrupciones que se utilizan.
- Por último se tiene la Clase USB, donde se programa lo que se va a enviar y recibir del sistema electrónico.

Como se muestra en la figura 2.27, se tiene la parte gráfica del programa que puede utilizar el usuario, teniendo a su alcance el manejo de botones que controlan una tarea específica y visualización de datos que son las etiquetas donde se puede leer el dato o ingresarlo.

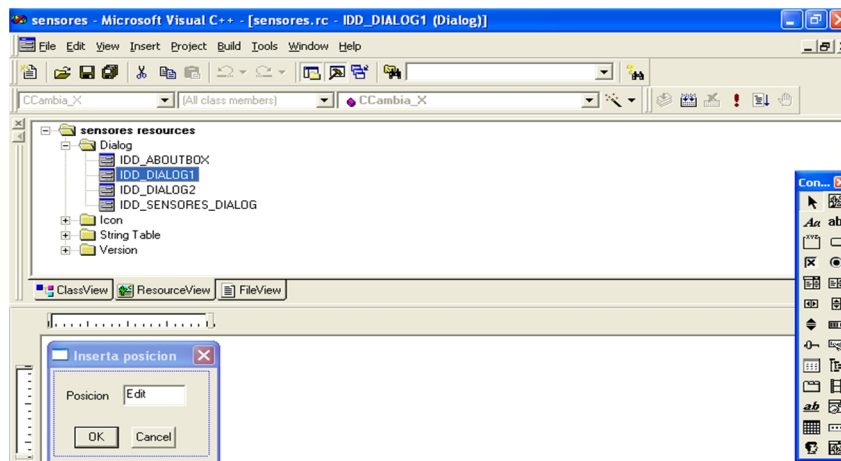


Figura 2.27. Cajas de diálogo.

El programa tiene una relación entre las clases y las cajas de diálogo, por lo que si se crean más cajas de diálogos, entonces también se crea su respectiva clase. Para esta tesis, se requieren de tres diálogos, donde los dos últimos dependen de la caja de diálogo principal que se crea al momento de ejecutar el programa.

### 2.3.2. Descripción en el ámbito del operador

La aplicación de Visual C++ tiene componentes de control y visualización manejados por el programa como “button” y “edit” (botón y caja de edición), para que el operador pueda utilizar las funciones requeridas.

La aplicación consta de tres cajas de diálogo, siendo la principal la que contiene las funciones relevantes para la aplicación.

Como se muestra en la figura 2.28 de la aplicación, el sistema tiene dos apartados.



Figura 2.28. Pantalla principal.



## **Apartado de controles**

Los controles que puede manejar el operador se describen a continuación.

- Button 3 BOMBA. Se comporta como un interruptor de prendido/apagado, generando el llenado del tanque.
- Button 4 VALVULA. Se comporta como un interruptor de prendido/apagado, con la finalidad de abrir y cerrar una válvula electrónica que permita el vaciado del tanque.
- Button 1 Inserta.., Este botón es para ingresar a otra caja de dialogo en donde se especifica la posición deseada del motor X.
- Button 2 Inserta.., Este botón es para ingresar a otra caja de dialogo en donde se especifica la posición deseada del motor Y.
- Button 5 CERRAR. Como su nombre lo indica, se utiliza para terminar la aplicación.

Para la posición deseada de los motores se tienen dos etiquetas.

- Edit 1 Posición X. Esta etiqueta muestra el valor que se introdujo en la segunda caja de dialogo llamada por el Button 3.
- Edit 2 Posición Y. Muestra el valor que se introdujo en la segunda caja de dialogo llamada por el Button 4.

## **Apartado de visualización**

Se tienen diferentes etiquetas de visualización con su nombre a lado.

La siguiente sección muestra, por medio de una interrupción, los valores en tiempo real de todas las etiquetas de visualización. Aquí se encuentran las siguientes.

- Edit 3 Sensor 1. Aquí se muestra el valor que manda el primer sensor externo, que tiene como función dar la posición de la plataforma. Si se visualiza el valor 5, indica que la plataforma se encuentra en su origen, pero si se visualiza el valor 0, indica que la plataforma se encuentra fuera de su posición origen.
- Edit 4 Sensor 2. Aquí se muestra el valor que manda el segundo sensor externo, que tiene como función dar la posición de la plataforma con respecto al segundo grado de libertad del motor. Al igual, si se visualiza el valor 5, indica que la plataforma se encuentra en su origen, pero si se visualiza el valor 0, indica que la plataforma se encuentra fuera de su posición origen.

- Edit 5 Indicador de nivel. Aquí se muestra el valor del medidor de nivel, para esta tesis se relaciona con el valor del voltaje, siendo la visualización de 5V el indicativo de que el tanque se encuentra lleno, mientras que 0V es el indicativo de que el tanque se encuentra vacío.
- Edit 6 Posición real X. El valor que se muestra en esta etiqueta es solo para corroborar que se encuentre el motor en la posición que se le especificó.
- Edit 7 Posición real Y. Al igual que en la etiqueta anterior, solo es para corroborar que se encuentre en la posición que se le especificó, pero con respecto al segundo motor.

### Cajas de diálogo dependientes

Como se muestra en la figura 2.29, se tienen dos diálogos que dependen de la aplicación principal o caja de dialogo principal.

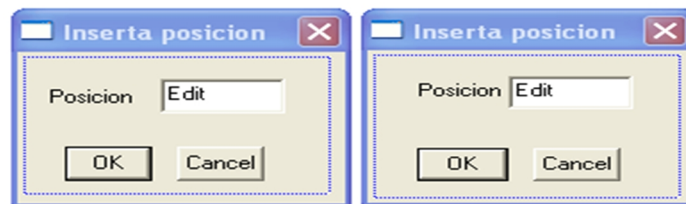


Figura 2.29. Diálogos dependientes para las posiciones X y Y.

Se crea esta dependencia debido a que se tiene una interrupción de 0.5ms, que hace que no sea posible escribir en la aplicación principal.

Las dos cajas de diálogos tienen las siguientes opciones:

- Button 1 OK. Ingresa el valor a la etiqueta Edit 1 de la aplicación principal, para que pueda ser enviada la nueva posición al sistema electrónico.
- Button 2 Cancel. Como su nombre lo indica, cancela el ingreso del valor y se regresa a la aplicación principal.
- Edit X o Edit Y Posición. El operador debe de ingresar el valor de la posición en la que se desea colocar al primer o segundo motor.

### 2.3.3. Descripción en el ámbito del programador

La programación en Visual C++ depende de clases, donde cada una de ellas tiene un uso específico.

Antes de comenzar a programar, se debe iniciar la aplicación con los siguientes pasos en el asistente para la creación de un proyecto nuevo en el entorno de desarrollo Visual C++ 6:

- Primero se debe de escoger el modo MFC basada en cajas de diálogos, donde se podrán utilizar aplicaciones del mismo sistema operativo.
- Colocar los archivos USB.h, USB.ccp, usb2550.dll, usb2550.h y la librería usb2550.lib, en el directorio de trabajo con el propósito de transmitir y recibir datos por medio de este puerto hacia el microcontrolador.
- Instalar el controlador para el reconocimiento del PIC18F4550 en la computadora.

Una vez iniciado el ambiente del proyecto creado, a continuación se describe la programación.

El primer paso es definir en la caja de diálogo con los controles y etiquetas que se quieren para la aplicación que manejará el operador. En el momento de poner estas herramientas, también crean sus respectivas relaciones:

- Se crea el tipo de variable que hace referencia a la etiqueta.
- El control hace referencia a la función que se debe programar.

Las relaciones creadas en esta aplicación se muestran en la figura 2.30.

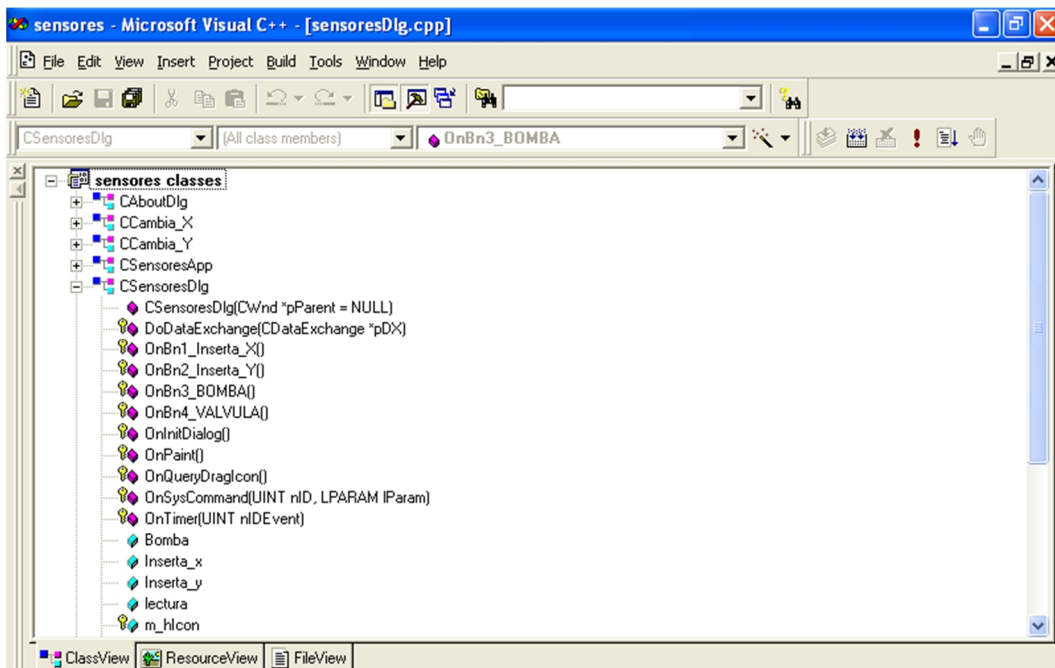


Figura 2.30. Relaciones para la clase CSensoresDlg.

Es importante mencionar que la clase CSensoresDlg es en donde reside la mayoría del código fuente que soporta la aplicación.

Como se muestra en la figura 2.30, aparte de las funciones creadas de cada botón (OnBn), se tienen otras que definen el desempeño de la aplicación. Solamente se crea la función OnTimer(UINT, nIDEvent) para generar una interrupción en tiempo real para la lectura de los datos recibidos por el sistema electrónico.

Después de mostrar todas las funciones asociadas a la aplicación, comienza la visualización de las variables relacionadas con las etiquetas.

En la clase CSensoresDlg se programa el tiempo de la interrupción, como también las funciones. A continuación, en la figura 2.31, se muestra el diagrama de flujo de cada función, debido a que cada una de ellas funciona como un programa individual.

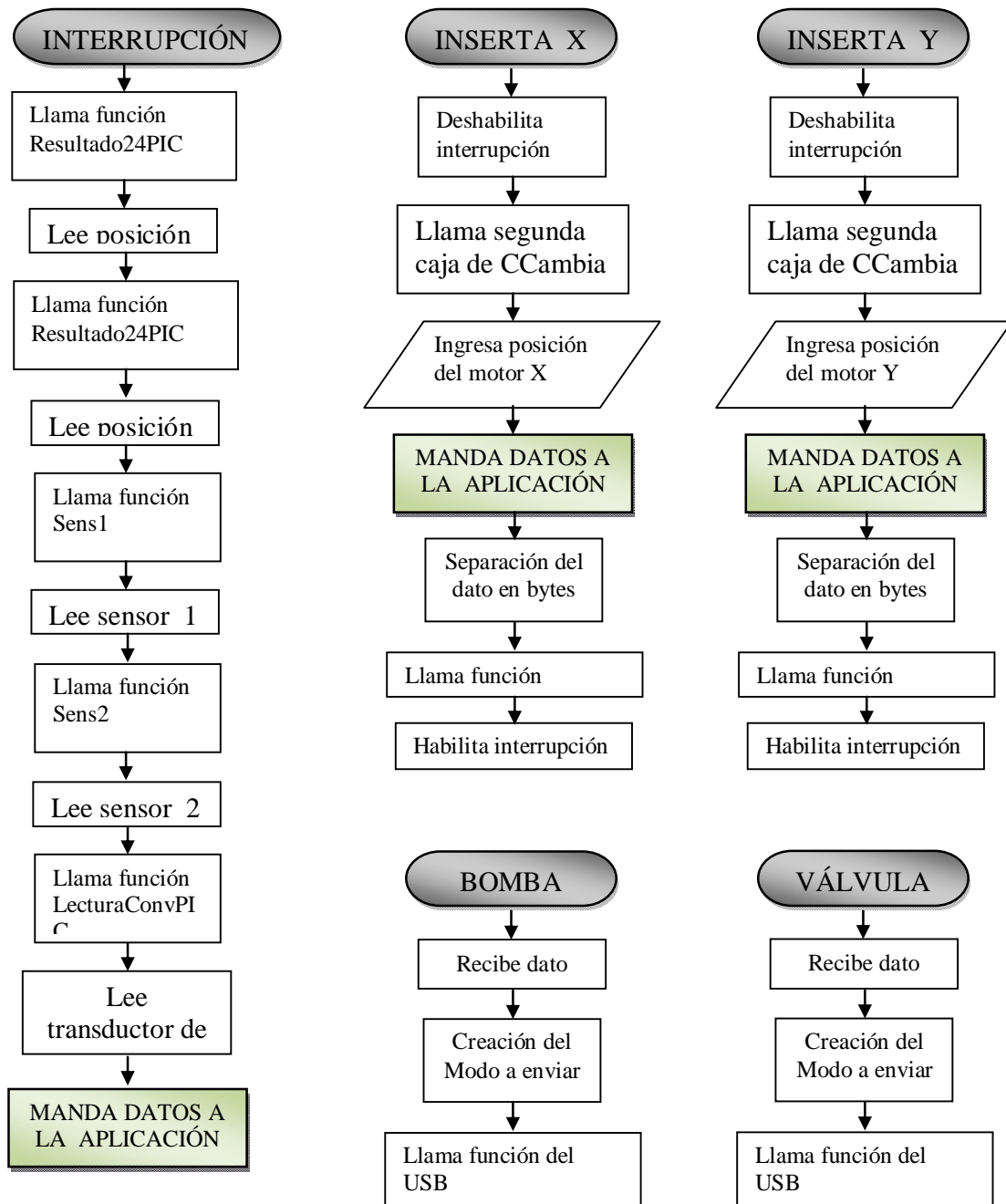


Figura 2.31. Funciones para la clase CSensoresDlg.

En la figura 2.32 se muestra el contenido de la clase USB, donde se crean las funciones de envío y recepción de datos al sistema electrónico.

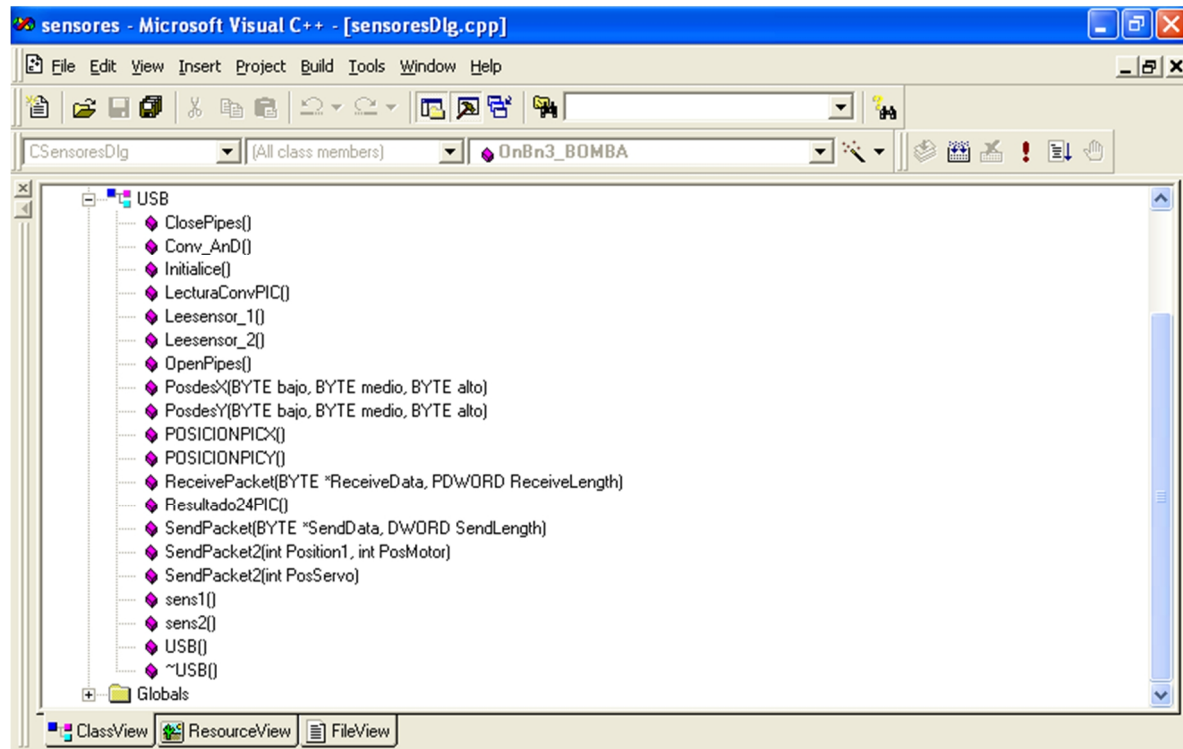


Figura 2.32. Clase USB.

En esta clase se tienen las funciones requeridas para la aplicación, como también las funciones del mismo archivo para que pueda funcionar la interfaz de comunicaciones. Al igual que la clase CSensoresDlg, cada función es un programa individual, por lo que la figura 2.33 muestra los diagramas de flujo de cada función.

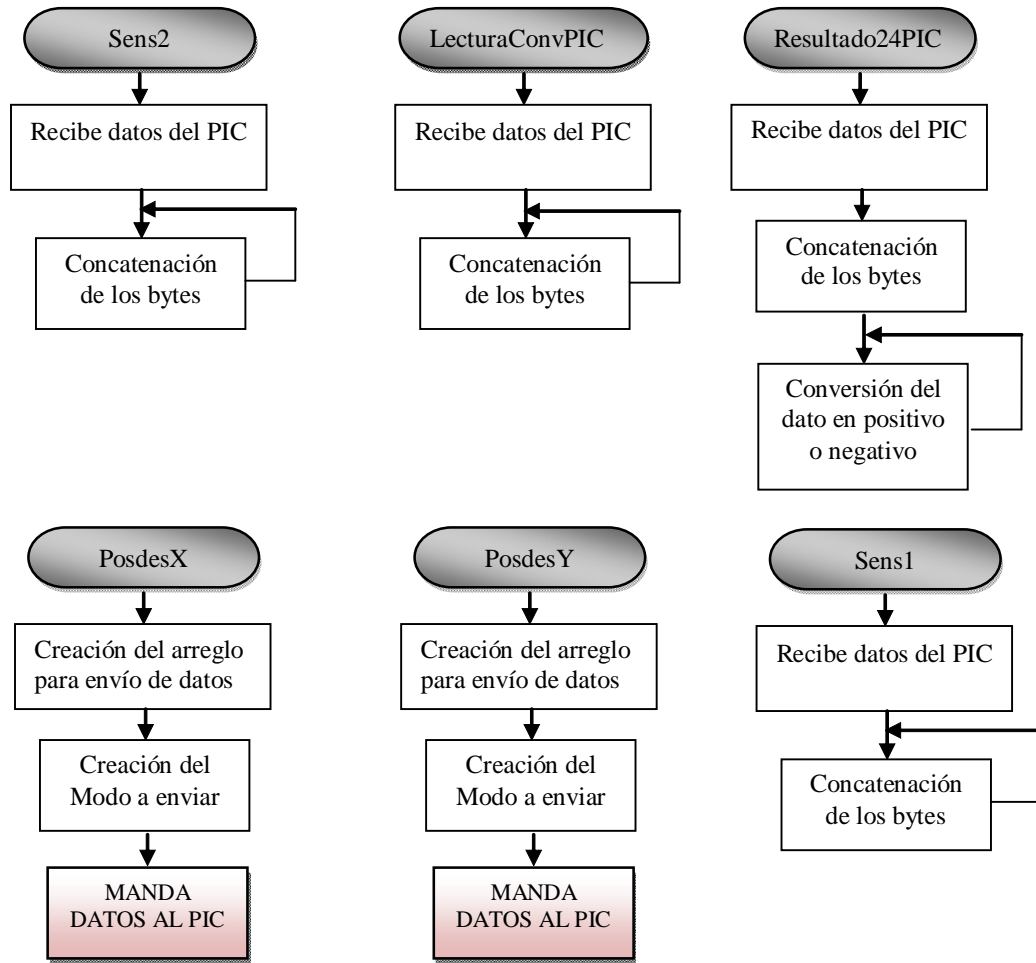


Figura 2.33. Funciones de la clase USB.

Para finalizar el programa, al crearse las dos cajas de diálogo dependientes, entonces también se crearon sus respectivas clases, donde se encuentran sus propias variables relacionadas con sus etiquetas como sus respectivos controles, como se muestra en la figura 2.34.

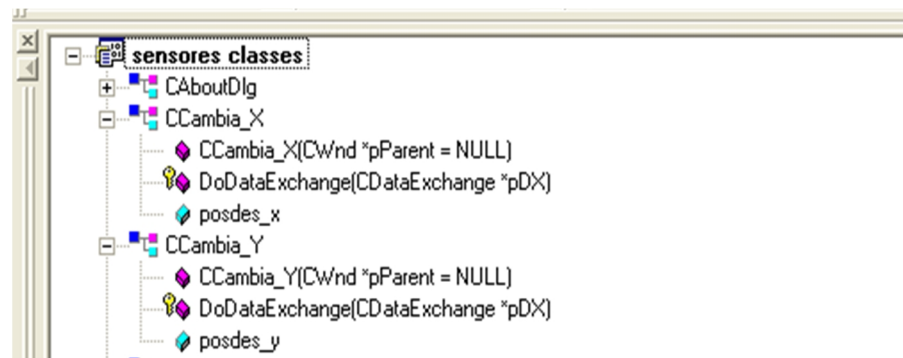


Figura 2.34. Clases de las cajas de diálogo.





# Capítulo 3

## Resultados

Uno de los objetivos planteados en la presente tesis, es que la aplicación realizada para la computadora obtenga los datos del sensor de nivel para su caracterización y determinar su rango de operación. Pero para poder realizar una lectura fiable, se requiere que la parte electrónica funcione con precisión. Un ejemplo, es en el caso de los motores, en donde el operador debe mandar la posición en la que quiere que el motor se establezca, y ésta debe ser precisa para mover la plataforma en un ángulo específico y así tomar el valor del nivel del fluido del tanque en ese instante a través del indicador de nivel de la aplicación. Por lo que se procede a la realización de pruebas entre el software de operación y componente electrónica para obtener el rango de error.

### 3.1. Resultados

El funcionamiento de los sensores, motores, bomba, válvula y transductor de nivel, está dado por la componente electrónica mostrada en la figura 3.1, la cual genera las señales específicas para obtener el comportamiento deseado de los dispositivos mencionados.

A su vez, el sistema electrónico también requiere de la interfaz de operación, debido a que ésta es la que da la orden principal para que microcontrolador PIC18F4550 pueda realizar alguna función. El proyecto final se muestra en la figura 3.2.

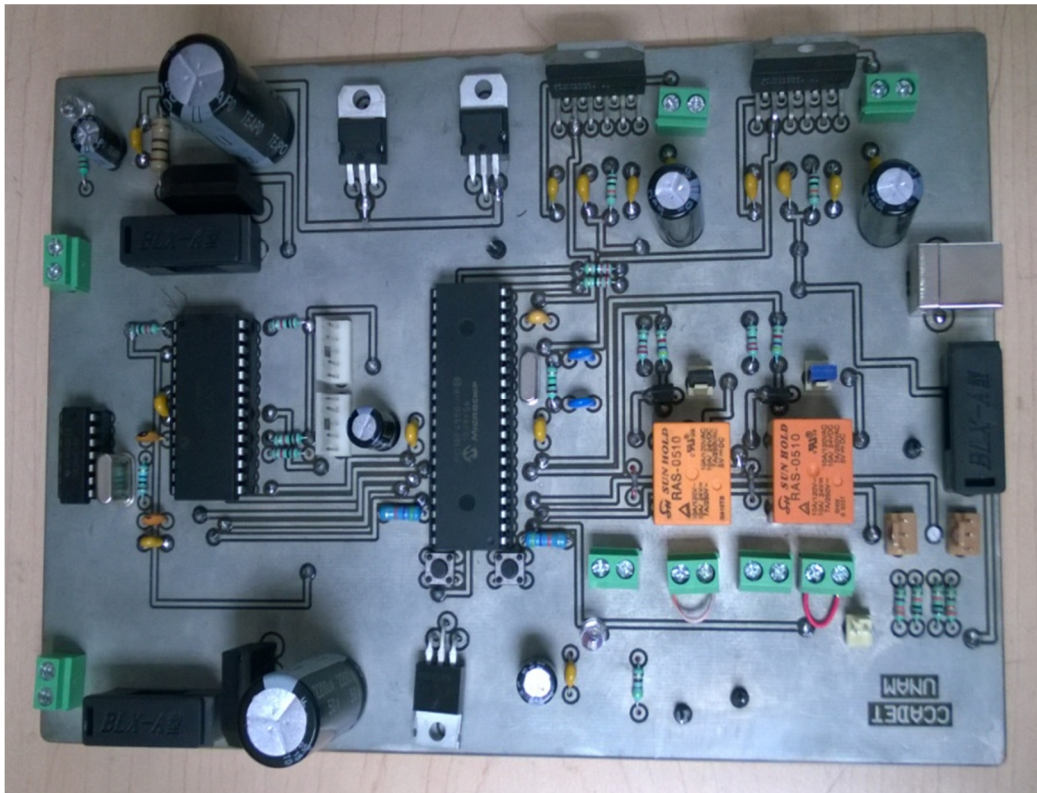


Figura 3.1. Sistema electrónico desarrollado.

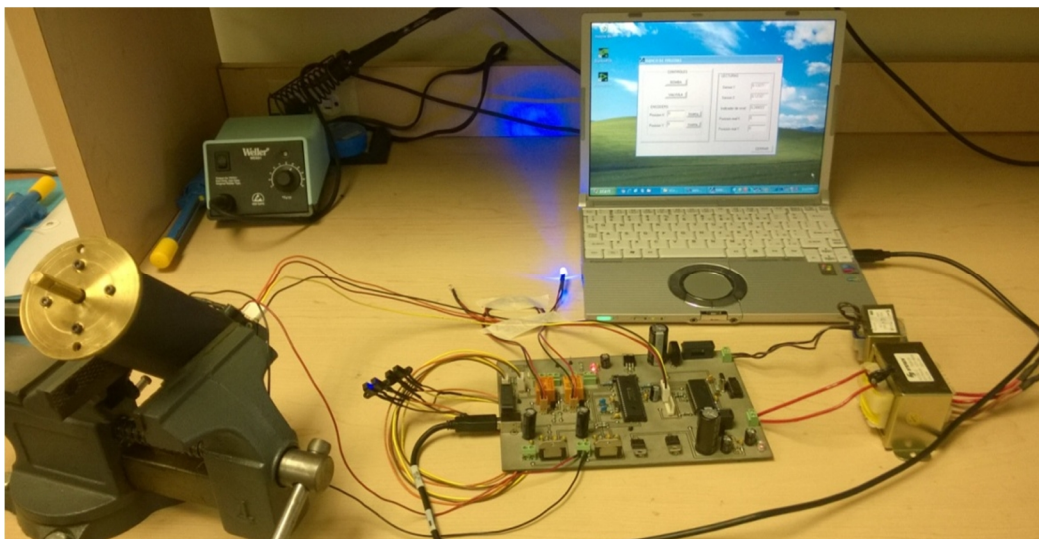


Figura 3.2. Sistema electrónico para la calibración de nivel.

Como se muestra en la imagen, hay una interacción de la aplicación con la componente electrónica, teniendo los dos sensores, los LED's que dan aviso de prendido/apagado, los transformadores y por último los motores. Para el caso de los motores, solo se da la muestra de uno, debido a que solo hay un soporte para que pueda ser sostenido.

A continuación se procede a las pruebas experimentales de cada punto de la aplicación.

### 3.1.1. Desempeño en posición de los motores

En este punto, se obtendrá una gráfica entre la posición deseada por el operador contra la posición real en la que se establece el motor y obtener su margen de error.

Como se muestra en la figura 3.3, al presionar el botón “Inserta” del canal X aparece otra ventana en la que se debe de colocar la posición a la que el motor debe de ir, finalizando con esta tarea con el botón OK de la segunda ventana.



Figura 3.3. Inserción de la posición deseada.

En el momento que se finaliza la tarea, el motor se mueve a la posición que se le indico, mostrando en la aplicación principal la posición ordenada por el operador como también la posición real en la que se queda el motor. Véase la figura 3.4.

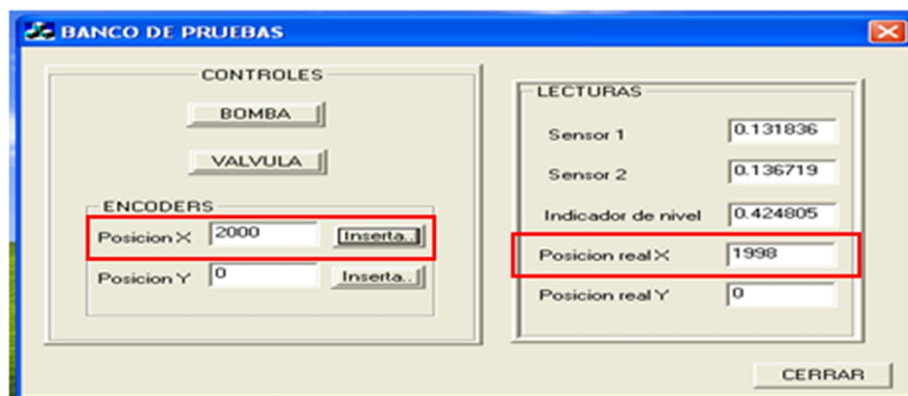


Figura 3.4. Posiciones real y deseada del canal X.

Teniendo este procedimiento, se realiza la captura de ellos repitiendo 5 veces el procedimiento para ver la variación que se presenta, obteniendo la tabla 3.1 y su grafica en la figura 3.5.

MOTOR CANAL X							
Posición deseada	Posición Real					Posición Real promedio	ERROR %
-10141	-10136	-10141	-10141	-10138	-10141	-10139.4	0.015%
-10131	-10119	10131	-10131	-10128	-10131	-10128	0.03%
-10121	-10119	10121	-10120	-10121	-10119	-10120.2	0.007%
-9865	-9867	-9865	-9867	-9865	-9866	-9866	0.01%
-7001	-7007	-7001	-7000	-6999	-7001	-7001.6	0.009%
-6263	-6265	-6265	-6265	-6265	-6263	-6264.6	0.03%
-3095	-3095	-3095	-3095	-3092	-3095	-3094.4	0.02%
-1861	-1864	-1859	-1861	-1861	-1861	-1861.2	0.01%
0	0	0	0	0	0	0	
1000	1002	999	1002	999	999	1000.2	0.02%
2000	1998	1999	1998	1998	1998	1998.2	0.09%
2020	2010	2011	2010	2006	2011	2009.6	0.5%
5000	5001	4998	5001	4999	4999	4999.6	0.008%
5681	5681	5681	5681	5679	5678	5680	0.02%
8672	8672	8669	8672	8672	8676	8672.2	0.02%
10852	10848	10848	10848	10849	10848	10848.2	0.04%
11181	11180	11180	11180	11180	11180	11180	0.008%
11191	11181	11182	11181	11182	11182	11181.6	0.08%

Tabla 3.1. Desempeño en posición del motor en X.

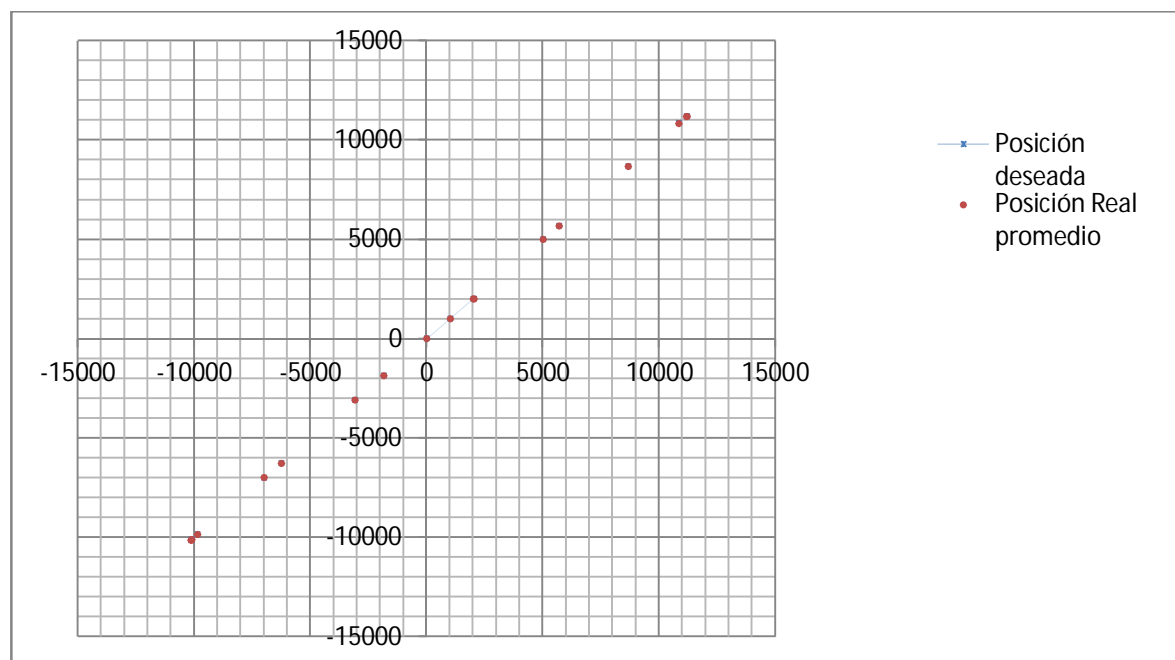


Figura 3.5. Desempeño en posición del motor en X.

En la gráfica se tiene con una línea la posición que debe tener el motor y los puntos rojos muestran la posición real en la que se queda el motor, observando que no es perceptible el desfase entre estas dos variables y esto es verificado con la tabla que se obtuvo.

Para obtener el porcentaje de error se utiliza la siguiente fórmula:

$$\text{Porcentaje de error} = \frac{\text{Posición Real} \times 100\%}{\text{Posición deseada}} - 100\%$$

Se procede a realizar el mismo procedimiento descrito pero para el otro motor. Se obtienen los valores de la misma manera que se hizo con el motor del canal X, solo que la inserción de valores y lecturas de datos se da en sus respectivas casillas, es decir, canal Y, como se muestra en la figura 3.6.

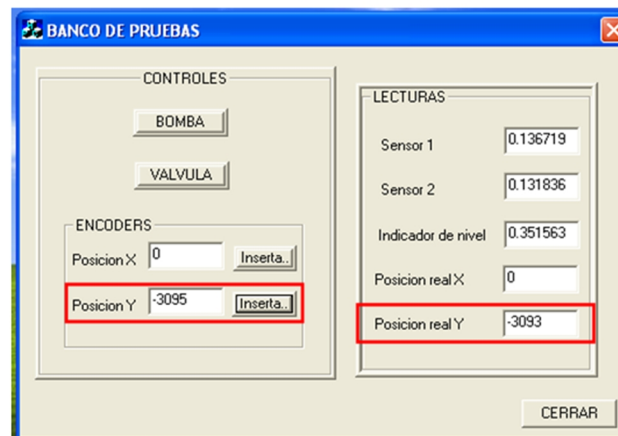


Figura 3.6. Posiciones real y deseada del canal Y.

Los resultados obtenidos se muestran en la tabla 3.2 y figura 3.7.

MOTOR CANAL Y							
Posición Deseada	Posición Real					Posición Real promedio	ERROR %
-10141	-10139	-10138	-10137	-10138	-10139	-10138.2	0.03%
-10131	-10124	10127	-10128	-10128	-10131	-10127.6	0.03%
-10121	-10122	10121	-10120	-10121	-10119	-10120.6	0.004%
-9865	-9865	-9863	-9863	-9865	-9866	-9864.4	0.006%
-7001	-6999	-7000	-6995	-6998	-7000	-6998.4	0.04%
-6263	-6262	-6263	-6263	-6262	-6263	-6262.6	0.006%
-3095	-3093	-3095	-3092	-3092	-3095	-3093.4	0.05%
-1861	-1858	-1859	-1861	-1861	-1861	-1860	0.05%
0	0	0	0	0	0	0	
1000	995	995	994	994	995	994.6	0.54%
2000	1995	1990	1994	1995	1995	1993.8	0.31%
2020	2008	2009	2009	2006	2008	2008	0.6%
5000	4998	4996	4993	4994	4994	4995	0.1%
5681	5673	5673	5678	5677	5678	5675.8	0.09%

<b>8672</b>	8668	8669	8669	8667	8665	<b>8667.6</b>	<b>0.05%</b>
<b>10852</b>	10844	10848	10848	10849	10848	<b>10847.4</b>	<b>0.04%</b>
<b>11181</b>	11176	11177	11174	11171	11175	<b>11174.6</b>	<b>0.06%</b>
<b>11191</b>	11185	11182	11178	11176	11176	<b>11179.4</b>	<b>0.10%</b>

Tabla 3.2. Desempeño en posición del motor en Y.

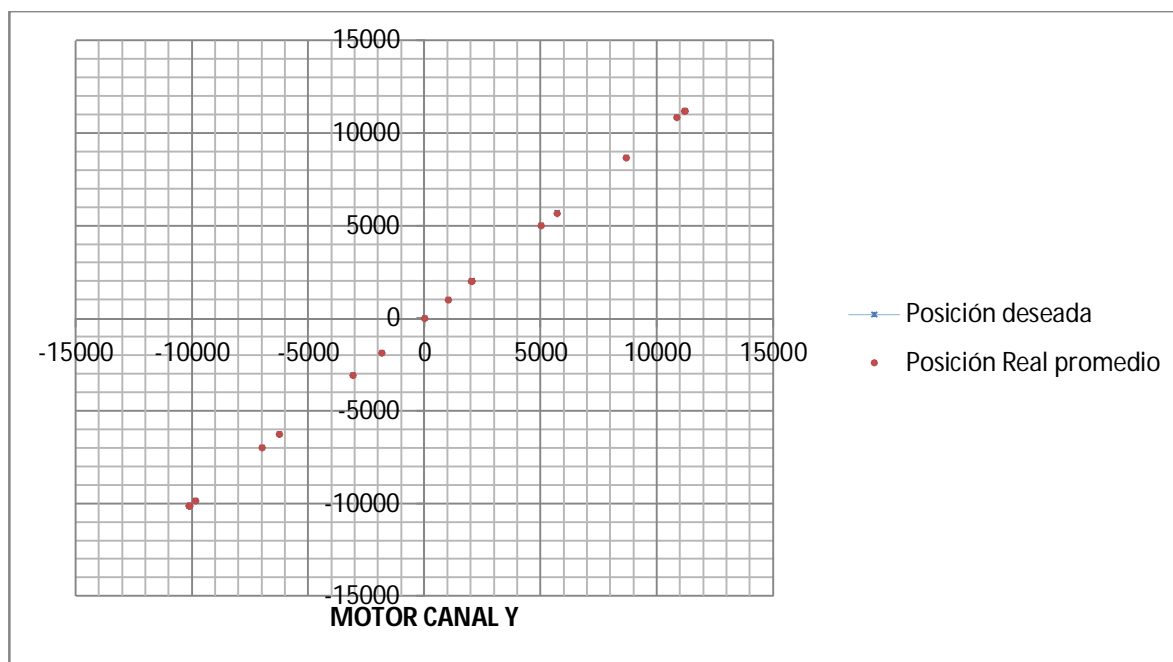


Figura 3.7. Desempeño en posición del motor en Y.

Como se observa en la gráfica, no se nota desfase entre los dos valores, debido a que su rango de error es mínimo.

### 3.1.2. Indicador de nivel

El indicador de nivel, despliega el valor de voltaje que entregue el transductor de nivel que se encuentra adentro del tanque. Para realizar esta lectura, se necesita que el indicador de nivel sea preciso, por lo que se analizará su rango de error con respecto a una fuente de voltaje como se muestra en la figura 3.8.

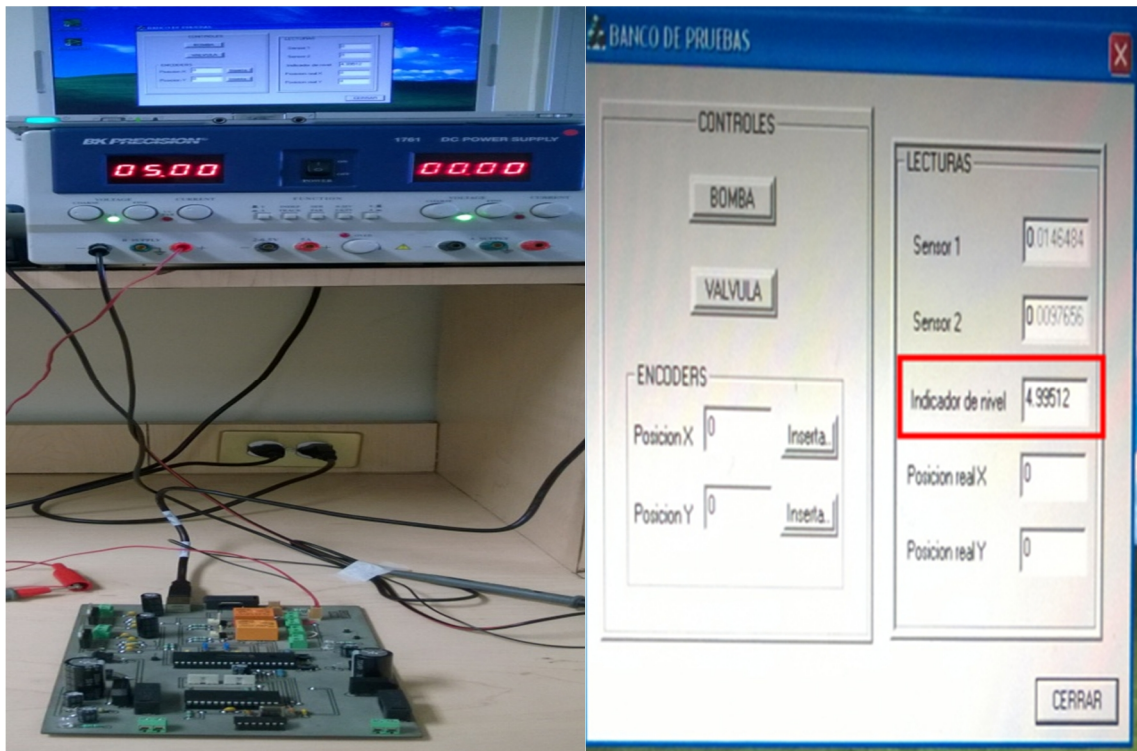


Figura 3.8. Experimento para evaluar la medición de nivel.

Obteniendo las lecturas que se muestra la tabla 3.3.

FUENTE DE VOLTAJE	INDICADOR DE NIVEL	Porcentaje de error
5	4.99	0.20%
4.43	4.44	0.22%
4.05	4.09	0.99%
3.5	3.52	0.57%
3.09	3.11	.64%
2.51	2.53	0.80%
2	2.02	1%
1.51	1.52	0.66%
1.04	1.06	1.9%
0.35	0.36	2.8%

Tabla 3.3. Resultados de la medición de nivel.

Con estos valores se muestra la gráfica de la figura 3.9, donde se puede apreciar un desfase entre ambos valores, que es confirmado con el porcentaje de error de la tabla.

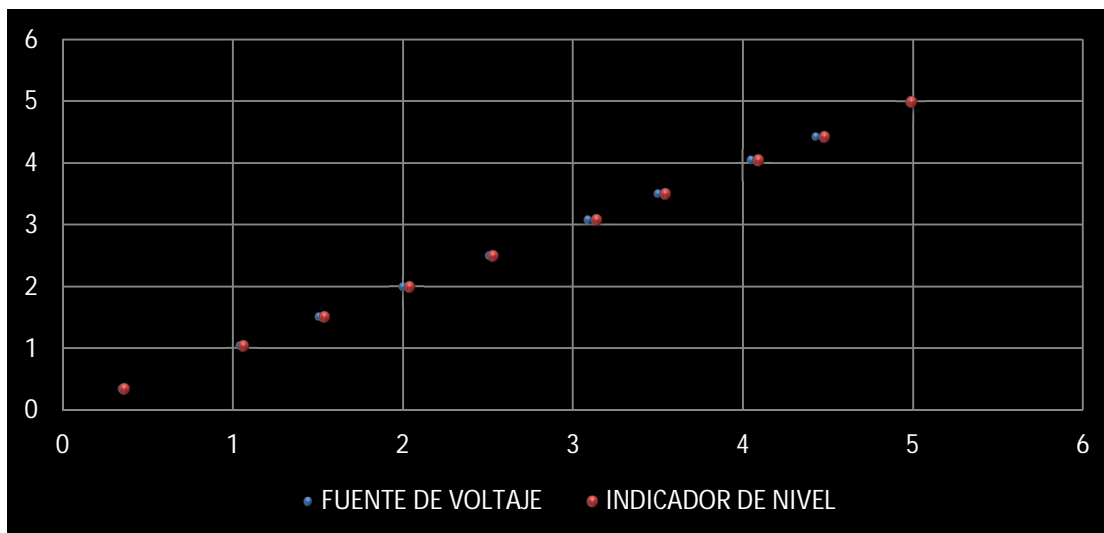


Figura 3.9. Resultados de la medición de nivel.

### 3.1.3. Bomba y válvula

En éste caso, como ambos botones son interruptores que van a un relevador, solo se procede a verificar el funcionamiento de la aplicación con dos LED's que se encuentran en la entrada respectiva. Como se muestra en la figura 3.10 al presionar los botones BOMBA y VALVULA hacen que los LED's se prendan indicando que comienza el funcionamiento de los dispositivos.

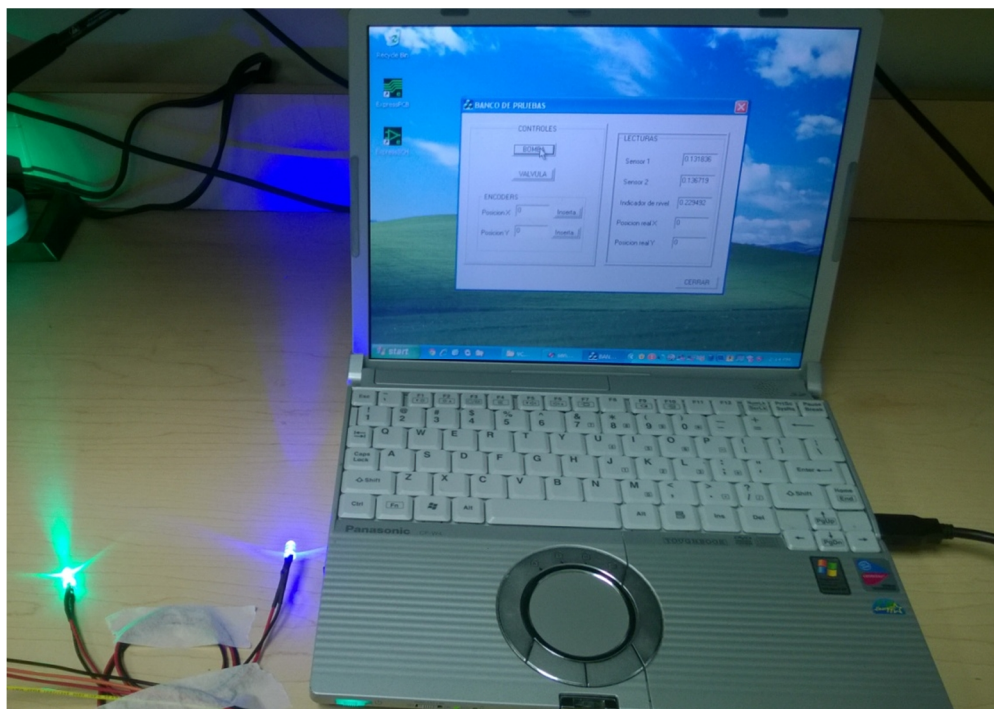


Figura 3.10. LEDs de bomba y válvula encendidos.



En la figura 3.11 se observa que el LED conectado a la salida de la bomba se encuentra encendido, mientras que al presionar en el botón VALVULA de la aplicación hace que no mande señal el PIC por lo que automáticamente se apaga el LED conectado hacia el circuito que es para la válvula.

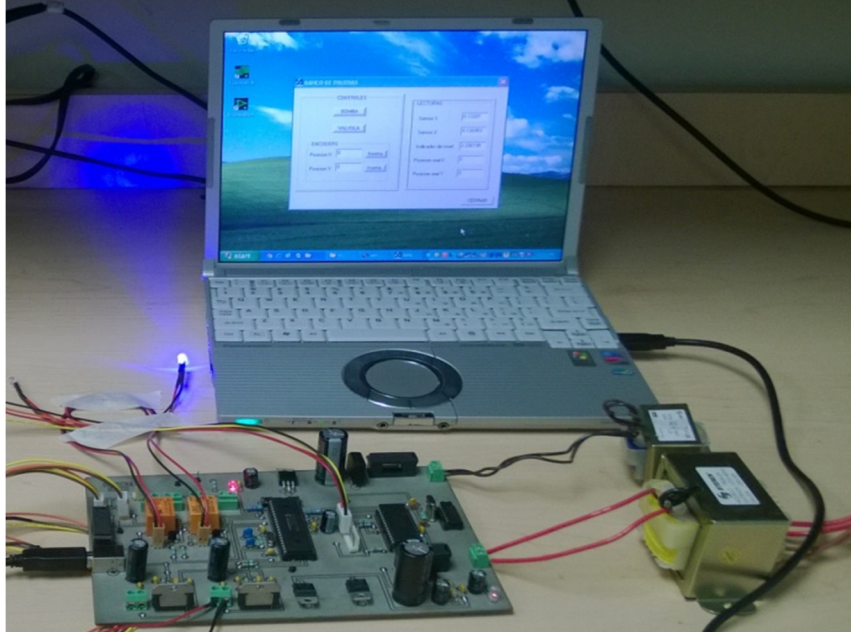


Figura 3.11. LED de bomba encendido y LED de válvula apagado.

Por otra parte, en la figura 3.12 ahora se muestra que al presionar en ambos botones ahora se apaga el LED que se encuentra en el circuito destinado para la bomba, mientras que el LED de la válvula se prende

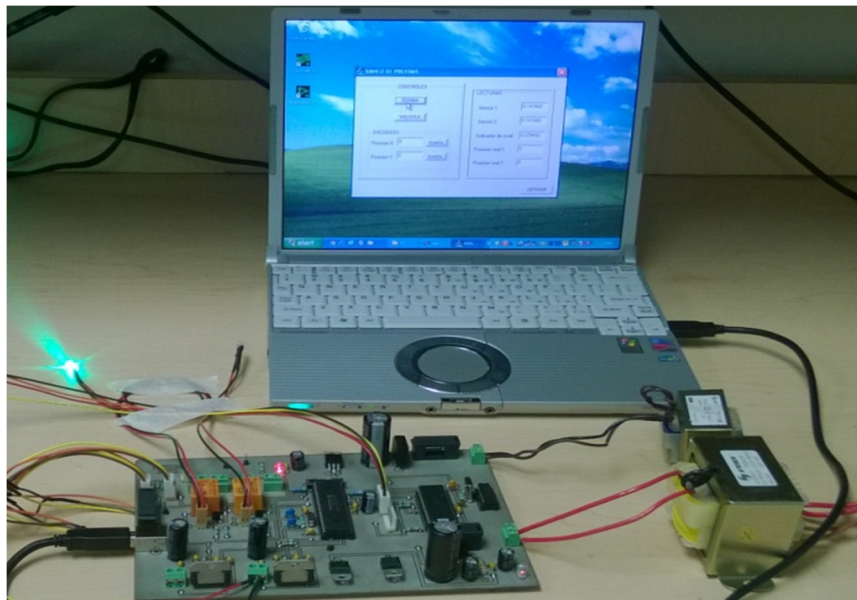


Figura 3.12. LED de bomba apagado y LED de válvula encendido.

### 3.1.4. Sensores de posición inicial

Los sensores son solo indicadores de la posición en la que se encuentra la plataforma, por lo que solo se procede a verificar que si mandan una señal en el momento de ser interrumpidos por un objeto opaco.

En las figuras 3.13 y 3.14 no se logran apreciar los resultados en la aplicación, a pesar de ello se describe a continuación.

En la figura 3.13 al colocar una tarjeta para hacer una interrupción en el sensor, éste deja de mandar la señal y por lo tanto el voltaje se va a cero y esto es mostrado en la aplicación.

En la figura 3.14 se dejan los sensores libres, por lo que mandan una señal de 5 V que al igual es mostrada en la aplicación.



Figura 3.13. Interrupción del sensor.

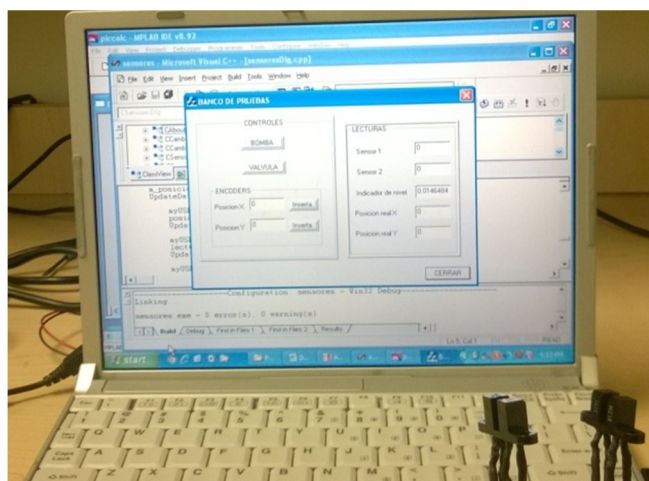


Figura 3.14. Sensores libres de interrupción.

## 3.2. Discusión

Mostrados los valores de cada una de las pruebas se procede a su análisis.

Como se observa en los resultados de la sección 3.1.1, ambos motores mantienen un margen de error menor al 1% por lo que se cumple el objetivo de poder controlar los motores a voluntad del operador. Por otra parte, también se demuestra el funcionamiento que tiene la fórmula de control proporcional.

$$\text{Ciclo} = 511 + (0.6)(\text{posición deseada} - \text{posición real})$$

Donde la precisión es dada con la constante proporcional.

No obstante, cabe destacar que el margen de error sube cuando se colocan valores muy cercanos a la posición especificada previamente, provocando que el motor no llegue a moverse o su movimiento sea mínimo.

Esta prueba es realizada con materiales del CCADET y son para verificar el funcionamiento de la componente electrónica, por lo que es probable que cambie el comportamiento en el momento de colocar la parte mecánica.

Por otra parte, el objetivo del indicador de nivel es poder dar una lectura fiable del contenido del fluido que tiene el tanque, y esto puede realizarse con la ayuda de un transductor de medición de nivel.

Por lo que estrictamente, la prueba que se debe hacer con éste sistema, es colocar un tanque móvil con su medidor de nivel para que también se conecte hacia el indicador de nivel de la aplicación. Una vez, conectado se debe de tomar el valor de ambos medidores con diferentes cantidades de fluido para observar el rango de error. (Esta prueba se debe hacer con el tanque en reposo para tener una medición precisa).

Los contras que se tienen y por lo que ya no se hizo este análisis, es porque éstos materiales no se encuentran disponibles en el CCADET y su costo es elevado como para poder comprarlo. Esto demuestra la limitación que se tiene al no tener la componente mecánica.

Aun así, el funcionamiento principal de cualquier medidor de nivel electrónico, es que se utilizan transductores que proporcionan en la salida un voltaje, que para ésta tesis, debe de ir hacia la componente electrónica realizada y ser mostrada en el indicador de nivel.

Teniendo en cuenta éste funcionamiento, se procedió a medir la precisión que tiene el indicador de nivel con relación a una fuente de voltaje. Con lo que se observa en los resultados obtenidos, que el porcentaje de error se mantiene en un rango de 0.5% a 1%, observando que en los dos últimos valores de la fuente y del indicador, a pesar de estar muy cercanos entre sí, el valor del porcentaje de error incrementa, pero aun así es justificable debido a que números menores a 1 en el lugar del divisor dan valores altos por lo que se procede a despreciarlos.

Teniendo resultados satisfactorios en la prueba, se comprueba que ésta parte del sistema cumple con el objetivo.

Por último, los demás sistemas mostrados demuestran que todo el proyecto de ésta tesis es funcional y puede realizar satisfactoriamente todos los puntos especificados inicialmente, aunque los resultados pueden variar en el momento que se tenga una componente mecánica.

### **3.3. Trabajo a futuro**

El objetivo principal es llevar a cabo el desarrollo y construcción de la componente mecánica por medio de los ingenieros mecánicos del CCADET para poder adaptar el prototipo electrónico y así realizar los cambios pertinentes en el programa lógico del sistema electrónico o hasta hacer cambios en la misma electrónica aplicada.

# Conclusiones

En esta tesis, se realizó un prototipo para la calibración de sensores de nivel en tanques móviles, con el objetivo de realizar estudios para mejorar la lectura de los sensores.

En la realización de cualquier prototipo se debe de tener noción del funcionamiento principal que se quiere, como también lo que conlleva.

En este caso, para la creación de éste prototipo, en primera instancia solo se quiere obtener el valor que da el transductor de nivel en un tanque que se encuentra en movimiento, lo que conlleva a varias incógnitas a resolver:

- ¿Cuál es el funcionamiento de un sensor en un tanque móvil y como plasmarlo en el proyecto?
- ¿Cómo se llenará el tanque?
- ¿Cómo vaciar el tanque?
- ¿Dónde se colocará el tanque?
- ¿Cómo realizar el movimiento del tanque?
- ¿Cómo mostrar los resultados?
- ¿Qué mostrar en la computadora?
- ¿Cómo realizar las funciones de la aplicación para que den el valor requerido?

Una vez resueltas las primeras incógnitas, entonces se analiza qué material se tiene disponible y cuál de ellos es el apropiado para cada función, para que al final solo se tenga que comprobar por medio de pruebas que el prototipo tenga el funcionamiento deseado.

Así se obtiene el resultado de una componente electrónica y un software de operación que pueden ser manejados por los operadores.

Teniendo la creación de la parte gráfica que puede manipular todos los componentes electrónicos, como también adquirir las señales de ellos para poder hacer una caracterización. Todo esto por medio del microcontrolador que tiene la capacidad de manejar los dispositivos electrónicos que se requieren.

El funcionamiento es demostrado por medio de las gráficas y tablas que muestran el rango de error de cada subsistema, dando un valor aceptable para que pueda ser utilizado confiablemente.

Todos estos puntos son los que hacen a un Ingeniero, que como lo dice la misma palabra es tener el ingenio y visión para poder resolver un problema y más aún, resolverlo con los materiales que tenga a su alcance.

# Bibliografía

Bolaños D., (2015), “Capacitor de Bypass”, [On line], Available at: <http://www.bolanosdj.com.ar/TEORIA/CAPACITBYPA.pdf>, [Accessed on 25 de noviembre de 2015].

Bruzos T., (2015), “Indicador de nivel de combustible del automóvil”, [On line], Available at: <http://www.sabelotodo.org/automovil/relojgasolina.html>[Accessed on 25 de noviembre de 2015].

Dee J., (2015), “Pulse Width Modulation”, [On line], Available at: <https://learn.sparkfun.com/tutorials/pulse-width-modulation>, [Accessed on 25 de noviembre de 2015].

Fairchild, (2014), *LM78XX 3 Terminal 1A Positive Voltage Regulator*, Fairchild, USA, pp 24.

Fairchild, (2001), *Phototransistor Optical Interrupter Switch*, Fairchild, USA, pp 5.

Huan H. W., (2005), *PIC Microcontroller: An Introduction to Software and Hardware Interfacing*, Thomson Delmar Learning, USA, pp 791.

Instituto Norbert Wiener, (2015), “Introducción al Visual C++”, [On line], Available at: [http://informatica.uv.es/iiguia/AIG/docs/intro\\_visual.htm](http://informatica.uv.es/iiguia/AIG/docs/intro_visual.htm) [Accessed on 25 de noviembre de 2015].

Microchip, (2007), *PIC18F2455/2550/4455/4550 Data Sheet*, Microchip, USA, pp 430.

Microchip, (2007), *MCHPFSUSB Firmware User's Guide*, Microchip, USA, pp 28.

- National Semiconductor, (1996), *LMD18201 3A, 55V H-Bridge*, National Semiconductor, USA, pp 9.
- Norman S. Nise, (2004), *Sistemas de control para ingeniería*, Compañía editorial continental, Primera reimpresión México, pp 12.
- Ogata K., (1996), *Sistemas de control entiempodiscreto*, Prentice Hall, México, pp 757.
- Omron, (2013), *PCB Relay G5LE*, Omron, USA, pp 6.
- On Semiconductor, (2005), *Complementary Silicon Plastic Power Transistors*, On Semiconductor, USA, pp 6.
- Rectron Semiconductor, (2005), *Single-Phase Glass Passivated Silicon Bridge Rectifier*, Rectron Semiconductor, USA, pp 2.
- Rubio A, (2003), *Diseño de circuitos y sistemas integrados*, Ediciones UPC, España, pp 230.
- ServoSystems, (2004), *Relyance Brush Type DC Servomotor*, ServoSystems, USA, pp 2.
- Texas Instruments, (2015), *SNx4HC04 Hex Inverters*, Texas Instruments, USA, pp 29.
- USDigital, (2000), *Encoder to microprocessor interface chip*, USDigita, USA, pp 6.



# **Anexos**

Se presentan en esta sección los detalles de la implementación electrónica.

## **A. Diagrama esquemático**

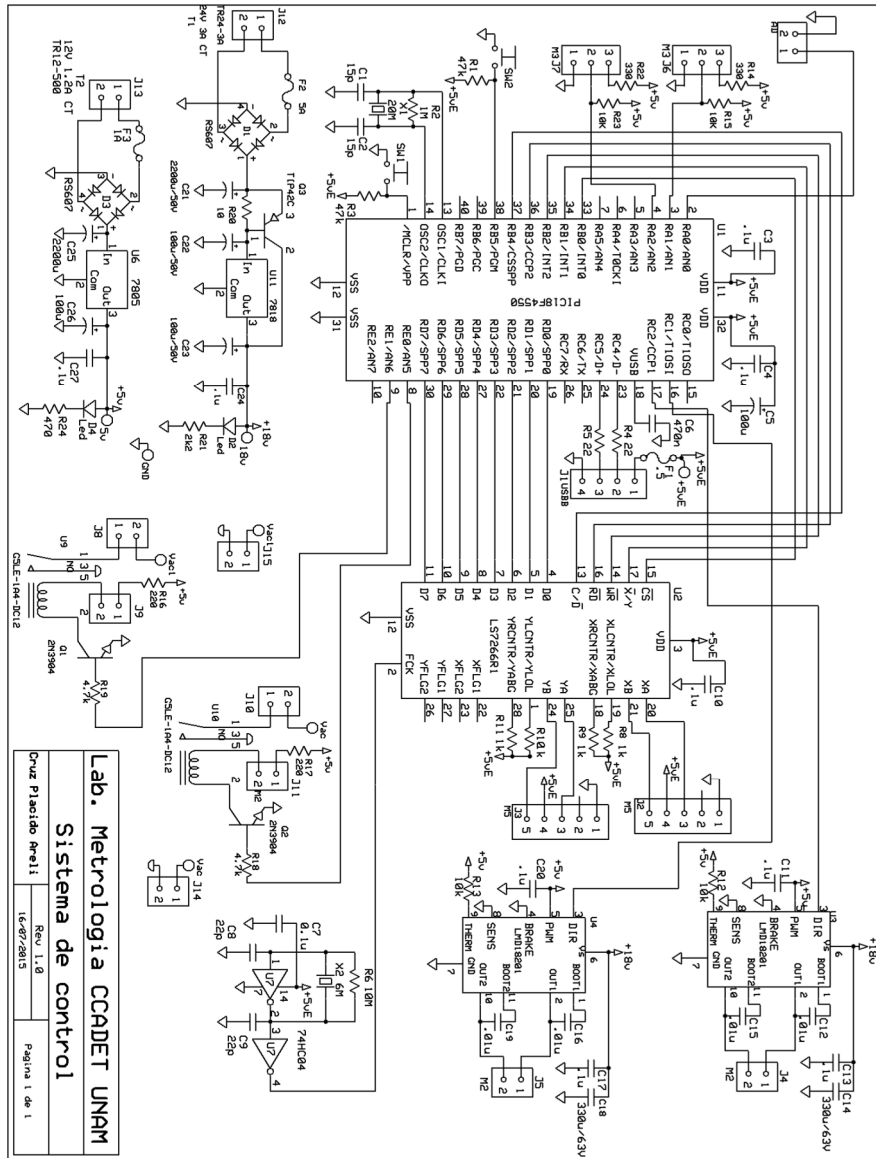


Figura A.1. Diagrama esquemático.

## B. Circuito impreso

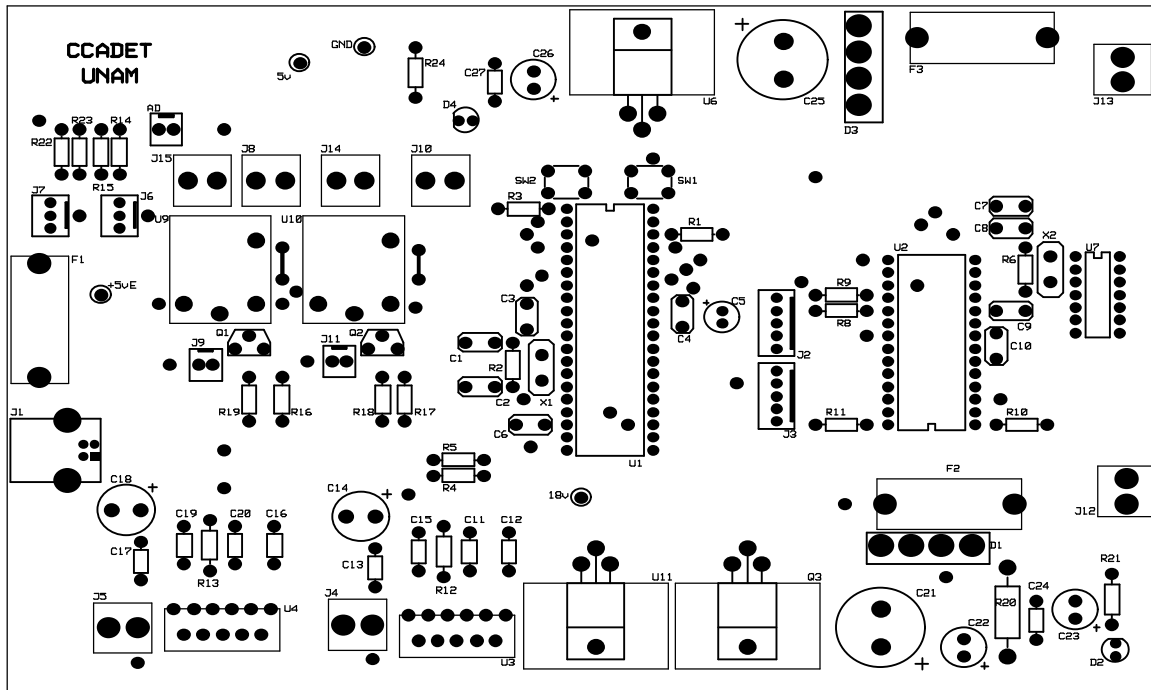


Figura A.2. Circuito impreso, mascarilla.

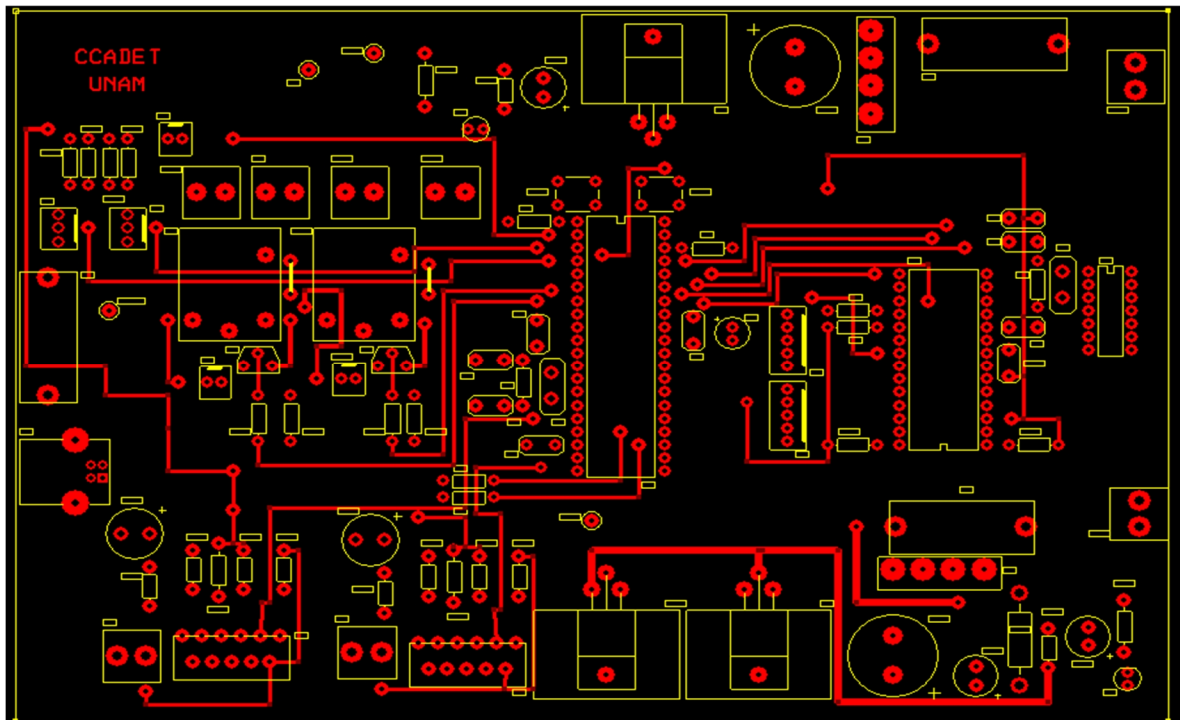


Figura A.3. Circuito impreso, componentes.

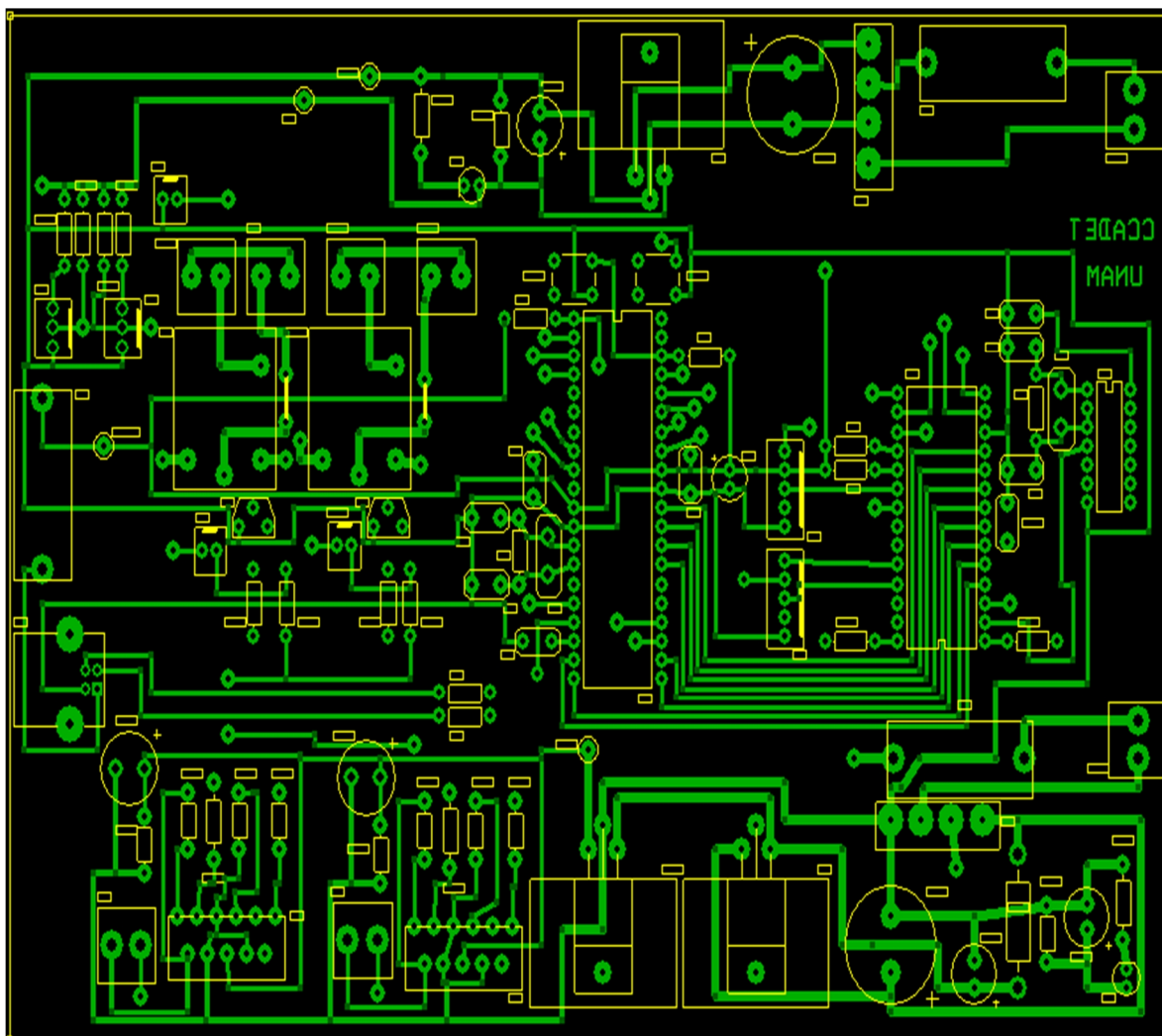


Figura A.4. Circuito impreso, soldadura.

### C. Lista de partes

TIPO	NOMBRE DE PARTE	PROVEEDOR
+5vE	Header AG	Electronica HEADER-1
18v	Header AG	Electronica HEADER-1
5v	Header AG	Electronica HEADER-1
AD	Molex 2 pines	Molex 2 pines Ultimo Recurso
C1	15pF	AGElectronica CC-15/1000V-WELSON
C2	15pF	AGElectronica CC-15/1000V-WELSON
C3	.1uF	AgElectronica CM-.1/50V
C4	.1uF	AgElectronica CM-.1/50V
C5	100uF	AgElectronica CE-100/10V-TEAPO

C6	470nF	AgElectronica CM-.47/50V
C7	0.1uF	AgElectronica CM-.1/50V
C8	22pF	AG Electronica CC-22/1000V
C9	22pF	AG Electronica CC-22/1000V
C10	.1uF	AGElectronica CM-.1/50V
C11	.1uF	AGElectronica CM-.1/50V
C12	.01uF	AgElectronica CM-.01/50V
C13	.1uF	AgElectronica CM-.1/50V
C14	330uF	AG Electronica CE-330/63V
C15	.01uF	AgElectronica CM-.01/50V
C16	.01uF	AgElectronica CM-.01/50V
C17	.1uF	AgElectronica CM-.1/50V
C18	330uF	AG Electronica CE-330/63V
C19	.01uF	AgElectronica CM-.01/50V
C20	.1uF	AgElectronica CM-.1/50V
C21	2200uF	AG Electronica CE-2200/50V-TEAPO
C22	100uF	AG Electronica CE-100/50V-TEAPO
C23	100uF	AG Electronica CE-100/50V-TEAPO
C24	.1uF	AgElectronica CM-.1/50V
C25	2200uF	AG Electronica CE-2200/50V-TEAPO
C26	100uF	AG Electronica CE-100/50V-TEAPO
C27	.1uF	AgElectronica CM-.1/50V
D1	RS607	AG Electronica KBL04
D2	Led	Steren 5/ROJO
D3	RS607	AG Electronica KBL04
D4	Led	Steren 5/ROJO
F1	.5A	Steren FEUR 5
F2	5A	Steren FEUR 5
F3	1A	Steren FEUR 1
GND	Header	AG Electronica HEADER-1
J1	USBB	AG-USBB/F90_DIP
J2	Molex 5	Ultimo recurso Molex 5 posiciones
J3	Molex 5	Ultimo recurso Molex 5 posiciones
J4	Molex 2	Ultimo recurso Molex 2 posiciones
J5	Molex 2	Ultimo recurso Molex 2 posiciones
J6	Molex 3	Ultimo recurso Molex 3 posiciones
J7	Molex 3	Ultimo recurso Molex 3 posiciones
J8	Terminal con 2 tornillos	Steren TRT-02
J9	Molex 2	Ultimo recurso Molex 2 posiciones
J10	Terminal con 2 tornillos	Steren TRT-02
J11	Molex 2	Ultimo recurso Molex 2 posiciones
J12	Terminal con 2 tornillos	Steren TRT-02
J13	Terminal con 2 tornillos	Steren TRT-02
J14	Terminal con 2 tornillos	Steren TRT-02
J15	Terminal con 2 tornillos	Steren TRT-02

Q1	2N3904	AgElectronica 2N3904
Q2	2N3904	AgElectronica 2N3904
Q3	TIP42C	AG-TIP42C
R1	47k $\Omega$	AG- R47k 1/4
R2	1M $\Omega$	AG- R1M 1/2
R3	47k $\Omega$	AG- R47k 1/4
R4	22 $\Omega$	AG- R22 1/4
R5	22 $\Omega$	AG- R22 1/4
R6	10M $\Omega$	AG- R10M 1/2
R8	1k $\Omega$	AG- R1k 1/4
R9	1k $\Omega$	AG- R1k 1/4
R10	1k $\Omega$	AG- R1k 1/4
R11	1k $\Omega$	AG- R1k 1/4
R12	10k $\Omega$	AG- R10k 1/2
R13	10k $\Omega$	AG- R10k 1/2
R14	330 $\Omega$	AG- R330 1/2
R15	10K $\Omega$	AG- R10k 1/2
R16	220 $\Omega$	AG-R220 1/4
R17	220 $\Omega$	AG- R220 1/2
R18	4.7k $\Omega$	AG- R4.7k 1/2
R19	4.7k $\Omega$	AG- R4.7k 1/2
R20	10 $\Omega$	Steren-R10 1
R21	2.2k $\Omega$	Steren-R2k2 1/4
R22	330 $\Omega$	AG- R330 1/2
R23	10K $\Omega$	AG- R10k 1/2
R24	470 $\Omega$	Steren R470 1/4
SW1		Steren-AU-1012
SW2		Steren-AU-1012
U1	PIC18F4550	AG-PIC18F4550-I/P
U2	LS7266R1	
U3	LMD18201	Newark 41K2746
U4	LMD18201	Newark 41K2746
U6	7805	AG Electronica L7805CV
U7	74HC04	AG-74HC04D
U9	G5LE-1A4-DC12	Newark 36K7449
U10	G5LE-1A4-DC12	Newark 36K7449
U11	7818	AG-MC7818CTG
X1	20MHz	AG-20MHZ-MINI
X2	6MHz	AG-6MHZ-MINI

Tabla A.1. Lista de partes.